

01129  
28



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

CONTROL DE TEMPERATURA CON LOGICA  
DIFUSA

T E S I S  
QUE PARA OBTENER EL TITULO DE  
INGENIERO ELECTRICO - ELECTRONICO  
PRESENTAN

JUAN LUIS GONZALEZ MANCILLA  
CARLOS MAYA LEON



DIRECTOR DE TESIS:  
ING. ROBERTO MACIAS PEREZ

MEXICO, D. F.

2003



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

TESIS CON  
FALLA DE ORIGEN

**Con Agradecimiento para:**

El *Ing. Roberto Macías Pérez* por su enorme ayuda y comprensión en la dirección de esta tesis.

**Con Honor para:**

El *Alma Mater* que nos vio crecer y enseñó a saborear todas las mieles del camino de la sencillez y la humildad.

**INDICE**

<b>CAPITULO 1</b> .....	3
<b>CAPITULO 2</b> .....	6
2.1 INTRODUCCIÓN.....	6
2.2 MOTIVACIONES DEL PROYECTO.....	6
2.3 PROBLEMA.....	7
2.4 SISTEMA DE CONTROL AMBIENTAL.....	11
2.5 PLANTEAMIENTO.....	12
<b>CAPITULO 3</b> .....	17
3.1 PRINCIPIOS DE CONTROL.....	17
3.2 LOGICA DIFUSA.....	18
3.3 PLANTEAMIENTO DEL CONTROL DIFUSO.....	26
<b>CAPITULO 4</b> .....	34
4.1 INTRODUCCIÓN.....	34
4.2 TECLADO y LCD.....	35
4.3 SENSOR DE TEMPERATURA.....	36
4.4 ADC.....	37
4.5 MICROPROCESADOR.....	38
4.6 ETAPA DE SALIDA.....	39
<b>CAPITULO 5</b> .....	42
5.1 INTRODUCCIÓN.....	42
5.2 EL KERNEL DE LA LÓGICA DIFUSA.....	42
5.3 DESARROLLO DEL KERNEL.....	43
5.3.1 MÉTODO DE DIFUSIÓN.....	44
5.3.1.1 CALCULO DE LAS PENDIENTES DE LOS CONJUNTOS.....	44
5.3.2 EVALUACIÓN DE REGLAS.....	56
5.3.3 MÉTODO DE DESDIFUSIÓN.....	60
5.2 COMUNICACIÓN CON EL USUARIO.....	66
<b>CAPITULO 6</b> .....	69
6.1 INTRODUCCION.....	69
6.2 RESULTADOS.....	69
6.3 CONCLUSIONES.....	71
<b>APÉNDICE A</b> .....	74
I. DESCRIPCIÓN.....	74
II. CARACTERÍSTICAS.....	74
III. CONEXIÓN LCD-CPU.....	74
III.1 CONTRASTE.....	75
IV. SOFTWARE.....	76
<b>APÉNDICE B</b> .....	79
<b>APÉNDICE C</b> .....	93
<b>BIBLIOGRAFIA</b> .....	122

## CAPITULO 1

### INTRODUCCION

En la época prehistórica, el fuego fue el elemento primordial que utilizó el ser humano para calentar su morada. La sombra y el agua fueron los elementos que utilizó para poder contrarrestar el exceso de calor. A lo largo de la historia se ha podido observar que el ser humano ha tratado de manipular las condiciones térmicas que prevalecen en un cierto medio, disipando calor o realizando el enfriamiento del medio que lo rodea. En la época medieval prevalecieron los grandes castillos con sus enormes chimeneas que sólo calentaban el área que las rodeaba. Los romanos eran propietarios de construcciones que contaban con una calefacción notablemente buena, calentando el aire y haciéndolo circular por paredes y pisos cóncavos. En los pueblos del Medio Oriente, en donde prevalece el clima seco, colgaban mantas húmedas frente a las puertas y, de esta manera, poder obtener una forma primitiva de enfriamiento por evaporación. Más tarde, con un poco más de desarrollo tecnológico y evolución de la ciencia, Leonardo da Vinci diseñó un gran enfriador por evaporación.

Sin embargo, el verdadero desarrollo de la calefacción y del sistema de aire acondicionado comenzó aproximadamente a fines del siglo pasado, aunque los sistemas de aire acondicionado, por medio de refrigeración mecánica, han tenido su desarrollo en los últimos cincuenta años.

En las últimas décadas se ha percibido un gran desarrollo en el ámbito eléctrico y electrónico, las aplicaciones se han incrementado y se ha llegado a campos que no se habían considerado, el campo de acondicionamiento de aire y calefacción no es la excepción.

Así, se han desarrollado sistemas de calefacción y aire acondicionado, en los cuales la mayoría del control es electrónico, se ha podido automatizar el sistema para acrecentar el confort del ser humano y, gracias a la nueva tecnología, se ha logrado reducir el tamaño de dichos sistemas en una proporción considerable.

Actualmente es muy común encontrar sistemas de calefacción y acondicionamiento de aire en grandes industrias, oficinas, casas habitación, centros de comercio, instalaciones deportivas y en la mayor parte de los transportes como son los automóviles, camiones, aviones, barcos, trenes, etc.

El proyecto que se presenta es un controlador de temperatura y su diseño adopta los principios del acondicionamiento de aire y calefacción. El sistema basa su estructura y control en la lógica difusa, herramienta que permite sustituir a sistemas clásicos “on/off” cuyo funcionamiento era controlado por la lógica binaria.

A lo largo de la historia, los principios de probabilidad han sido la principal herramienta para representar la incertidumbre en modelos matemáticos y por ello, toda la incertidumbre era asumida como una caracterización de incertidumbre aleatoria.

Sin embargo, la teoría de los conjuntos difusos permiten adoptar una forma de representar incertidumbre asociada con vaguedad, con imprecisión y con la carencia de información con respecto a un elemento particular. De igual forma, la teoría difusa da la posibilidad de utilizar variables lingüísticas en sustitución de variables cuantitativas que permitan representar conceptos imprecisos.

Obviamente, los principios de calefacción y acondicionamiento de aire no varían mucho con respecto a lo que aquí se menciona, pero si cabe mencionar que el control para acondicionar la temperatura a un estado confortable para el ser humano, si difiere un poco de los que generalmente se encuentran en diversos libros.

Las condiciones de temperatura, humedad, limpieza y movimiento del aire, para acondicionar algún sistema cerrado, son de enorme importancia, ya que su interacción con el comportamiento físico del cuerpo humano permite establecer si una persona se siente confortable o molesta con el ambiente que prevalece en un cierto medio.

La mayor parte de los sistemas de control ambiental son usados para dar mayor confort a las personas, pero también son usados para obtener las condiciones que se requieren en determinados procesos.

Se presenta un proyecto en el cual se tienen que considerar las condiciones ambientales a controlar para poder proporcionar un confort adecuado para el ser humano, con esto, se presentará un diseño apto para la aplicación deseada.

En el capítulo dos se analizará el sistema como una aplicación general, considerando al sistema una caja negra, esto ayudará a analizar los diversos principios del sistema de control ambiental: calefacción, ventilación y acondicionamiento de aire. Posteriormente se analizará las partes que constituyen a un sistema de control ambiental. Con esto se podrá tener una idea de las partes que serán de utilidad para conformar el sistema que tendrá la aplicación deseada. Este mismo capítulo permitirá analizar sistemas muy sencillos de control ambiental, así como diseños muy completos cuyo control electrónico es la base de su éxito. De la misma manera, se justificará el porqué utilizar lógica difusa para el control y no otro principio.

Al final del capítulo dos, se sugerirá un diseño con el cual se controlará la temperatura en el interior de un recinto utilizando los principios de lógica difusa y las partes necesarias para el desarrollo del sistema, realizando los pasos adecuados para cubrir las necesidades planteadas.

En el capítulo tres se realiza una descripción de lo que es la lógica difusa. La herramienta difusa es una alternativa para describir con precisión lo vago o no plenamente definido, permite adoptar un concepto ambiguo y establecer el grado de pertenencia de un elemento a un conjunto; en cambio lo aleatorio describe la incertidumbre en la ocurrencia de un evento.

Se exponen sus orígenes y principios, su teoría, la comparación con la lógica común o que todos conocen (lógica Aristotélica, binaria o “crisp”), las aplicaciones en la que es

participe, las herramientas que hasta la fecha se tienen para poder trabajar con ella y las ventajas y desventajas que se presentan al desarrollar un sistema difuso.

El capítulo cuatro hace referencia a toda la parte física del sistema (Hardware). La parte física esta dividida en partes: la parte de entrada o de sensado, la parte de control, en donde se aplica la lógica difusa y la parte de salida o de potencia. Toda la electrónica del sistema es expuesta en este capítulo, se hace una descripción de los componentes que conforman cada etapa, de la comunicación entre ellos mismos y de la comunicación entre una etapa y otra.

Todo lo referente a la programación del sistema (Software) es expuesto en el capítulo cinco. Se hace mención a la parte microcontroladora, a las herramientas ocupadas para la realización de programas, diagramas de flujo y listados de cada operación realizada por el sistema. Se mostrará la capacidad del programa principal y se hará mención a las diversas contrariedades que se puedan presentar debido a posibles modificaciones en el programa principal, y de esta manera poder recomendar soluciones para un mejor funcionamiento en el control.

Finalmente se tendrán las conclusiones en las cuales se expondrán los puntos de vista finales y definitivos acerca del comportamiento del sistema. Dentro de este capítulo se hará mención a las diversas impresiones que se experimentaron en el momento en que se realizaron pruebas al sistema, si se detectaron errores se mencionarán y se expondrán las soluciones que se tomaron.

## CAPITULO 2

### PLANTEAMIENTO DEL PROYECTO

#### 2.1 Introducción.

En muchas ocasiones el ser humano ha tenido que ingeniárselas para poder luchar contra las situaciones adversas que presenta el clima en un cierto ambiente o en cierta región, por ejemplo, el exceso de calor, el frío o las heladas, la lluvia o la humedad, etc.. Así, existe una tendencia a crear espacios ambientales con características favorables que puedan cumplir con las exigencias del cuerpo humano.

Sin embargo, para poder considerar los diversos parámetros que intervienen en el control y acondicionamiento de un ambiente es conveniente abundar en ciertos conceptos y comportamientos, tanto del medio ambiente como del cuerpo humano y de esta manera, concluir en adecuadas herramientas que permitan plantear diversas soluciones para la obtención de condiciones ambientales interiores que proporcionen un confort adecuado al ser humano.

Las soluciones para poder adaptar el clima de un medio a las necesidades del ser humano pueden ser diversas, muy rudimentarias o sistemas altamente sofisticados; en realidad, los sistemas pueden ser pequeños o enormes, aplicaciones domésticas o en vehículos, así como aplicaciones industriales, pero lo importante es que el ser humano utiliza lo que tiene (ciencia y tecnología) para obtener bienestar propio, adaptando el medio que lo rodea a sus necesidades.

#### 2.2 Motivaciones del Proyecto

Se sabe que la temperatura de confort para el ser humano está entre los 21° y 27° centígrados, pero esto depende de la humedad que prevalezca en el sitio donde se desea mantener este rango.

A lo largo de la historia, uno de los principales objetivos del ser humano ha sido mantener, dentro de un grado aceptable de perfección y estabilidad, los principales estándares que permiten la comodidad de los compartimientos que el ser humano utiliza para su estancia.

Ciertamente, para que un medio sea confortable, es necesario que mantenga un cierto grado de temperatura compatible con el nivel de temperatura al cual el ser humano se siente a gusto. En este punto es donde surge la dificultad: ¿Cómo saber a que nivel de temperatura se siente a gusto una persona?, ¿todas las personas se sienten a gusto a un mismo nivel de temperatura?. Es ilógico pensar que todas las personas sienten comodidad en un mismo grado de temperatura, sin embargo, la tecnología tiene que permitir desarrollar técnicas para dicho fin y que se adapten a las exigencias de cada persona.

De esta manera, dentro de las necesidades que se podrían establecer para el diseño de un compartimiento óptimo, la parte correspondiente al control de temperatura ha originado una severa importancia, no solamente por el confort, sabemos que el funcionamiento de diversos equipos eléctricos y electrónicos exigen, para su buen funcionamiento, un sistema el cual mantenga rangos de temperatura establecidos por los propios fabricantes.

Para poder entender la necesidad de desarrollar un control de temperatura, sería conveniente y necesario mencionar el comportamiento térmico del cuerpo humano. Se sabe que el cuerpo del ser humano es una máquina de calor. Al cuerpo se le introduce combustible llamado alimento y éste lo transforma en energía. Dicha energía es la utilizada en cada actividad de la vida cotidiana.

El alimento que se ingiere diariamente sirve para que el cuerpo conserve una temperatura interna, así como en la profundidad de los tejidos de  $37^{\circ}\text{C}$ . Se sabe que la temperatura a la que permanece la piel es cercana a los  $32^{\circ}\text{C}$ . La temperatura ambiente promedio a la que el ser humano se siente cómodo es de  $24^{\circ}\text{C}$ . Se sabe también que el cuerpo irradia calor en forma continua. Como el calor se desplaza siempre de los sitios calientes a los fríos, existe un desplazamiento de calor debido a esa diferencia que hay entre la temperatura interior del cuerpo ( $37^{\circ}\text{C}$ ) y la temperatura ambiente promedio ( $24^{\circ}\text{C}$ ).

El cuerpo tiene una forma de asegurar que se irradiará hacia el aire la cantidad apropiada de calor. Si por alguna razón el cuerpo no puede liberar suficiente calor, se percibe calor excesivo o se sufre de postración a causa del calor. En caso contrario, si se pierde demasiado calor se presenta la hipotermia. De esta manera, para poder controlar esta disipación de calor, el cuerpo perspira o suda.

El cuerpo suda en todo momento. Lo normal es que a medida que aparece humedad sobre la piel, de inmediato se evapora hacia el aire. Este mecanismo funciona bien en tanto el aire que rodea al cuerpo humano está seco. Si el aire ambiente contiene demasiada humedad no absorberá la perspiración del cuerpo humano y de inmediato surgirá la incomodidad. Se percibirá calor y sudor porque el cuerpo pierde su mecanismo normal de disipar calor.

Por otro lado, resulta de la misma importancia bajar la temperatura ambiente como disminuir la humedad del aire. En una habitación caliente surgirá la comodidad siempre y cuando, el aire se mantenga seco.

De esta manera, se considera de vital importancia, para obtener la comodidad en el interior de un compartimiento, el control de la temperatura, considerando los parámetros del aire que se relacionan con ella: humedad y movimiento.

### **2.3 Problema**

El calor puede describirse como una forma de energía en transferencia. El calor es la forma de energía que se transmite de un cuerpo a otro debido a una diferencia de temperatura. Hay que hacer notar que el calor solamente puede pasar, en forma

natural, desde una temperatura mayor a una menor. Desde luego, si no existe una diferencia de temperaturas, no presentará flujo de calor. Recordemos que la temperatura es una medida de la actividad térmica de un cuerpo, la cual depende de la velocidad de las moléculas y demás partículas, de las cuales se compone cualquier cuerpo.

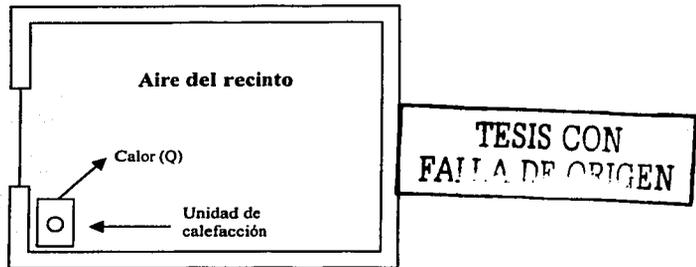


Fig. 2.1 Flujo del calor dentro del compartimiento

Se sabe que el calor es una manifestación de energía y por lo tanto cumple con la primera ley de la termodinámica o la ecuación de la energía: "La energía no se crea ni se destruye". De esta forma, se dice que en un sistema, el cambio de energía total es igual a la energía agregada al sistema menos la energía eliminada del mismo. Esto puede expresarse de la siguiente forma:

$$E_{\Delta} = E_i - E_o \quad \dots (2.a)$$

donde:

$E_{\Delta}$  = Variación de energía almacenada en el sistema.

$E_i$  = Energía de entrada al sistema.

$E_o$  = Energía de salida del sistema.

De esta forma, se concluye que el calor no puede destruirse, ya que cumple con la ley de la energía, pero sí se puede controlar y desplazar. Un acondicionador de aire, por ejemplo, ayuda a retirar exceso de calor en un determinado compartimiento y puede disiparlo al exterior.

Para poder adoptar el planteamiento de este problema, donde se desea controlar el calor de un compartimiento, buscando el óptimo confort para el ser humano, es necesario abundar en ciertos temas que serán base primordial en la solución que se tome para la realización del proyecto que se expone.

A lo largo de los años, se ha mencionado que la energía puede clasificarse en energía en transferencia y energía almacenada. Dos formas de energía almacenada son la energía cinética y la energía potencial. La primera es la energía que almacena un cuerpo debido a su movimiento o velocidad, mientras que la segunda es la energía que

almacena un cuerpo debido a su posición o elevación. Pero se sabe que los cuerpos también almacenan energía debido a su temperatura y presión. Por ejemplo, un gas a alta presión tiene energía almacenada, el agua a alta temperatura mantiene almacenada energía calorífica.

De tal manera que, se le llama entalpía a la energía almacenada en forma de temperatura y presión. Teniendo conocimiento de lo que es entalpía, podemos hacer mención del calor sensible y calor latente.

Cuando se agrega o elimina calor a una sustancia y esto provoca un cambio de temperatura, se conoce a el cambio de entalpía en la sustancia como cambio de calor sensible. Sin embargo, cuando este calor agregado o eliminado a la sustancia provoca un cambio de estado, entonces se conoce a el cambio de entalpía en la sustancia como cambio de calor latente.

Auxiliándose con la ecuación 2.a y aplicándose a una sustancia cualquiera, se puede determinar la relación entre la cantidad de calor que se agrega o que se elimina y la variación de calor sensible. De esta manera tenemos: variación de calor sensible o entalpía igual al calor neto agregado o eliminado, es decir:

$$Q = mC(T_1 - T_2) \dots (2.b)$$

y si se involucra al tiempo:

$$Q' = mCT' \dots (2.c)$$

donde:

$Q'$  = Variación de calor con respecto al tiempo.

$m$  = Masa de la sustancia.

$C$  = Calor específico de la sustancia.

$T'$  = Variación de la temperatura con respecto al tiempo.

El principal objetivo que plantea el proyecto es poder controlar la temperatura interna de un compartimiento, controlando el calor absorbido y transferido, dependiendo del comportamiento de la temperatura ambiente.

La absorción de calor de un compartimiento se puede presentar de diferentes formas. La energía calorífica que emiten los rayos solares puede ser absorbida por medio de la superficie del mismo recinto, así como los accesorios que puedan formar parte de su estructura.

Otra fuente de energía calorífica que se presenta es la irradiada por el mismo cuerpo humano.



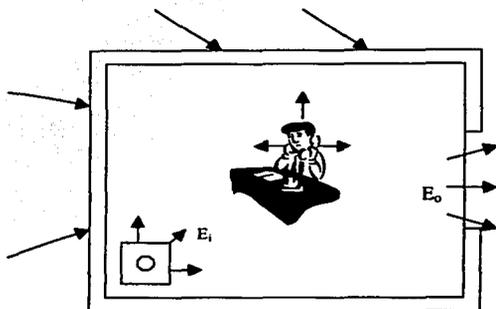


Fig. 2.2 Diversas formas de transferencia de calor.

Entre las alternativas para reducir la absorción de calor, podemos mencionar, en primer término, el tipo de color de la estructura del compartimiento. Los colores claros reflejan más y absorben en menor grado la energía solar, de esta manera, se obtendrá un grado menor de temperatura en el interior. Los colores oscuros absorben más y reflejan menos, por lo tanto, se mantendrá un mayor grado de temperatura en el interior.

El calor del sol irradiado al interior del compartimiento tiende a aumentar debido a que accesorios, como los vidrios, lo pueden reflejar hacia el interior, evitando la disipación al exterior, a lo que se le llama efecto de invernadero. Pero si se tiene un vidrio de color (polarizado), éste refleja cantidades considerables de energía solar.

Así, podemos encontrar diversas formas para poder disminuir la absorción de energía calorífica al interior del compartimiento, pero la realidad es que no siempre tendrán éxito, ya que tiene que considerarse un parámetro de gran importancia que envuelve al problema que se está exponiendo: la temperatura ambiente. Sin embargo, en este proyecto se planea controlar la temperatura de un sistema, controlando el flujo de calor absorbido, obviamente considerando la temperatura exterior al compartimiento y considerando la transferencia de calor que se pueda presentar.

Por otro lado, la disminución de temperatura, en un compartimiento, se debe, principalmente a la transferencia de calor que existe desde el aire caliente que pueda existir en el interior del compartimiento, hasta el aire frío del exterior; esta transferencia de calor se presenta a través de las paredes, ventanas y demás partes del compartimiento. La infiltración es otra forma de perder calor debido a las fugas de aire frío que se presentan.

Para poder contrarrestar las pérdidas de calor que se presenten, lo que produce una disminución de temperatura, es necesario agregar continuamente energía al interior del compartimiento, y de esta manera mantener la temperatura deseada.

TESIS CON  
FALLA DE ORIGEN

## Control de temperatura con lógica difusa

La cantidad de energía calorífica que se debe suministrar, con el objetivo de mantener el aire del compartimiento a una temperatura deseada, se llama carga de calefacción, la cuál es consecuencia de pérdida de calor por transferencia de calor y pérdidas por infiltración o ventilación.

La transferencia de calor puede efectuarse de tres modos distintos: radiación, conducción y convección.

La transferencia de calor por radiación se presenta entre dos cuerpos separados como resultado de la llamada radiación electromagnética, conocida también como movimiento ondulatorio.

La transferencia de calor por conducción es una forma de transferencia que se presenta a través de un cuerpo que no manifiesta movimiento alguno, puede decirse que es el resultado de acciones moleculares o electrónicas. Esta forma de transferencia es muy común a través de los sólidos y de cuerpos en contacto.

En cambio, existe otra forma de transferencia más común en líquidos y gases, es la convección. Esta forma de transferencia de calor es aquella que resulta del movimiento global de líquidos o gases.

Si el movimiento de un fluido se origina por una diferencia de densidad que acompaña a un cambio en la temperatura, a la transferencia de calor producida de esta manera se le llama convección natural. Cuando un fluido es obligado a moverse por la acción de una bomba o ventilador, la transferencia de calor producida de esta manera se llama convección forzada.

Teniendo la mayoría de las herramientas que se ocuparán en el desarrollo del proyecto, se procederá plantear el diseño que se propone para la solución del problema que se expone. Si en la evolución del presente proyecto, se llegara a solicitar la explicación de alguna herramienta, base o antecedente que no se haya mencionado con anterioridad, se proporcionarán las correspondientes fichas bibliográficas, o si fuese indispensable mencionar dicha parte, se abundará en ella sobre la marcha.

### **2.4 Sistema de Control Ambiental**

Un Sistema de Control Ambiental (SCA) se puede definir como aquel sistema que está conformado por los siguientes subsistemas: calefacción, ventilación y acondicionamiento de aire. Obviamente, al considerarlos como tal, cada uno realiza una función diferente, pero siempre con un fin común.

El acondicionamiento de aire es el proceso en el cual se realiza un tratamiento de aire en un ambiente interior con el objetivo de establecer y mantener los estándares requeridos de temperatura, humedad, limpieza y movimiento, para una comodidad exigida por el ser humano.

Es decir, los principales parámetros del aire pueden ser manipulados o controlados para determinar un cierto comportamiento del mismo. Si se desea controlar la temperatura del aire, esto se logra calentando o enfriando al mismo. Con lo que respecta a la humedad (contenido de vapor de agua en el aire), se controla aumentado o eliminando vapor de agua en el aire. La limpieza puede lograrse mediante la filtración eliminando contaminantes indeseables con filtros u otros dispositivos. Otra medida para la limpieza es la ventilación, haciendo introducir aire del exterior al interior, con lo cual se diluye la concentración de contaminantes. Cuando se habla de movimiento de aire, se refiere a la velocidad y a los lugares donde se puede desplazar y distribuir.

La calefacción es aquel sistema que permite suministrar calor dentro de un compartimiento, mientras que al sistema que elimina este calor se le llama sistema de enfriamiento.

De esta manera, podemos resumir que para la realización del proyecto se tienen que tomar en consideración tres importantes partes: una fuente de calefacción, una fuente de enfriamiento y un sistema de distribución.

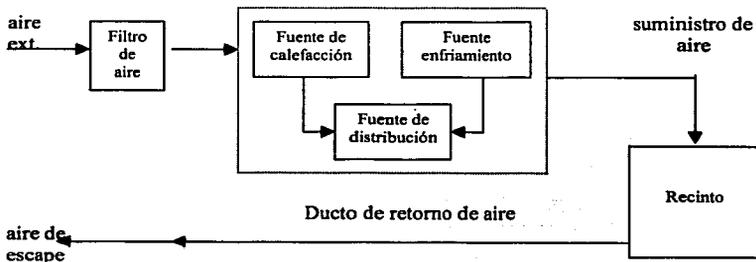


Fig. 2.3 Componentes de un sistema de calefacción y enfriamiento

## 2.5 Planteamiento

Asimismo, la solución que se presenta en este proyecto es una de una diversidad de soluciones que se pueden encontrar en el campo técnico y comercial, ya que muchos compartimientos tienen instalado algún sistema que tenga como objetivo nivelar la temperatura interior a un cierto rango, teniendo como objetivos: confort y condiciones de operación.

TESIS CON  
FALLA DE ORIGEN

La particularidad que presenta este proyecto es el uso de la Lógica Difusa para el sistema automatizado, el cual controlará el nivel de temperatura en el interior del compartimiento.

El sistema constará de dos modos de selección para su funcionamiento: manual y automático, pero ambos tomarán como herramienta, para su funcionamiento, la Lógica Difusa.

¿Por qué utilizar un sistema de control automático?. Un sistema de control automático permite seleccionar la temperatura deseada. Primeramente se fija la temperatura deseada, posteriormente, el sistema ajustará de manera automática la variable temperatura, velocidad del ventilador y la potencia de la resistencia, la cual será la fuente de calefacción.

Los controles automáticos para un sistema de calefacción y acondicionamiento de aire, se puede comparar, en funcionamiento e importancia, con el sistema nervioso y el cerebro de un ser humano.

El diseño e instalación de los controles automáticos para un sistema de acondicionamiento de aire deben ser compatibles con el mismo sistema; asimismo deben trabajar en forma correcta. De no ser así, el sistema no producirá condiciones satisfactorias.

Los controles automáticos deben mantener las condiciones de diseño en el compartimiento. Dependiendo de dichas condiciones, se seleccionará la capacidad de calefacción y enfriamiento del sistema. Además, gracias a los controles automáticos la intervención humana se reduce considerablemente, dentro del funcionamiento y servicio del sistema, esto reduce costos y posibilidades de errores. Una de las consideraciones más importantes que surgen al hacer uso de un sistema de control automático es que éste tiene la capacidad para reducir el uso de energía.

El sistema que se expone está integrado de la siguiente forma:

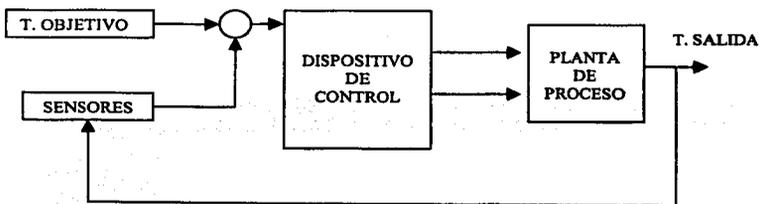


Fig. 2.4 Diagrama de bloques del sistema de control propuesto

TESIS CON  
FALLA DE ORIGEN

De acuerdo a la figura 2.4 se tiene, en la entrada del sistema, la variable a controlar que es la temperatura. Dicha variable será sensada en el interior y exterior del compartimiento.

Se tiene el dispositivo de control, el cual percibe los diversos cambios en la variable controlada y transmite una acción a un dispositivo que se encargará de corregir la condición que cambió.

El otro dispositivo mencionado es la planta de proceso, llamada también etapa de potencia. Este dispositivo se encargará de percibir la regulación del sistema de control, con la cual se varía la capacidad u operación de la misma planta.

Al final del proceso, se tendrá como salida el cambio deseado en la variable, pero sólo en un instante. Esta salida se retroalimenta al sistema para compararla con otra muestra de la variable controlada en otro instante de tiempo. Por lo tanto, la característica del control propuesto es proporcional derivativa, la variable a controlar tiende a un determinado valor en un cierto rango de tiempo.

Por otro lado, para establecer las condiciones necesarias y específicas que requiere el sistema de control para su funcionalidad, se hace uso de las herramientas matemáticas que hasta el momento se exponen.

Para realizar un intercambio de temperatura con el cual se pueda adquirir o disipar calor, el proceso realizará una convección forzada con el aire que entre al sistema.

Ahora, para poder transmitir calor hacia el interior del compartimiento, es necesario hacer uso de un actuador que realice dicho trabajo en un intervalo de tiempo considerable.

Para poder considerar las características de dicho actuador, se tomará como base la ecuación 2.b :

$$Q = mC(T_1 - T_2)$$

Inicialmente, considérese una variación de temperatura  $\Delta T$  en un intervalo de tiempo, es decir el objetivo del control es llegar a obtener una temperatura deseada a partir de un valor de referencia, por ejemplo, podemos tener una temperatura ambiente de  $14^\circ \text{C}$  y se desea tener una temperatura de  $21^\circ \text{C}$  después de un tiempo determinado. De esta manera , tenemos un  $\Delta T$  definido como  $\Delta T = T_2 - T_1$  , donde  $T_2 = 21^\circ \text{C}$  y  $T_1 = 14^\circ \text{C}$  .

La constante  $C$  corresponde al calor específico del aire, ya que lo que estará entrando o saliendo al sistema será el mismo aire a diferentes grados de temperatura.

En lo que se refiere a la variable  $m$ , corresponderá al gasto másico que pueda mantener el sistema, ya que es necesario tener conocimiento de que cantidad de aire entra al sistema en cierto intervalo de tiempo.

Para determinar esta constante fue necesario auxiliarse de un medidor de flujo, de esta manera y considerando la lectura obtenida en el mismo, se determinó el gasto másico. En la figura 2.5 se ilustra el tipo de medidor de flujo que auxilió a esta parte del proceso experimental.

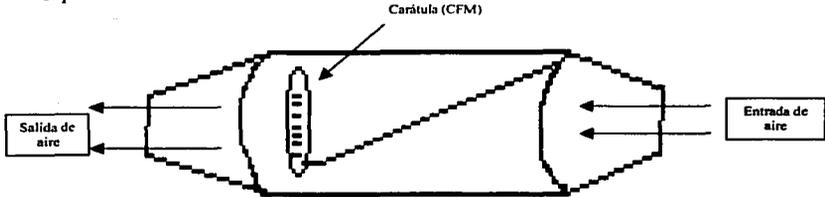


Fig. 2.5 Medidor de flujo de aire

Haciendo uso de un ventilador, se genera el flujo de aire, el cual se hace pasar por el medidor, éste marcará un valor en pies cúbicos por minuto (CFM).

El valor leído, una vez realizada la prueba, fue de  $15 \text{ ft}^3 / \text{min}$ .

Tenemos la siguiente equivalencia:  $1 \text{ CFM} = 1.7 \text{ m}^3 / \text{h}$ .

De esta manera, realizando la conversión:  $15 \text{ CFM} = 25.5 \text{ m}^3 / \text{h}$ , este valor equivale al gasto volumétrico que tiene el sistema.

Pero, para poder obtener el gasto másico ( $g_m = m$ ), el cual es necesario para poder obtener los datos completos de la ecuación 2.b, se multiplica el valor obtenido anteriormente por la densidad del aire  $\rho = 1.29 \text{ kg} / \text{m}^3$ :

$$g_m = m = (1.29 \text{ kg} / \text{m}^3) (25.5 \text{ m}^3 / \text{h}) = 32.86 \text{ Kg} / \text{h}$$

De esta manera, tenemos que pasan  $32.86 \text{ Kg}$  de aire por cada hora.

Realizando la conversión:

$$(32.86 \text{ Kg} / \text{h}) (2.2 \text{ lb} / 1 \text{ kg}) = 72.292 \text{ lb} / \text{h}$$

Por otro lado, el calor específico correspondiente al aire tiene un valor de  $C_{esp, a} = 0.24 \text{ BTU} / \text{lb} \text{ } ^\circ\text{F}$ .

Suponga que se tiene  $0^\circ \text{ C}$  y se desea tener  $20^\circ \text{ C}$ , para lograr lo anterior es necesario considerar cierta potencia del actuador encargado de proporcionar energía calorífica, pero si esto se requiere en un tiempo menor a una hora, dicha potencia deberá variar considerablemente.

TESIS CON  
FALLA DE ORIGEN

Consideremos una variación de temperatura de  $\Delta T = 30^\circ \text{C}$  en una hora. Aún sabiendo que el comportamiento no es lineal, se puede aceptar el grado de respuesta que proporcione el actuador una vez teniendo la máxima potencia, ya que la variación de  $0^\circ$  a  $20^\circ \text{C}$  podrá obtenerse en un intervalo de tiempo menor a una hora.

Ya establecidos los parámetros que requiere la ecuación 2.b, se procede a realizar el proceso de cálculo para obtener la disipación de energía por unidad de tiempo que permita determinar las características del actuador a utilizar.

Tenemos:

$$Q = mC\Delta T$$

donde:  $m = 72.292 \text{ lb / h}$   
 $C_{\text{esp a}} = 0.24 \text{ BTU / lb }^\circ\text{F}$   
 $\Delta T = 30^\circ \text{C}$ ,  $\Delta T = 111.6^\circ \text{F}$

así:

$$Q = (72.292 \text{ lb / h}) (0.24 \text{ BTU / lb }^\circ\text{F}) (111.6^\circ \text{F})$$
$$Q = 1936.48 \text{ BTU / h}$$

pero,  $1 \text{ BTU} = 1055 \text{ J}$ , entonces:

$$Q = (1936.48 \text{ BTU / h}) (1055 \text{ J / 1BTU}) (1\text{h} / 3600 \text{ s})$$
$$Q = 567.5 \text{ J / s} = 567.5 \text{ W}$$

De esta manera, el actuador que proporcionará energía calorífica será una resistencia que tenga una potencia de aproximadamente 500 Watts.

En el capítulo cuatro y cinco se mencionará todo acerca de cómo está constituida físicamente cada parte del sistema. En el capítulo cuatro corresponderá mencionar todo lo correspondiente al hardware del sistema, se hace mención de toda la estructura física, tanto de la parte de sensado, como de la parte de control y la etapa de potencia, se abunda en los principios que se adoptaron para la realización de cada etapa y los pasos que se siguieron para su respectivo desarrollo.

En el capítulo cinco corresponderá mencionar todo acerca del software del sistema, su estructura y desarrollo, los principios que se siguieron y el resultado que se obtuvo una vez finalizado.



## CAPITULO 3 CONTROL DIFUSO

### 3.1 PRINCIPIOS DE CONTROL

El diseño del control de temperatura se desarrolla con un arreglo de componentes físicos, organizados de tal forma que el funcionamiento del mismo puede alterar, regular o mandar, a través de una acción de control, a otro sistema físico para que éste exhiba ciertas características o comportamientos deseados.

Por otro lado, de acuerdo a la teoría de control, un control de lazo abierto es aquel en el cual la acción de control es independiente de la salida del sistema físico. Sin embargo, en un sistema de control de lazo cerrado (también conocido como un sistema de control retroalimentado), la acción de control depende de la salida del sistema físico.

Un tostador donde la cantidad de calor es determinada manualmente o una máquina de lavado en la cual la temperatura y el tiempo de ciclo de centrifugado son determinados de igual forma, son ejemplos de sistema de control de lazo abierto, donde la acción de control no es una función de la salida.

El termostato de un cuarto, el cual sensa la temperatura de un compartimiento y activa una unidad de enfriamiento o calentamiento cuando un cierto umbral de temperatura es alcanzado o, un mecanismo de pilotaje automático, el cual realiza correcciones del curso de un aeroplano cuando son sensados valores de algunos parámetros por instrumentos del mismo, son ejemplos de sistemas de control retroalimentado.

De esta manera, las ventajas que proporciona un control de lazo cerrado permite desarrollar un control más versátil que se adapte a las condiciones deseadas por el diseñador.

Si el objetivo de un sistema de control es mantener una variable física en un valor constante ante la presencia de disturbios, el sistema es de tipo regulatorio o llamado regulador. El control de temperatura de un cuarto y el pilotaje automático son ejemplos de controladores regulados.

Sin embargo, existen controladores en los cuales se requiere que la variable física siga alguna función de tiempo deseada. Por ejemplo, un sistema automático de aterrizaje, en el cual se requiere seguir una rampa para llegar al punto deseado.

La salida o respuesta del sistema de control es ajustada por una señal de error. Esta señal de error es la diferencia que existe entre la respuesta actual del sistema y la señal de entrada que proporciona el sensor, pero cabe mencionar que la respuesta deseada está especificada por una entrada de referencia.

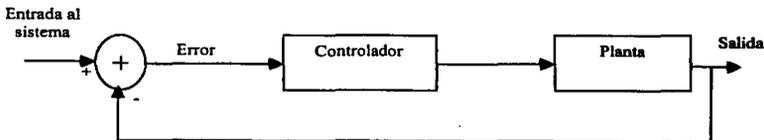


Fig. 3.1 Sistema de control de lazo cerrado

TESIS CON  
FALLA DE ORIGEN

Para controlar una variable física, primeramente se debe medir. De esta manera, inicialmente se tiene un dispositivo que mida dicha variable llamado sensor; una parte posterior al sensor es la planta. En un sistema de control de lazo cerrado, ciertas señales del sistema, específicamente las señales de entrada, son determinadas por la respuesta del sistema o señales de salida. Al conectar un compensador o controlador dentro del lazo se pueden obtener resultados y características satisfactorias durante el control.

Es indiscutible que la teoría de control es extensa y, conforme aumenta la estructura y funcionalidad del sistema, la complejidad del control y todo lo que él conlleva, aumenta su desarrollo hasta llegar a un máximo de complejidad que origine un sistema de control poco práctico y costeable.

Sin embargo, el objetivo es realizar un sistema de control, donde se mantenga la temperatura dentro de los parámetros deseados en el interior de un recinto y que base su desarrollo en los principios expuestos por la teoría difusa y sus aplicaciones al control de procesos, la cual puede brindar, en diversas ocasiones, una menor complejidad a la que pueda existir al considerar un control clásico.

### 3.2 LÓGICA DIFUSA

En el año de 1965 el Dr. Lotfi A. Zadeh dio a conocer el primer documento sobre la teoría de Lógica Difusa (Fuzzy Logic) siendo el pionero de esta teoría. Puede decirse que la lógica difusa es otra forma de pensar, ya que a diferencia de la lógica clásica o aristotélica donde el todo se puede clasificar como falso ó verdadero, en Lógica Difusa se puede tener diferentes grados de pertenencia, es decir, ya no se tiene la restricción de los valores falso o verdadero, sino que se pueden tener varios grados como muy falso, poco falso, poco cierto, muy cierto, etc. Numéricamente, la representación de grados de pertenencia sería como: 0.8 falso, 0.2 falso, 0.2 cierto, 0.8 cierto, etc.

Si comparamos la representación de los conjuntos en un diagrama de Venn-Euler sería como se muestra en la fig. 3.2:

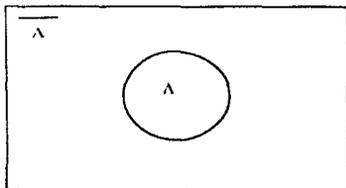


FIG. 3.2(a)

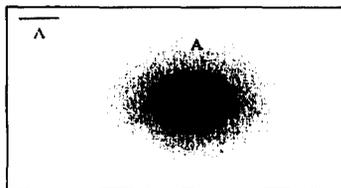


FIG. 3.2 (b)

Fig. 3.2 Representación Euler de conjunto crisp y difuso.

TESIS CON  
FALLA DE ORIGEN

En la figura 3.2(a) se muestra la representación de los conjuntos "Crisp" o aristotélicos, donde se observa que el contorno de los conjuntos están definidos y se nota con claridad la división entre conjuntos, mientras que en la fig 3.2(b) se esquematiza a los conjuntos difusos, donde se observa que la división de estos no están bien definida.

Se observa que el centro de los conjuntos es más oscuro esto es al grado de pertenencia de los elementos, cabe mencionar que el grado de pertenencia es complementario, es decir, mientras un elemento pertenece más a un conjunto pertenece menos al otro. Esto último se puede representar numéricamente, por ejemplo, si tiene un grado de pertenencia de 0.75 falso tendrá el 0.25 de verdadero.

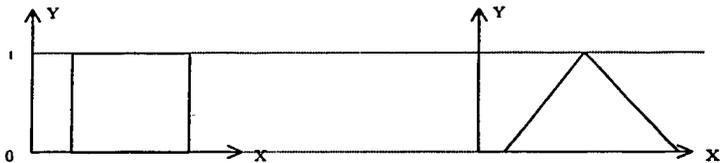


Fig. 3.3(a)

Fig. 3.3(b)

Fig. 3.3 Representación cartesiana de conjuntos crisp y difusos.

Otra forma de representar a los conjuntos es por medio de planos cartesianos. En la figura 3.3(a) se observa la representación de un conjunto "Crisp" en donde el falso (cuya representación numérica es de un "0") pasa al verdadero (que se puede representar por un "1") de manera abrupta, mientras que en la figura 3.3 (b) se puede ver a la representación de un conjunto difuso, donde la transición de falso a verdadero y viceversa se desarrolla por medio de una curva, la cual es una línea donde el cambio es paulatino. Cabe mencionar que los conjuntos "Crisp" se pueden considerar como una situación especial de un conjunto difuso, es decir, la lógica difusa abarca a la lógica aristotélica. Una de las características de los conjuntos difusos es que no cumplen con las propiedades de la Ley del Medio Excluido y Ley de la Contradicción.

La gráfica que se muestra en la figura 3.3 (b) es llamada función de membresía y representa el valor correspondiente de una variable X a un valor difuso Y. A Y también se le llama  $\mu(x)$ . La forma del conjunto difuso representada en la figura 3.3(b) no solamente es triangular (tipo  $\lambda$ ) sino que puede obtener diferentes formas tales como Gaussiana, Singleton, tipo Z, tipo S, tipo  $\eta$ .

Un conjunto difuso tiene partes que se identifican como lo muestra la figura 3.4.

TESIS CON  
 FALLA DE ORIGEN

Control de temperatura con lógica difusa

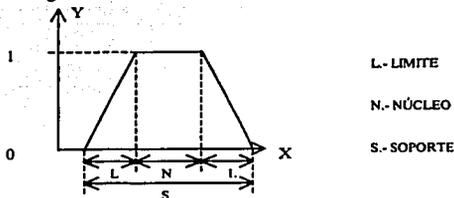


Fig. 3.4 Partes que identifican a un conjunto difuso.

La álgebra de los conjuntos difusos es similar a la de los conjuntos Crisp, ya que comparten muchas de sus respectivas propiedades diferenciándose por las dos leyes antes mencionadas, pero cabe mencionar que sus representaciones pueden variar, por ejemplo un conjunto difuso se representa por letras mayúsculas, al igual que un conjunto Crisp, pero sobre una línea curvada y su definición estaría representada como sigue:

$$\overset{\sim}{A}(x) = \{x_i / 0 \leq \mu(x_i) \leq 1\} \text{-----}(3.1)$$

La representación del complemento, sería de la siguiente forma:

$$\overline{\overset{\sim}{A}} = \mu_{\overset{\sim}{N}}(x) = 1 - \mu_{\overset{\sim}{N}}(x) \text{-----}(3.2)$$

Para poder trabajar con estos conjuntos debe ser considerada la forma en que se aplican los operadores de unión, intersección y complemento. Este último se representó en la ecuación (3.2) cuya interpretación sería: si se tiene un determinado valor difuso, por ejemplo, 0.35 el complemento será lo que falte para un valor de uno, (0.65).

El operador unión puede visualizarse si consideramos una tabla binaria, ya que la unión es semejante al operador "OR" como se muestra en la tabla 3.1. Si se observa con detalle el resultado es el mayor (1) de los valores, de esta manera, podemos decir que el resultado de la unión de dos conjuntos serán los elementos con mayor valor de pertenencia, y su representación es la siguiente:

$$\mu_{\overset{\sim}{A} \cup \overset{\sim}{B}}(x) = \max(\mu_{\overset{\sim}{A}}(x), \mu_{\overset{\sim}{B}}(x)) \text{-----}(3.3)$$

A	B	C = (AUB)
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 3.1

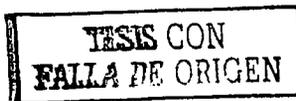
TESIS CON  
 FALLA DE ORIGEN

Para la intersección al igual que en la unión puede interpretarse a partir de una tabla binaria, donde la operación de intersección es semejante a la "AND". A partir de la tabla 3.2 se deduce que el resultado sigue al valor menor de la tabla, de esta manera, la solución de la intersección son los elementos con menor grado de pertenencia y su representación es la siguiente:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \text{-----(3.4)}$$

A	B	C = (A∩B)
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 3.2



A partir de estos operadores y de sus propiedades pueden realizarse prácticamente todas las operaciones entre conjuntos difusos como son las relaciones, composiciones y otras.

De esta manera, se visualiza la facilidad de la lógica difusa en diferentes áreas y con ello tener la posibilidad de desarrollar sistemas más rápidos, más robustos y de menor costo, tanto para los requerimientos de cálculo, de software y en algunos casos de hardware.

Con la lógica difusa es posible manejar un lenguaje más sencillo, es decir, menos rígido. Por ejemplo: al referirse al parámetro temperatura, ya no es necesario que tome un valor rígido de 5 grados centígrados, sino que puede ubicarse en una etiqueta lingüística llamada temperatura fría con un cierto grado de pertenencia, de igual forma 40 grados centígrados se podría ubicar en otra etiqueta llamada temperatura caliente, también con un cierto grado de pertenencia.

Para pasar una variable a un valor difuso (fuzzificación) se tienen diferentes métodos, estos métodos dan el valor de pertenencia de la variable a los diferentes conjuntos difusos. Estos métodos pueden ser: Intuición, Inferencia, Ordenación por Rango, Conjuntos Difusos, Redes Neuronales, Algoritmos Genéticos, Razonamiento Inductivo.

Para la aplicación expuesta se utilizan conjuntos difusos cuyos puntos son determinados. Con los valores difusos se evalúan las reglas que son las que determinan el comportamiento del sistema, al evaluar realizan una modulación de los conjuntos de salida y a partir de ésta modulación se origina la salida difusa para la cual se tiene diferentes métodos de desfuzzificación como son: Principio del Máximo, Centroide, Promedio Pesado, Promedio de Máximos, Centro de Sumas, Centro del Área Mayor, Primero o Último de Máximos, Singletons.

Para conocer la estructura de un control difuso se tiene el siguiente diagrama en el cual se muestran todos los bloques que intervienen en el proceso.

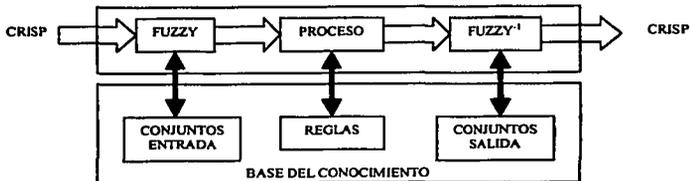


Fig. 3.5 Diagrama s bloques de un control difuso.

El bloque FUZZY junto con el bloque CONJUNTOS DE ENTRADA se encarga de pasar un valor físico a un valor difuso, este valor depende básicamente de cómo estén conformados los conjuntos de entrada (tipo, soporte, límites, núcleo). El siguiente bloque participante es el de PROCESO que junto con el bloque de REGLAS evalúan las condiciones establecidas en el diseño de un sistema y de esta forma se obtienen los primeros resultados, los cuales pasan a los últimos bloques que son los de FUZZY<sup>-1</sup> y al de CONJUNTOS DE SALIDA, los cuales se encargan de evaluar los resultados preliminares junto con los conjuntos de salida para, finalmente, dar un el valor nítido de salida, pero este valor deberá ser congruente con los actuadores de control. Esta es la forma simplificada de como trabaja un sistema de control difuso.

A continuación se expone un ejemplo que permita ilustrar, con más claridad, el proceso de un control difuso. La forma de proceder es la siguiente: se definen los conjuntos difusos de tal manera que sean congruentes con las variables a utilizar, por ejemplo: Poco Caliente (0 °C a 20 °C), Caliente (10 °C a 30 °C) y Muy Caliente (20 °C a 40 °C). La representación de los conjuntos se ilustra en la figura 3.6.

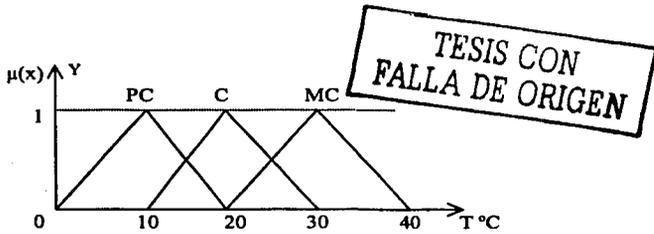


Fig. 3.6 Representación de conjuntos difusos de entrada

donde

PC = Poco Caliente

C = Caliente  
MC = Muy Caliente

Ahora definimos de forma similar nuestros conjuntos de salida, donde la variable será el ciclo de trabajo de la resistencia. Los conjuntos definidos son Poco Intensa, Intensa, Muy Intensa. Ver fig. 3.7.

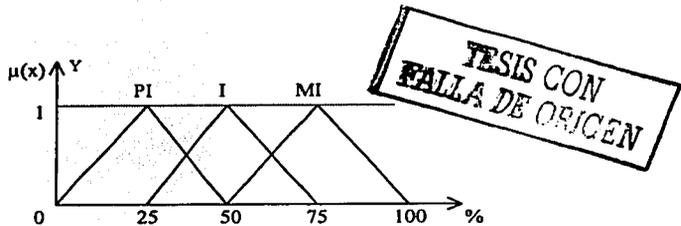


Fig. 3.7 Representación de conjuntos difusos de salida

donde

PI = Poco Intenso  
I = Intenso  
MI = Muy Intenso

Teniendo ya definidos los conjuntos de entrada y de salida se procede a realizar las reglas. Estas reglas se basan en los Sistemas Basados en Reglas y una regla lingüística esta formada por juicios de asignación, juicios de condición y juicios incondicionales, es decir, la estructura de la regla es de la siguiente forma: IF  $A^S$  THEN  $B^S$ , donde  $A^S$  representa a los conjuntos de entrada y  $B^S$  los conjuntos de salida.

Como se menciona con anterioridad, la relación que se da entre la entrada y la salida es por medio de reglas, por ejemplo: si es poco caliente entonces la potencia es muy intensa, si es muy caliente entonces la potencia es poco intensa, etc. El sistema se encarga, por medio de algún método de fuzzificación, de pasar la variable física a un valor difuso, esto se realiza por medio de los conjuntos de entrada, posteriormente se evalúan las reglas que proporcionan valores difusos de salida, los cuales son evaluados a través de los conjuntos de salida y por último mediante un método de desfuzzificación se obtiene la salida nítida correspondiente, para esto el sistema cuenta con un Kernel, quien se encarga de decodificar y manejar la información entre las variables difusas, los conjuntos de entrada, la evaluación de reglas y el correspondiente conjunto de salida.

El Kernel es una parte medular del sistema, ya que es el encargado de llevar la lógica difusa a la plataforma a utilizar. El ejemplo anterior tiene una entrada con su conjunto difuso definido y una salida también con su conjunto difuso definido por lo que las reglas son relativamente pocas, pero conforme el número de entradas aumenten las

## Control de temperatura con lógica difusa

reglas aumentan de tal forma que el Kernel deberá ser lo suficientemente robusto para manejar el sistema.

El capítulo 5 se encargará precisamente de la constitución del Kernel y del software en general.

Continuando con la solución del ejercicio se debe tener las reglas definidas para poder llevar a cabo el control, para esto se tienen las siguientes reglas:

Si  $x$  es PC entonces MI

Si  $x$  es C entonces I

Si  $x$  es MC entonces PI

Puesto que es un ejemplo con una sola entrada y una sola salida las reglas resultan muy sencillas y fáciles de definir. Teniendo ya las reglas definidas se tiene completo el sistema de control, ahora sólo se tiene que evaluar las diferentes entradas al sistema, para ello se considera que en un instante de tiempo la entrada  $x$  es de  $13^{\circ}\text{C}$ . Se debe recordar que es un control continuo, por lo que se tienen que evaluar muchos valores en un periodo corto de tiempo, pero solo se considera un valor para simplificar el ejercicio. Primero se transfiere el valor "CRISP" a un valor "Difuso" como se muestra en la figura 3.8.

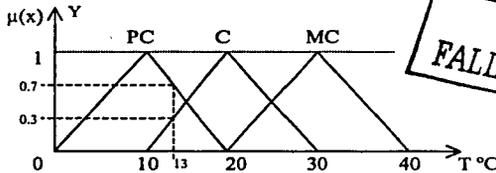


Fig. 3.8

Como se puede observar en la figura 3.9, los conjuntos habilitados son PC y C por lo que tenemos dos valores difusos, uno para cada conjunto. Ahora, evaluando las reglas se tiene que para la primera corresponde al conjunto de salida MI, mientras que para la segunda corresponde I, el tercer conjunto no se habilita, es decir tendrá un valor cero, esto se puede ver más claramente en la figura 3.9.

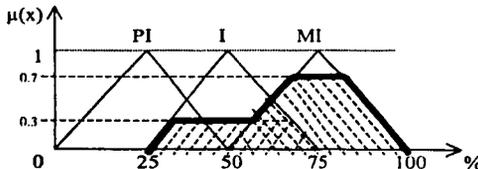


Fig. 3.9

Si aislamos el resultado quedará como se indica en la figura 3.10.

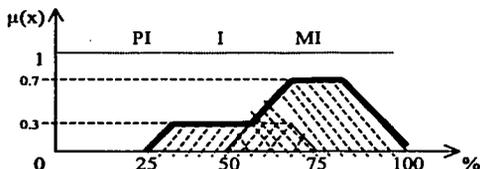


Fig. 3.10

La solución final de este problema se obtiene al aplicar un método de desfusión y de este modo se obtiene el ciclo de trabajo a la cual deberá operar el actuador. Cabe mencionar que el método de desfusión se selecciona a partir de la respuesta requerida al sistema, por ejemplo, que tan exacto debe ser, que tan rápido debe responder, etc. Para muchos sistemas el método de SINGLETONS es suficiente para obtener resultados bastante aceptables, además es el método que requiere, hasta cierto punto, menor requerimiento de software y calculo, lo que lo hace más rápido y más sencillo de manejar. Retomando el ejemplo, primeramente debe definirse los Singletons de la salida tal como muestra la figura 3.11. Como se puede observar, la representación consisten en unas flechas que apunta al valor asignado y dependiendo de este valor es como responderá la salida, es decir, al hacer cualquier cambio en los Singletons existirá un cambio en la salida. Se colocaron también los conjuntos anteriores con una línea punteada solo para mostrar la diferencia que existe.

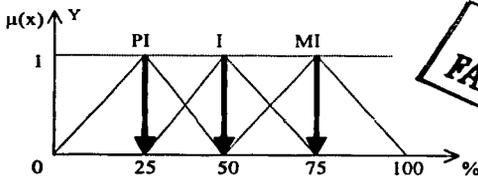


Fig. 3.11 Singletons

TESIS CON  
 FALLA DE ORIGEN

Procediendo a evaluar reglas, se tiene respuesta mostrada en la figura 3.12

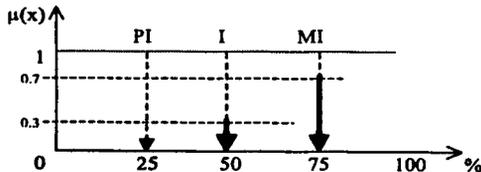


Fig. 3.12

Al aplicar la ecuación 3.5 se obtiene el resultado buscado, el cual puede ser ilustrado en la figura 3.13.

$$Z^* = \frac{\sum \mu(z)_i \cdot z_i}{\sum \mu(z)_i} \text{-----(3.5)}$$

$$Z^* = ((0.3 \cdot 50) + (0.7 \cdot 75)) / (0.3 + 0.7)$$

$$Z^* = 67.5 \%$$

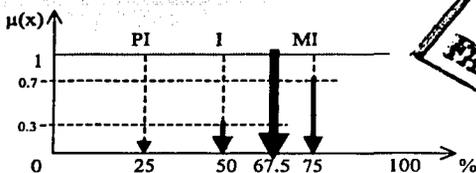
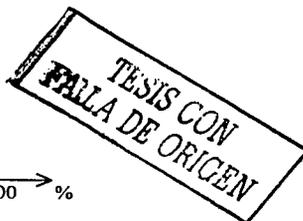


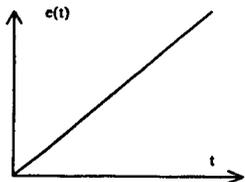
Fig. 3.13



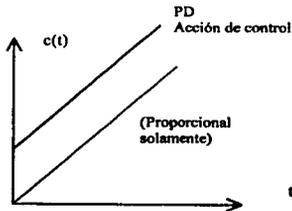
### 3.3 PLANTEAMIENTO DEL CONTROL DIFUSO

Es conveniente establecer un orden para el planteamiento del diseño del sistema de control difuso que permita abordar, de una forma coherente, la solución del problema.

Primeramente, cabe mencionar que la acción de control a desarrollar tendrá característica derivativa, es decir, considérese el valor de salida del control proporcional a la velocidad de variación de la señal de error actuante (sistema retroalimentado). Así, la acción del control derivativa presenta características de anticipación. (Ver figura 3.14)



3.14 (a) Entrada: rampa unitaria



3.14 (b) Salida de control: proporcional derivativa

Fig. 3.14 Acción de Control Derivativa

Sin embargo, la acción derivativa nunca puede derivarse a una acción que aún no ha tenido lugar.

En la figura 3.15 se muestra la Máquina de Inferencia Difusa, la cual esquematiza la estructura del control a realizar:

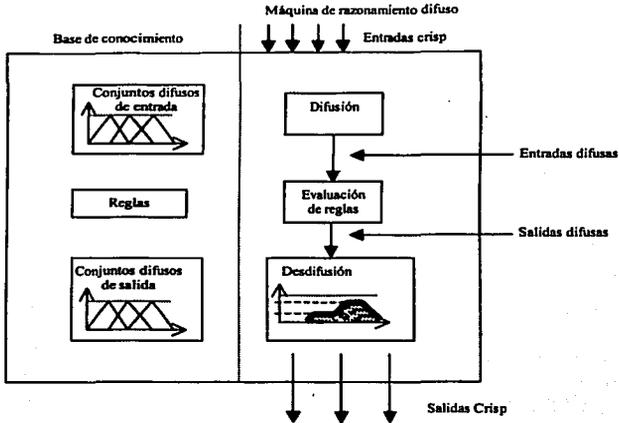


Fig. 3.15 Máquina de Inferencia Difusa

Lo anterior permite identificar las partes difusas del sistema. Hay que observar que partes presentan mayor ambigüedad para el uso de un control difuso y, en caso contrario, apegarse a la metodología de un control clásico.

Una vez identificadas las partes difusas del sistema deben ser señaladas las entradas y salidas del mismo, con el fin de obtener variables de control.

Cabe mencionar que hay que analizar y simplificar el problema planteado, lo que permitirá particionar y resolver de una forma modular, simplificando el trabajo.

De esta manera se definen funciones de membresía de entrada y salida, para ello es necesario especificar el universo para definir cual es el rango de variación de las mismas, así como su comportamiento para establecer de forma correcta los conjuntos difusos.

Una vez establecido el universo se prosigue a particionarlo, dependiendo de las entradas y salidas se junta su escala lineal, logarítmica o cualquier otra.

TESIS CON  
FALLA DE ORIGEN

## Control de temperatura con lógica difusa

De acuerdo a la figura 3.1 la respuesta del sistema de control es ajustada por una señal de error, así primeramente se considera al error como una variable de entrada y estará definido como  $E=T_0-T_f$ , donde:

$E$  = Error

$T_0$  = Temperatura inicial

$T_f$  = Temperatura final

La temperatura inicial es aquella que se sensa una vez que se ha iniciado el control.

La temperatura final es aquella que se presenta después de que el proceso de control ha realizado sus funciones en un intervalo de tiempo dado.

Por otro lado, el sistema basará su funcionalidad en los principios del control proporcional derivativo, por lo cual se define como segunda entrada a la variación con respecto al tiempo de la señal de error actuante que presenta el sistema de control. Es decir:

$E'$  = Velocidad de variación del error => Control proporcional derivativo.

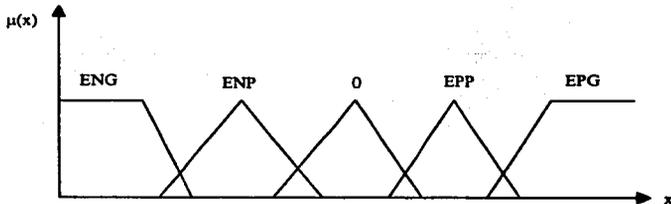
De esta manera, se tiene:

$$E' = \Delta E / \Delta t = \Delta(T_0 - T_f) / \Delta t$$

$$\text{Así, } E' = \Delta E / \Delta t = (E_2 - E_1) / (t_2 - t_1)$$

Se define el universo del discurso y se particiona para establecer los conjuntos correspondientes a las dos entradas.

Se establecen los conjuntos correspondientes a la primera entrada, cuya variable es el error, de la siguiente manera:



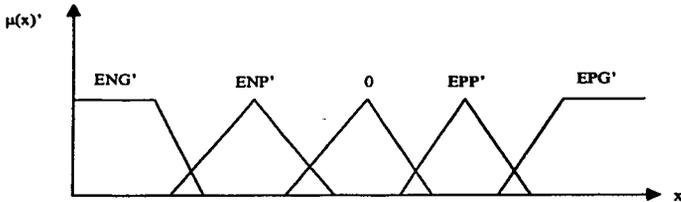
Donde:

$\mu(x)$  corresponde a la función de membresía que representa al error

## Control de temperatura con lógica difusa

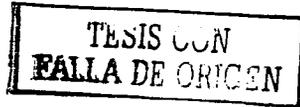
ENG: Error Negativo Grande  
ENP: Error Negativo Pequeño  
0 : Cero  
EPP: Error Positivo Pequeño  
EPG: Error Positivo Grande

Por otro lado, se definen los conjuntos que corresponden a la segunda entrada, cuya variable representa la variación con respecto al tiempo del error.



Donde:

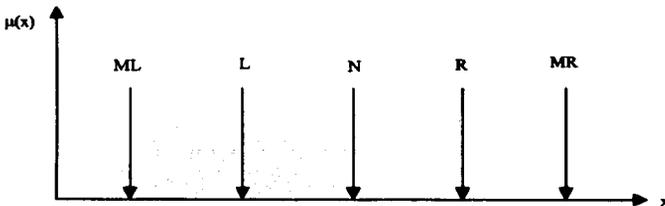
$\mu(x)'$  corresponde a la derivada del error  
ENG': Error Negativo Grande  
ENP': Error Negativo Pequeño  
0 : Cero  
EPP': Error Positivo Pequeño  
EPG': Error Positivo Grande



Las salidas determinadas en el proceso serán aplicadas directamente a los actuadores del sistema y se verán reflejadas en el funcionamiento de un motor y/o una resistencia.

De esta manera, se definen los siguientes conjuntos difusos de salida:

Para el motor, tenemos:

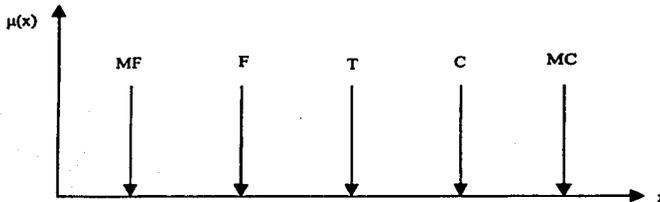


Donde:

ML: Muy Lento  
L : Lento  
N : Nulo  
R : Rápido  
MR: Muy Rápido

Las definiciones lingüísticas corresponden a la velocidad del motor.

Para la resistencia se considera lo siguiente:



Donde:

MF: Muy Fría  
F : Fría  
T : Templada  
C : Caliente  
MC: Muy Caliente



Estas definiciones lingüísticas corresponden a la disipación de energía por unidad de tiempo que presente la resistencia.

Una vez, habiendo establecido los conjuntos difusos de entrada y salida, así como su respectivo universo escalado, se procede a determinar las acciones de control de manera lingüística, es decir, la elaboración de las reglas de control.

En una primera instancia, puede realizarse la escritura de reglas obvias que implican el determinar, a groso modo, el proceso de control.

Una vez que se determinan todas las reglas puede ser exentas algunas de ellas, o bien determinar casos especiales que no resulten tan obvios al implicar criterios contradictorios. Cabe señalar que en diversas ocasiones, la exención de reglas puede realizarse con la ayuda de un experto en el campo donde se desea realizar el control.

De acuerdo a los conjuntos definidos, las reglas propuestas son las siguientes:

Reglas correspondientes a la salida del primer actuador (resistencia):

IF ENG AND ENG' THEN F  
 IF ENG AND ENP' THEN MF  
 IF ENG AND Z' THEN MF  
 IF ENG AND EPP' THEN MF  
 IF ENG AND EPG' THEN MF

IF ENP AND ENG' THEN F  
 IF ENP AND ENP' THEN F  
 IF ENP AND Z' THEN MF  
 IF ENP AND EPP' THEN MF  
 IF ENP AND EPG' THEN MF

IF Z AND ENG' THEN T  
 IF Z AND ENP' THEN F  
 IF Z AND Z' THEN F  
 IF Z AND EPP' THEN MF  
 IF Z AND EPG' THEN MF

IF EPP AND ENG' THEN C  
 IF EPP AND ENP' THEN T  
 IF EPP AND Z' THEN T  
 IF EPP AND EPP' THEN F  
 IF EPP AND EPG' THEN MF

IF EPG AND ENG' THEN MC  
 IF EPG AND ENP' THEN C  
 IF EPG AND Z' THEN C  
 IF EPG AND EPP' THEN T  
 IF EPG AND EPG' THEN F

TESIS CON  
 FALLA DE ORIGEN

La FAM correspondiente a estas reglas es la siguiente:

	ENG	ENP	Z	EPP	EPG
EPG'	MF	MF	MF	MF	F
EPP'	MF	MF	MF	F	T
Z	MF	MF	F	T	C
ENP'	MF	F	F	T	C
ENG'	F	F	T	C	MC

Reglas correspondientes a la salida del segundo actuador (motor):

IF ENG AND ENG' THEN L  
 IF ENG AND ENP' THEN ML  
 IF ENG AND Z' THEN ML  
 IF ENG AND EPP' THEN ML  
 IF ENG AND EPG' THEN ML

IF ENP AND ENG' THEN L  
 IF ENP AND ENP' THEN L  
 IF ENP AND Z' THEN L  
 IF ENP AND EPP' THEN ML  
 IF ENP AND EPG' THEN ML

IF Z AND ENG' THEN N  
 IF Z AND ENP' THEN L  
 IF Z AND Z' THEN L  
 IF Z AND EPP' THEN ML  
 IF Z AND EPG' THEN ML

IF EPP AND ENG' THEN R  
 IF EPP AND ENP' THEN N  
 IF EPP AND Z' THEN N  
 IF EPP AND EPP' THEN L  
 IF EPP AND EPG' THEN ML

IF EPG AND ENG' THEN MR  
 IF EPG AND ENP' THEN R  
 IF EPG AND Z' THEN R  
 IF EPG AND EPP' THEN N  
 IF EPG AND EPG' THEN L

TESIS CON  
FALLA DE ORIGEN

La FAM correspondiente a estas reglas es la siguiente:

	ENG	ENP	Z	EPP	EPG
EPG'	ML	ML	ML	ML	L
EPP'	ML	ML	ML	L	N
Z	ML	ML	L	N	R
ENP'	ML	L	L	N	R
ENG'	L	L	N	R	MR

Una vez que el sistema realiza la evaluación de reglas, se entregarán salidas difusas, las cuales deberán manipularse de acuerdo al proceso de desdifusión para obtener salidas

## **Control de temperatura con lógica difusa**

nítidas que puedan ser captadas por los actuadores. Dicho proceso se realizará basándose en el método de “singletons”.

El kernel del sistema fue desarrollado dentro del rubro correspondiente al software. De esta manera, en el capítulo 5 se expondrá todo lo relacionado al desarrollo de dicho kernel, indicando el método utilizado para su realización.

## 4.1 INTRODUCCIÓN

Actualmente se cuenta con una gran variedad de dispositivos electrónicos con los cuales se pueden desarrollar una gran variedad de sistemas, pero para poder seleccionarlos se necesita, por un lado, conocer las necesidades y alcance del sistema a desarrollar y por otro conocer los diferentes dispositivos electrónicos disponibles. Otro punto importante es el de saber las aplicaciones recomendables de dichos componentes ya que algunos se especializan para aplicaciones destinadas para la investigación y para aplicaciones militares mientras otros son diseñados para trabajar en condiciones que no requieren características tan especiales.

Un aspecto que no se debe perder de vista es el costo beneficio que puede dar un diseño determinado, pues por un lado esta el tamaño físico del sistema contra el precio de los componentes además también se debe tomar en cuenta el número de dispositivos a utilizar y la complejidad del software para llegar al objetivo del sistema.

En la figura 4.1 se muestra un diagrama a bloques del sistema a desarrollar tomando en cuenta el costo beneficio que nos proporciona. En el transcurso del presente capítulo se dará a detalle cada uno de los componentes utilizados.

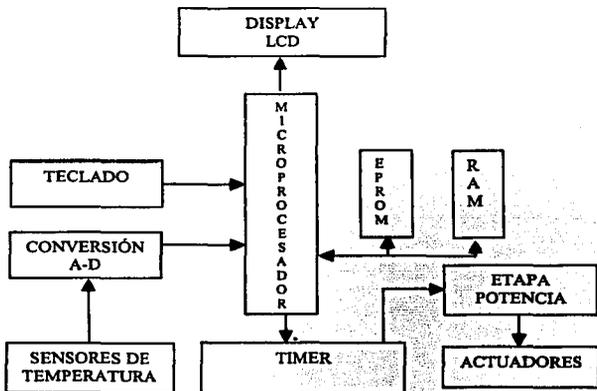


Fig. 4.1 Diagrama a bloques del sistema electrónico.

## 4.2 TECLADO y LCD

Para poder tener algunas opciones de funcionalidad del sistema de control, se cuenta con una interfaz que permita la comunicación del sistema con el usuario y viceversa, para ello se hará uso de un teclado y una pantalla de cristal líquido (LCD).

El teclado cuenta con 4 teclas las cuales tendrán un funcionamiento particular y se encontrarán dispuestos de la siguiente forma:



La tecla SELEC se encarga de desplegar, en el LCD, las diferentes opciones de funcionalidad que tiene el sistema de control por el LCD. La tecla INTRO habilita alguna de la opción elegida en despliegue. Las otras dos teclas restantes se utilizan al seleccionar el modo programa ya que en este modo se tiene la posibilidad de cambiar variables que intervienen en el proceso de control, la tecla "+" incrementará el valor indicado mientras que la tecla "-" lo disminuirá esto se explicará con más detalle en el capítulo siguiente.

Para establecer la comunicación con el teclado es necesario contar con una interfaz de comunicación, esto se logra con un circuito 74HC368, el cual es un buffer inversor con selección. Tal como indica la figura 4.2, tenemos el pin de selección, el cual llamaremos PT y le corresponde el número de puerto 00H.

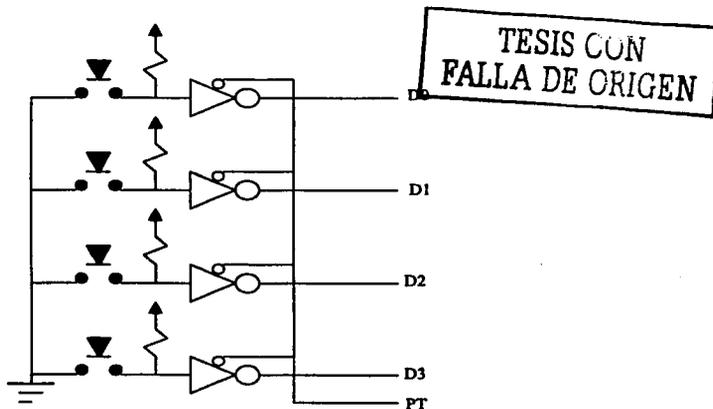


Fig. 4.2 Buffers inversores con selección



un control optimo, es decir, que la frecuencia de corte del filtro paso bajas sea adecuado al sistema y como en este caso las variaciones se dan de manera lenta, se considera un filtro con un ancho de banda  $f_c = 10\text{Hz}$ .

Así tenemos:

Respuesta del sensor de temperatura (LM35):  $10\text{mV}$  por  $1^\circ\text{C}$

Tenemos para  $90^\circ\text{C}$  un valor de  $990\text{mV} = 0.99\text{V}$

La representación matemática de un amplificador no inversor es:

$$V_o = (1 + R_f / R_i) V_i$$

$$V_o / V_i = 1 + R_f / R_i$$

Sustituyendo los valores deseados:

$$R_f / R_i = 5.05 - 1$$

$$R_f / R_i = 4.05$$

Si  $R_f = 33\text{K}\Omega$ , entonces  $R_i = 8.147\text{K}\Omega$

#### 4.4 ADC

Los convertidores analógicos digital se conectan al microprocesador como se indica en la figura 4.5. en donde su conexión es direccionamiento de memoria, es decir, ya no se ocupan como puertos, ya que para ahorrar circuitería los convertidores analógicos digital se conectan en habilitadores de memoria que corresponden a una dirección dada. Para el convertidor 1 la habilitación será llamada ADC1 y para el convertidor 2 será ADC2. En la figura 4.5 se muestra la conexión de un solo ADC ya que son idénticos en la forma de conectarlos variando solamente en la habilitación.

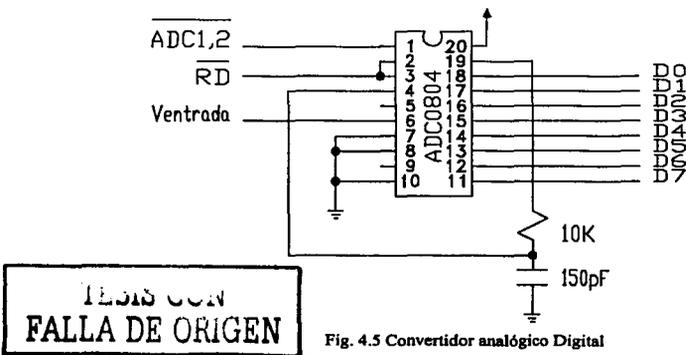


Fig. 4.5 Convertidor analógico Digital

#### 4.5 MICROPROCESADOR

En lo que respecta al microprocesador, como ya se indicó anteriormente, se ocupará el Z-80 en un sistema básico, es decir, tendrá una memoria EPROM, una RAM, su generador de reloj, su reset, y decodificadores para la selección de memorias y puertos. En la figura 4.6 se muestra el diagrama del circuito de reloj y de reset. Estos circuitos son de uso típico por lo que su funcionamiento es conocido.

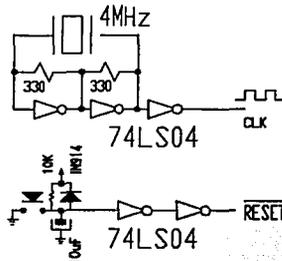


Fig. 4.6 Circuitos de reloj y reset utilizados en el sistema básico

Ahora en la figura 4.7 se ilustra, a nivel de bloques, el sistema básico completo del Z-80.

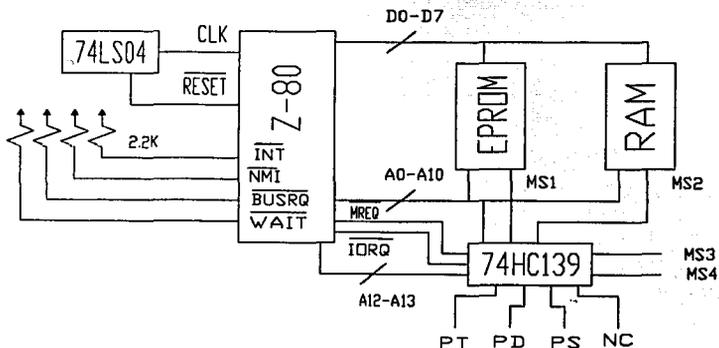


Fig. 4.7 Sistema básico del Z-80

**TESIS CON  
FALLA DE ORIGEN**

Para la selección de puertos y memoria se requiere de un decodificador, el cual se encarga de discriminar la información hacia un lado u otro.

La forma de conectar el decodificador es de gran importancia ya que desde ese instante se determinan los bloques de memoria y números de puertos. Para esto se determina una memoria de programa de 4Kbytes y una memoria de datos de 2Kbytes, así como los puertos con el número que resulta al manejar A6 y A7. Los bloques de memoria que de manejan son, para EPROM de la \$0000 hasta \$0FFF y para RAM desde la \$1000

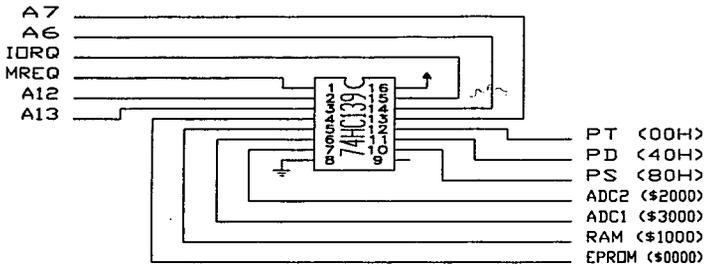


Fig. 4.8 Configuración del decodificador

hasta \$17FF. Para ello se utilizan A12 y A13. La conexión se muestra en la figura 4.8.

#### 4.6 ETAPA DE SALIDA

Por último, se implementa la interfaz de potencia, la cual será la encargada de manejar los actuadores de salida. Los dispositivos que fungirán como tales serán un generador de calor y de un ventilador. Una resistencia será utilizada como generadora de calor, mientras que el ventilador ayudará el proceso de convección forzada y enfriamiento, ambos serán de corriente alterna. Cabe mencionar que el control de salida permitirá manipular la corriente alterna, es decir, su funcionamiento estará basado en la técnica de ángulo de disparo.

La razón por la cual es usada la técnica de ángulo de disparo es por la flexibilidad que presentan las fuentes de corriente alterna en, prácticamente, cualquier lugar, mientras que una corriente continua puede requerir de una mayor estructura que se ve reflejada directamente en los costos del proyecto, además el sentido del mismo retoma un valor más práctico.

Ahora, se considera un TIMER que permitirá manejar los tiempos de operación, es decir, llevará el cálculo de en que momento se deben activar las salidas, ya que se controlan

ambos ciclos de la onda senoidal. Asimismo, se cuenta con un circuito detector de cruce por cero como muestra la figura 4.10. Este último se estructura con un amplificador operacional en un arreglo de comparador como muestra la figura 4.10. La conexión del TIMER se ilustra en la figura 4.9 y como se puede observar se utiliza un 84C53. La peculiaridad de este driver es que es muy flexible y que se adapta en forma óptima a nuestras necesidades, para más información refiérase al apéndice C.

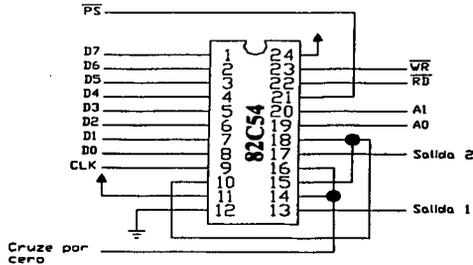


Fig. 4.9 Conexión del TIMER 82C54

Como interfaz de potencia se utilizará un aislamiento óptico, es decir, por medio de un optoaislador/optoaclopador MOC3011 será separado el bloque electrónico de baja potencia con la parte de alta potencia, donde se utilizarán TRIACs como dispositivos de salida tal como muestra la figura 4.11. Las configuraciones tanto del MOC3011 como de los TRIACs MAC218 y MAC223 son las sugeridas o típicas dadas por el fabricante.

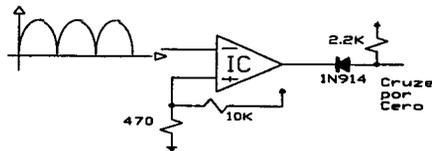


Fig. 4.10 Circuito detector de cruce por cero

**TESIS CON  
FALLA DE ORIGEN**

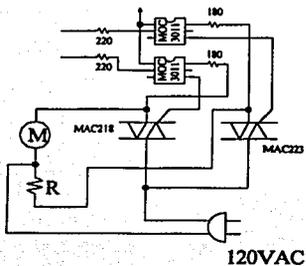


Fig. 4.11 Etapa de alta potencia (salida)

TESIS CON  
FALLA DE ORIGEN

## 5.1 INTRODUCCIÓN

Este capítulo hace referencia al desarrollo del software del sistema. En esta parte se desarrolla la programación del sistema con lógica difusa. La lógica difusa permite utilizar variables lingüísticas para el desarrollo de un sistema de control, del cual se debe tener previo dominio y conocimiento.

Para iniciar con la programación del sistema se consideran dos grandes bloques independientes: el primero será el Kernel de la lógica difusa y el segundo la interfaz con el usuario. Cabe mencionar que estos dos grandes bloques se unirán de tal forma que exista dependencia, pudiendo afectar a las condiciones de trabajo de la lógica difusa a través de la interfaz del usuario.

El diseño del sistema permite exhibir la mayor sencillez, flexibilidad y facilidad de uso, con el fin de tener acceso a varias opciones y parámetros de control.

## 5.2 EL KERNEL DE LA LÓGICA DIFUSA

Una parte importante del control es el Kernel, el cual es desarrollado bajo los principios de la lógica difusa y consta de las siguientes partes:

- a) La difusión
- b) Evaluación de reglas
- c) La desdifusión

Retomando la figura 3.5 del capítulo 3, podemos recordar que la *base del conocimiento* es la parte donde se definen los conjuntos de entrada, las reglas y los conjuntos de salida. Tanto las reglas como los conjuntos de entrada y de salida son definidas a partir del conocimiento que se tenga del proceso a controlar. Cabe mencionar que el Kernel sólo evalúa a partir de la base del conocimiento, por lo que una vez realizado el mismo y se desece una condición de respuesta diferente del sistema, bastará con cambiar las reglas, los conjuntos de entrada, los conjuntos de salida o bien todos.

El Kernel es el núcleo de la Lógica Difusa, por lo que esta formado por un conjunto de programas, los cuales se encargan de desarrollar la difusión de la entrada nítida (creación de valor difuso), la evaluación de las reglas y la desdifusión (proceso de transformación del resultado difuso a un valor reconocido por el actuador de salida). En el diagrama 5.1 se ilustra la estructura del Kernel.

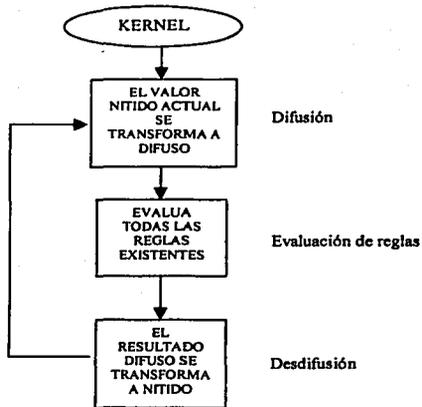
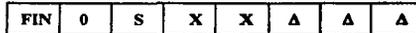


Diagrama 5.1 Descripción General del kernel

### 5.3 DESARROLLO DEL KERNEL

Como parte de la estructura del sistema, se define una palabra de control que establece 4 aspectos como se muestra en la figura 5.1



Donde

FIN. Si es "1" indica el final de las reglas.

S. Si es "1" indica que es salida, si es "0" es entrada.

XX. Indica la entrada o salida correspondiente (va desde 1 a 4).

AAA. Indica la función de membresía en uso (va desde 1 a 8).



Fig. 5.1 Estructura de la palabra de control

La programación de esta parte es descrita posteriormente a detalle, pero cabe mencionar que las reglas se colocan en forma binaria en la memoria, es decir, con base en la palabra de control, se puede establecer a qué variable lingüística corresponde y a través de una operación AND (definida en la teoría de la Lógica Difusa) que se realiza entre ellas, se define el valor de la variable lingüística de salida.

### 5.3.1 MÉTODO DE DIFUSIÓN

El método utilizado para el proceso de difusión es el *punto pendiente*. Para ello es necesario contar con las pendientes ya calculadas de los conjuntos; para facilitar este proceso el sistema calcula las pendientes a partir de cuatro puntos por lo que se pueden manejar conjuntos triangulares y trapezoidales (ver figura 5.2). El cálculo de las pendientes se realiza al iniciar el sistema o bien al cambiar algún parámetro de los conjuntos.

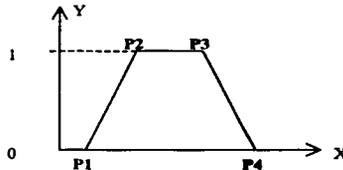


Fig. 5.2 Definición de Puntos

#### 5.3.1.1 CALCULO DE LAS PENDIENTES DE LOS CONJUNTOS

Una vez definidos los cuatro puntos de los conjuntos difusos, se desarrolla el programa que permite realizar el cálculo de las pendientes en forma automática. Inicialmente, se deberán desarrollar algunas operaciones aritméticas importantes, como son la multiplicación y la división, ya que son herramientas que no están implementadas en el set de instrucciones del microprocesador Z - 80.

#### **Multiplicación**

El programa de multiplicación se desarrolla de la siguiente manera: Primeramente se verifica el multiplicador bit a bit de derecha a izquierda, si el valor de dicho bit es "1", se suma el multiplicando al resultado acumulado (estado inicial = cero), se rota el resultado una posición a la derecha; en caso contrario, si el valor del bit en curso es "0", solamente se rotará el resultado una posición a la derecha procediendo a verificar el siguiente bit del multiplicador. El diagrama de flujo de este proceso se ilustra en el diagrama 5.2

TESIS CON  
FALLA DE ORIGEN

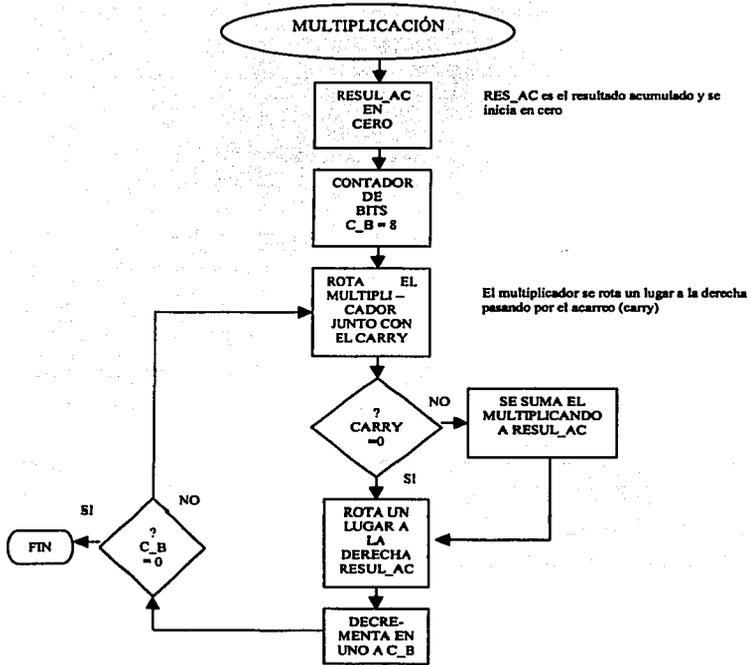


Diagrama 5.2 Descripción de la multiplicación

TESIS CON FALLA DE ORIGEN

Con base en el diagrama de flujo 5.2 se desarrolla la decodificación de la multiplicación; el listado 5.1 expone el ensamblador para esta operación con sus respectivos comentarios.

```

;*****
;PROGRAMA QUE EFECTUA UNA MULTIPLICACION DE 8BITS X 8BITS Y UN RESULTADO
;*****DE 16BITS *****
MULTI:    PUSH AF          ;Se guarda el contenido de AF
          PUSH BC
          PUSH DE
          PUSH HL
          LD DE,00H       ;El resultado acumulado es igual a cero
          LD A,(IX+VAR1L) ;Lugar del multiplicando
          CP 0H           ;Compara si no es cero, si lo es
          JR Z,MULRESUL   ;el resultado es cero
          LD C,A          ;Multiplicando -> C
          LD A,(IX+VAR2L) ;Lugar del multiplicador
          CP 0H           ;Compara si no es cero, si lo es
          JR Z,MULRESUL   ;el resultado es cero
          LD H,A          ;Multiplicador -> H
          LD B,8H         ;C_B = 8 (contador de bit)
MULT1:    RR H             ;Rota multiplicador a la derecha con carry
          JR NC,MULNOSUM  ;Checa el bit en curso
          LD A,D          ;El bit en curso es "1" y se sumar
          ADD A,C         ;al resultado acumulado
          LD D,A
MULNOSUM: RR D             ;Rota el resultado acumulado
          RR E
          DJNZ MUL1      ;Se realizó para los ocho bits?
MULRESUL: LD (IX+VAR3L),E ;EL RESULTADO ACUMULADO ESTA EN DE
          LD (IX+VAR3H),D ;Y SE PASA A UN LUGAR DE MEMORIA
          POP HL         ;Regresa los valores de los registros
          POP DE
          POP BC
          POP AF         ;B,C,D,E,H,L,A,F
          RET           ;Regresa de la multiplicación
    
```

Listado 5.1 Código de la Multiplicación

### División

La división se desarrolla de la siguiente manera: Consiste en introducir el divisor, iniciando por el bit menos significativo, bit a bit, por la izquierda a un registro temporal. Dicho registro es restado al dividendo; si se genera un acarreo, el registro temporal es sumado al resultado de la resta anterior, obteniendo así el minuendo original.

El complemento del acarreo será el primer bit de la estructura del cociente, el cual es guardado en un registro, bit a bit con corrimiento a la izquierda. De esta manera, el primer bit que se origina de la primera substracción será el menos significativo del cociente.



Si no existe acarreo (bit de acarreo = 0), de igual forma se complementa y se introduce ese bit en el registro correspondiente al cociente.

Se continúa con la siguiente sustracción. El número de iteraciones a realizar depende directamente del número de corrimientos del divisor en el temporal. Dichas iteraciones continuarán hasta finalizar los corrimientos del divisor en el registro temporal.

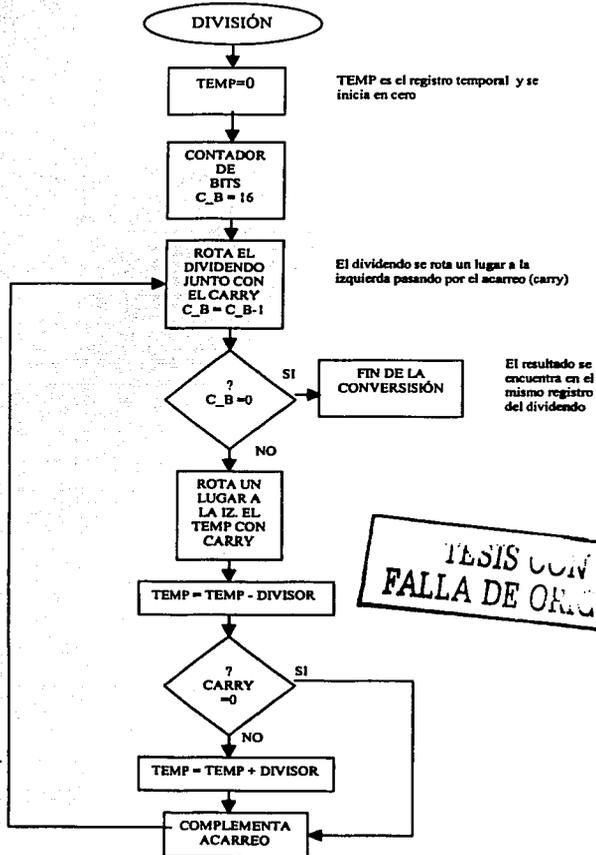


Diagrama 5.3 Descripción de la división

La estructura de la división se ilustra en el diagrama de flujo 5.3 y con base en él, se desarrolla el lenguaje ensamblador para la ejecución de la misma. Véase el listado 5.2.

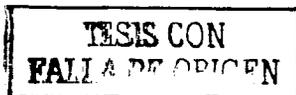
```

;*****
;***PROGRAMA QUE EFECTUA UNA DIVISION ENTERA DE 16BITS X 16BITS Y UN ***
;*****RESULTADO DE 16 BITS *****
DIV:    PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL
        LD HL,00H           ;Usados para guardar el resultado parcial
        LD B,11H           ;Numero de bits m s uno (16 bits + 1)
        LD E,(IX+VAR2L)    ;Obtiene el valor del dividendo
        LD D,(IX+VAR2H)
DIV1:   RL E                ;Rota el valor bajo del dividendo con carry
        RL D                ;Rota el valor alto del dividendo con carry
        DJNZ DIV2          ;? fin de bits
        LD (IX+VAR3L),E    ;Guarda el valor final a
        LD (IX+VAR3H),D    ;la variable VAR3
        POP HL             ;Regresa valores principales
        POP DE
        POP BC
        POP AF
        RET                ;Retorna de la subrutina
DIV2:   RL L                ;Rota el valor del dividendo parcial
        RL H
        LD A,L             ;Recupera el valor del dividendo parcial
        SUB (IX+VAR1L)     ;y réstalo al divisor
        LD L,A
        LD A,H
        SBC A,(IX+VAR1H)
        LD H,A
        JR NC,DIVNOSUM    ;? existe acarreo
        LD A,L             ;Si hay acarreo por lo que
        ADD A,(IX+VAR1L)  ;se regresa al valor inicial
        LD L,A
        LD A,H
        ADC A,(IX+VAR1H)
        LD H,A
DIVNOSUM: CCF              ;Complementa el acarreo
        JR DIV1           ;y repite para el siguiente bit
    
```

Listado 5.2 Código de la división

Una vez que las operaciones básicas han sido programadas, se procede a desarrollar el cálculo de las pendientes. Para ello, se manejan bloques, es decir, considérese el uso de direccionamiento indexado, así como el uso de pares de registros del Z-80.

Inicialmente se definen las localidades donde serán almacenadas las pendientes calculadas y los puntos de los conjuntos; de esta manera, se establece la **CIM1E1**, la cual queda definida como: Conjunto uno, pendiente uno y entrada uno. De igual forma, para los puntos se tiene la **CIPIE1** definida como: Conjunto uno, punto uno y entrada uno. Representando estas localidades en un mapa de memoria quedaría como se ilustra en la figura 5.3.



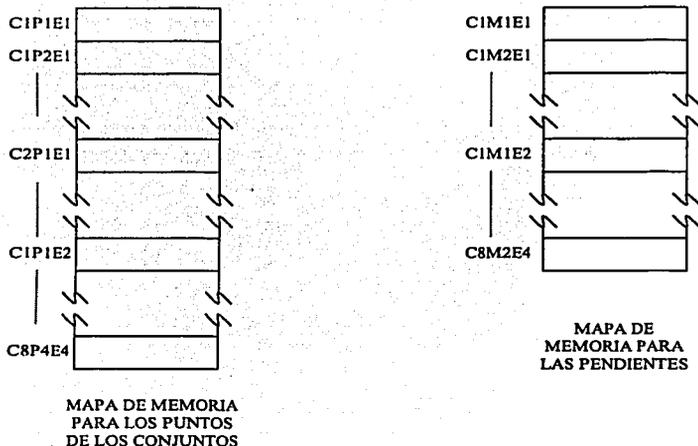


Fig. 5.3 Mapas de memoria de pendientes y puntos de los conjuntos difusos

Se desarrolla el algoritmo que consiste en un ciclo que repetirá el cálculo de las pendientes de ambos lados de un conjunto, es decir,  $M_1 = (P_2 - P_1)/FFH$  y  $M_2 = (P_4 - P_3)/FFH$ . Considerando dos entradas, con una representación de cinco conjuntos por entrada y dos pendientes por conjunto, el ciclo se repetirá 20 veces en decimal o bien 14H en hexadecimal. El diagrama de flujo 5.4 ilustra el desarrollo del algoritmo correspondiente. Cabe mencionar que cuando  $P_1 = P_2$  o  $P_3 = P_4$ , donde el valor es FFH en  $\mu(x)$ , el valor de la pendiente será también FFH, esto para facilitar el cálculo al evaluar  $\mu(x)$ .

TESIS CON  
FALLA DE ORIGEN

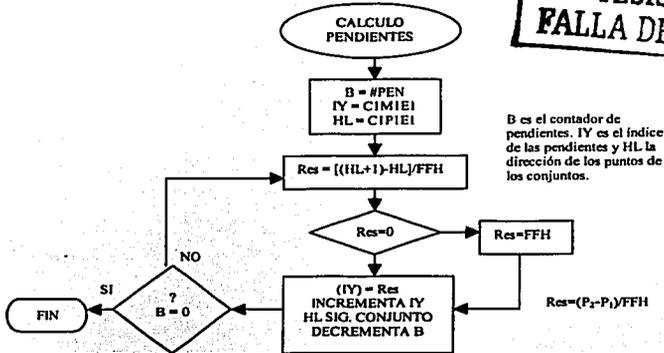


Diagrama 5.4 Diagrama de flujo para el cálculo de las pendientes

Con base en el diagrama anterior, se desarrolla el ensamblador que se ilustra en el listado 5.3.

```

;***** PROGRAMA QUE CALCULA LAS PENDIENTES DE LOS CONJUNTOS *****
PENDIEN:  LD A, (NUMENT)          ;Calcula el número de pendientes que
          RLCA                  ;esta dado por el número de entradas
          LD (IX+VAR1L),A       ;y número de conjuntos.
          LD A, (NUMCON)        ;pendientes = conjuntos*2*entradas
          LD (IX+VAR2L),A
          CALL MULTI
          LD B, (IX+VAR3L)       ;# de pendientes -> B
          LD IY,C1M1E1          ;Lugar donde estarán las pendientes
          LD HL,C1P1E1+1        ;Lugar de los puntos de los conjuntos
          LD (IX+VAR2L),0FFH    ;Inicia los valores de la división
          LD (IX+VAR2H),00H
          LD (IX+VAR1H),00H
LOOP1:   LD A, (HL)              ; A = P2 ÷ P4
          DEC HL                 ; HL se coloca en el punto anterior
          SUB (HL)               ;VALOR DEL DIVIDENDO EN A
          LD (IX+VAR1L),A        ;Pasa el valor a la división
          CALL DIV
          LD A, (IX+VAR3L)       ; A = (P2 - P1)/FF ÷ A = (P4 - P3)/FF
          CP 0H                  ;Compara si los puntos son iguales el
          JR NZ,PENOCERO        ;resultado será FFH
PENOCERO: LD (IY+0),A           ;Se coloca el resultado donde corresponde
          INC HL                 ;-----
          INC HL                 ;Se coloca en el sig. conjunto
          INC HL                 ;-----
          INC IY                 ;Siguiente lugar de pendiente
          DJNZ LOOP1            ;? están todas las pendientes
          RET                    ; Si -> retorna
  
```

Listado 5.3 Código para cálculo de pendientes



El método *punto pendiente* que obtiene el valor difuso correspondiente al valor nítido de entrada se describe a continuación:

- a) se calculan dos incrementos :

$$\Delta_1 = \text{valor nítido} - P_1 \quad \text{y} \quad \Delta_2 = P_2 - \text{valor nítido}$$

Si cualquiera de los dos incrementos es negativo, entonces el valor nítido estará fuera del conjunto, por lo tanto el valor difuso ( $\mu$ ) será igual a cero.

- b) Si los incrementos no son negativos se procede a calcular el producto con las pendientes:

$$\text{Producto 1} = \Delta_1 \cdot S_1$$

$$\text{Producto 2} = \Delta_2 \cdot S_2$$

Donde,  $S_1$  es la pendiente 1 y  $S_2$  es la pendiente 2

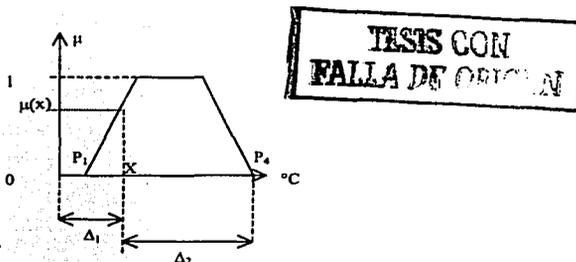
- c) Una vez teniendo los productos se obtiene el mínimo, el cual será el resultado difuso:

$$\mu = \min(\text{Producto 1}, \text{Producto 2})$$

Este proceso se repite para cada uno de los conjuntos de entrada. Al terminar, se tendrán los valores difusos para la entrada considerada. Dado que el sistema es de dos entradas, lo anterior se realiza para ambas. Al terminar con este bloque se desarrolla la evaluación de reglas utilizando la palabra de control descrita en la figura 5.1

#### **Conversión del valor nítido a difuso**

Para obtener un valor difuso a partir del valor nítido se consideran los puntos de los conjuntos y las pendientes. En la figura 5.4 se presenta la gráfica de un conjunto difuso con sus correspondientes puntos y pendientes, dicha gráfica ayuda a representar el proceso de conversión.



Donde:

$$\Delta_1 = (x - P_1)$$

$$\Delta_2 = (P_4 - x)$$

$$\mu(x) = \min(\Delta_1 \cdot M_1, \Delta_2 \cdot M_2, FF)$$

Fig. 5.4 Representación de puntos y pendientes de un conjunto difuso

El proceso de conversión es llamado punto pendiente y consiste en calcular los valores de los incrementos  $\Delta_1$  y  $\Delta_2$ . Si algún valor de incremento resultara negativo esto implica que el valor difuso será nulo ( $\mu = 0$ ), ya que dicho valor de  $x$  no corresponde al conjunto en cuestión. Caso contrario, si los incrementos son positivos se obtienen los productos que resultan de multiplicar a los mismos con sus respectivas pendientes, es decir, al  $\Delta_1$  obtenido a partir del  $P_1$  (punto uno), le corresponde la  $M_1$  y al  $\Delta_2$  la  $M_2$  (ver figura 5.4). El valor difuso  $\mu(x)$  será el correspondiente al mínimo de los productos obtenidos.

$$PROD_1 = \Delta_1 M_1$$

$$PROD_2 = \Delta_2 M_2$$

$$\mu(x) = \min [ PROD_1, PROD_2, FF ]$$

Por otro lado, esta conversión se realiza en un ciclo, ya que es desarrollada para todas las pendientes de todos los conjuntos. El programa se divide en dos partes: La primera describe un ciclo para todas las pendientes de todos los conjuntos y ambas entradas, la segunda sólo describe cómo hacer el cálculo del valor difuso.

Este proceso se ilustra en los diagramas 5.5 a) y b)

**TESIS CON  
FALLA DE ORIGEN**

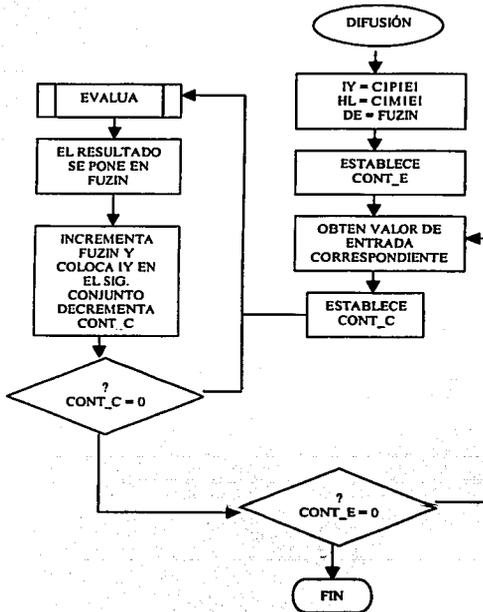
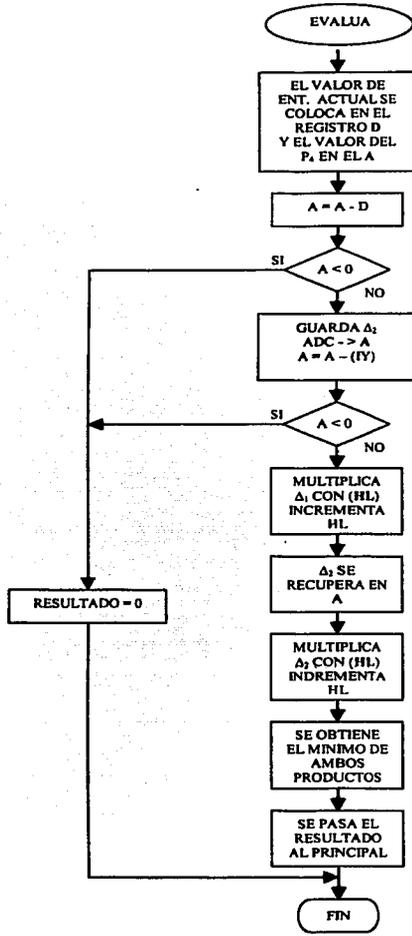


Diagrama 5.5 a) Descripción de la primera parte de conversión (ciclo)

**TESIS CON FALLA DE ORIGEN**



El valor actual de entrada es que se tiene antes de esta rutina. P4 es (IV + 3)

$$\Delta_2 = (P_4 - ADC)$$

CIP1E1 es el lugar de los puntos de los conjuntos.  
CIM1E1 es el lugar de las pendientes.  
FUZIN es el lugar donde se colocaran los resultados de la fusificación.

$$\Delta_1 = (ADC - P_1)$$

CON\_E es el contador de las entradas .

CON\_C es el contador de los conjuntos.

HL tiene la dir. de la pendiente en curso.

Diagrama 5.5 b) Descripción de la segunda parte de conversión (Evaluación)

Una vez descrito el método de conversión con sus respectivos diagramas de flujo, se exponen los listados de ensamblador de ambos algoritmos con sus respectivos comentarios.

```

;***** PROGRAMA QUE EFECTUA LA FUSIFICACIÓN *****
FUZY:      PUSH AF          ;-----
           PUSH BC          ; Retiene valores
           PUSH DE          ; anteriores
           PUSH HL          ;-----
           LD IX,C1P1E1     ;PUNTO 1 DEL CONJUNTO 1 Y ENTRADA 1
           LD HL,C1M1E1     ;Lugar de las pendientes calculadas
           LD DE,FUZIN      ;Lugar en donde se pone el valor difuso
           EXX              ;HL' Apuntará hacia el BLOQUE DE ENTRADAS
           LD HL,BLKENT     ;
           EXX              ;
           LD A,(NUMENT)    ;
           LD C,A           ; # de entradas
FUZYB:     EXX
           LD A,(HL)        ;
           INC HL           ;
           EXX              ;
           LD (IX+GEN1),A   ;Variable que utiliza valúa para calcular
           LD A,(NUMCON)   ;
           LD B,A           ;# de conjuntos
FUZYVAL:   CALL EVALUA     ;Función que realiza el calculo
           LD (DE),A        ;En A se encuentra el Min(P1,P2,FF)
           INC DE           ;lo pone en FUZIN e incrementa el lugar
           INC IY           ;-----
           INC IY           ;IY se ubica en el conjunto
           INC IY           ;siguiente
           INC IY           ;-----
           DJNZ FUZYVAL    ;Checa si ya se realizo para los conjuntos
FUZYCPA:   CP B             ;Se posiciona en el lugar correcto de
           JR Z,FUZYPOS    ;los resultados en FUZIN
           INC A
           INC DE
           JR FUZYCPA
FUZYPOS:   DEC C           ;Checa si ya se realizo para las entradas
           JR NZ,FUZYB
           POP HL           ;-----
           POP DE           ;Recupera los valores anteriores
           POP BC           ;de los registros
           POP AF           ;-----
           RET              ;Regresa
;***** FUNCION EVALUA *****
EVALUA:    PUSH BC
           PUSH DE
           LD D,(IX+GEN1)   ;Recupera valor de la entrada
           LD A,(IY+P4)     ;Se coloca en el punto 4
           SUB D             ; A = P4 - ADC
           JR C,VALNEG      ; Si es negativo Termina
           LD E,A           ; A -> E
           LD A,D
           SUB (IY+0)       ; A = ADC - P1

```

## Control de temperatura con lógica difusa

```
JR C, VALNEG ; Si es negativo Termina
LD B, 02H ; Por ser dos pendientes por conjunto
LD (IX+VAR1H), 00H ; Variables de la mult. sin utilizar
LD (IX+VAR2H), 00H ; Variables de la mult. sin utilizar
VAL2: LD (IX+VAR1L), A ; Valor del incremento
LD A, (HL) ; Valor de la pendiente
LD (IX+VAR2L), A
CALL MULTI ; Realiza la multiplicación
LD A, (IX+VAR3H) ; -----
CP 0 ; Verifica si el resultado es > FFH
JR Z, VAL0 ; -----
LD (IX+VAR3L), 0FFH ; Si es >FFH coloca FFH
VAL0: LD C, (IX+VAR3L) ; Resultado -> C
INC HL ; Sig. pendiente
LD A, E ; E -> A
LD E, C ; C -> E
DJNZ VAL2 ; ? para las dos pendientes
LD B, A ; *****
SUB E ; Obtiene el valor
JR NC, MENE ; mínimo de los
LD A, B ; productos de los
JR VALFIN ; incrementos y
MENE: LD A, E ; pendientes
JR VALFIN ; *****
VALNEG: LD A, 00H ; Valor cero si in incremento es (-)
INC HL
INC HL
VALFIN: POP DE
POP BC
RET
```

### Listado 5.4 Difusión

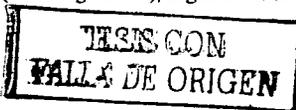
Al tener los valores difusos establecidos y calculados, se procede a evaluar las reglas. De esta manera, se tendrán los valores correspondientes de salida. Cabe mencionar que el valor de las salidas puede variar respecto a la forma de desdifusión, para este sistema se considera el método de *Centro de Máximos*.

### 5.3.2 EVALUACIÓN DE REGLAS

Una vez que se tienen los valores difusos en un bloque de memoria establecido, denominado FUZIN, se procede a operar y evaluar a las correspondientes reglas, respetando el formato indicado en la figura 5.1.

Retomando la figura 5.1, se considera que el bit 7 de la palabra de control corresponde a la etiqueta FIN, el cual es el bit que indica la continuación de la evaluación de las reglas o el final de ellas. Para ello, dicha etiqueta obtiene un valor lógico "1" cuando se presente el final de la evaluación y tendrá el valor "0" lógico para reglas válidas, es decir, continuará evaluando indefinidamente si el valor del bit 7 es "0".

Con lo que respecta a S (bit 5), indica si es el antecedente (valor lógico "0") o consecuente (valor lógico "1"), según el valor que tome corresponderá la función a realizar.



El valor de **X X** (bit 4 y 3) indica el número de antecedente o consecuente tal como se muestra en la tabla 5.1. Para establecer si es consecuente o antecedente se tiene que verificar el valor de **S**.

Bit 4 X	Bit 3 X	Consecuente o antecedente
0	0	1
0	1	2
1	0	3
1	1	4

Tabla 5.1 Número de antecedentes o consecuentes

Por último, los tres primeros bits (bit 0, bit 1 y bit 2) indican la función de membresía correspondiente. Esta palabra de control es capaz de contener máximo ocho funciones de membresía, para el sistema que se desarrolla se trataran solo cinco. En la tabla 5.2 se muestran los números de las membresías.

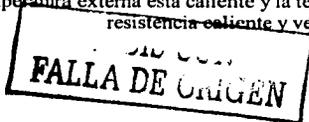
Bit 2 Δ	Bit 1 Δ	Bit 0 Δ	Número de membresía
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Tabla 5.2 Número de funciones de membresía

Como ejemplo, se desarrollan tres reglas, para ello considérese que el sistema está compuesto por dos entradas y dos salidas. La primera entrada corresponde a la temperatura en el interior del sistema, la segunda entrada proporciona la lectura de la temperatura ambiente en el exterior del sistema. Las salidas corresponden al ciclo de trabajo de la resistencia de calentamiento y del ventilador. Cabe mencionar que al término del procesamiento de una salida, el sistema comienza con una nueva regla.

Para ejemplificar, considérese las siguientes tres reglas:

- Si la temperatura externa esta fría y la temperatura interna esta caliente entonces la resistencia fría y ventilador rápido.
- Si la temperatura externa esta caliente y la temperatura interna esta fría entonces la resistencia caliente y ventilador rápido.


  
 FALLA DE ORIGEN

Control de temperatura con lógica difusa

- Si la temperatura externa esta muy fría y la temperatura interna esta muy fría entonces la resistencia muy caliente y ventilador muy lento.

Estas variables lingüísticas pueden ser reducidas a símbolos más sencillos, es decir, sintetizar las variables como:

**MC** = (muy caliente)  
**C** = (caliente)  
**N** = (normal)  
**F** = (frío)  
**MF** = (muy frío)

Para una mejor representación, la variable **E** representará una entrada, mientras que la variable **S** una salida; cada variable con su número de identificación correspondiente, por ejemplo:

**E1** => Entrada 1  
**S2** => Salida 2

Para representar la salida del ventilador, se tiene:

**ML** = (muy lento)  
**L** = (lento)  
**N** = (normal)  
**R** = (rápido)  
**MR** = (muy rápido)



Asimismo, la conjunción “y” se puede reemplazar por **AND** y el “entonces” por **THEN**.

Cabe mencionar que la evaluación de reglas consiste, precisamente, de una operación **AND** entre ellas. Al presentarse un consecuente o respuesta a una salida, se inicia una nueva regla.

Representando las reglas anteriores con la simbolización definida, se tiene lo siguiente:

**SI E2F AND E1C THEN S2F AND S1R**

**SI E2C AND E1F THEN S2C AND S1R**

**SI E2MF AND E1MF THEN S2MC AND S1ML**

Así, se inicia con el análisis de la primera regla. Esta indica que se trata de la entrada o el antecedente 2, considerando la función de membresía 4 y no es final de reglas. De esta manera, se define el valor en hexadecimal a partir de la palabra de control.

**FIN = 0      S = 0      X X = 0 1**

Con una representación binaria, se tiene:

0	0	0	0	1	0	1	1
---	---	---	---	---	---	---	---

y su representación en hexadecimal:

0BH

Al continuar con la evaluación de reglas, se obtiene una tabla conformada por valores hexadecimales, que representan a cada una de ellas.

0BH

01H Si E2F AND E1C THEN S2F AND S1R

2BH

23H

09H

03H Si E2C AND E1F THEN S2C AND S1R

29H

23H

0CH

04H Si E2MF AND E1MF THEN S2MC AND S1ML

28H

20H

De esta manera, la evaluación de reglas consiste en un ciclo cuyo fin será cuando encuentre el código FFH (basta con que el bit 7 sea "1") en la tabla de las reglas, ya que para fines del sistema representará el FIN. En el diagrama de flujo presentado el diagrama 5.6 se ilustra la lógica que desarrolla el programa de evaluación de reglas.

TESIS CON  
FALLA DE ORIGEN

Con base en el diagrama 5.6, se desarrolla el ensamblador para la evaluación de las reglas, con sus respectivos comentarios. Ver listado 5.5.

```

;***** FUNCION QUE EVALUA REGLAS *****
EREGLAS:  PUSH AF
          PUSH BC
          PUSH DE
          PUSH HL
          LD H, FUZH          ;Lugar de FUZIN y FUZOUT (Parte alta)
          LD DE, REGLAS      ;Dirección donde est n las reglas
RSAL:     LD B, OFFH        ;Establece como mínimo el valor PFH
RSIG2:    LD A, (DE)        ;Carga la regla en curso
          BIT 7, A          ;? fin de las reglas
          JR Z, RSIG        ;NO continua con la evaluación
          POP HL           ;SI termina
          POP DE
          POP BC
          POP AF
          RET
RSIG:     INC DE            ;Se posiciona a la siguiente regla
          LD L, A          ;Se completa la dirección (HL)
          BIT 5, A         ;? antecedente o consecuente
          JR Z, RB50       ;Salta si es antecedente
          LD (HL), C       ;Consecuente -> se pone el resultado
          JR RSAL          ;Siguiente regla
RB50:     LD A, (HL)       ;Se realiza operación AND
          SUB B            ;La operación AND en la lógica difusa es el
          JR NC, RNG       ;obtener el mínimo y esto lo realiza
          LD B, (HL)       ;este bloque
RNG:      LD C, B
          JR RSIG2
    
```

Listado 5.5 Ensamblador para la evaluación de reglas

Hasta este momento se tienen los resultados a partir de la evaluación de reglas, los cuales se encuentran localizados en FUZOUT y son cinco valores por salida en este caso.

### 5.3.3 MÉTODO DE DESDIFUSIÓN

El método de desdifusión en este caso es el de Centro de Máximos que está dado por la ecuación 5.1.

$$Z^* = \frac{\sum \mu(z)_i \cdot z_i}{\sum \mu(z)_i} \text{-----}(5.1)$$

Con base en la ecuación 5.1 puede definirse el algoritmo correspondiente y se construirá a partir de un ciclo que realice las multiplicaciones y las divisiones, así como la acumulación de los resultados, recordemos que los conjuntos de salida son del tipo SINGLETONS. En el diagrama de flujo 5.7 se observa la secuencia de dicho programa.

Control de temperatura con lógica difusa

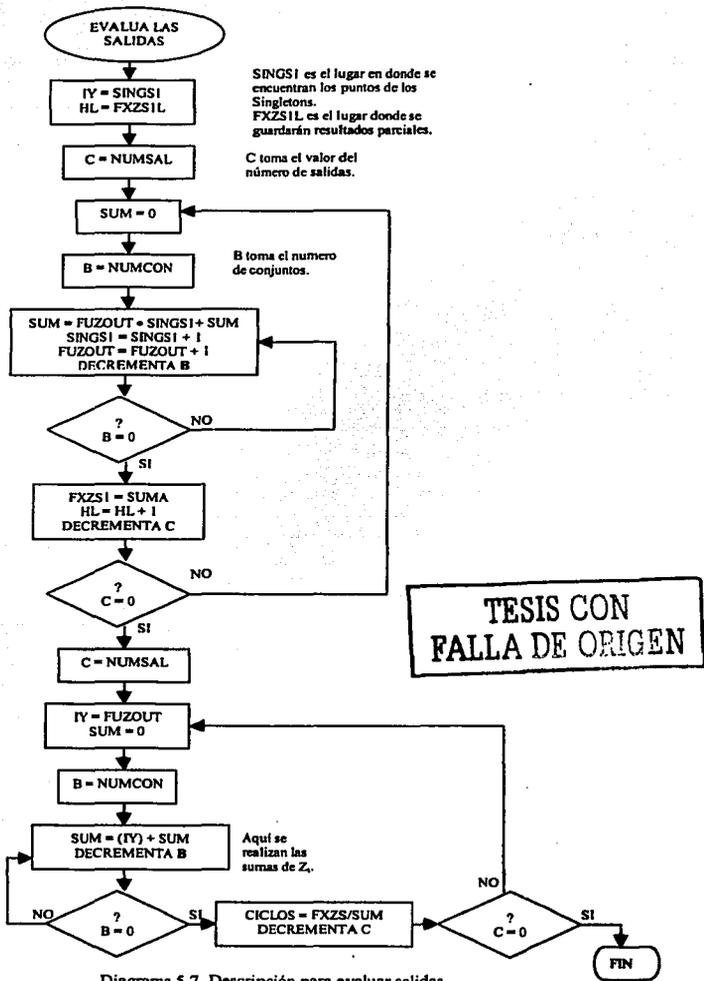


Diagrama 5.7 Descripción para evaluar salidas

Control de temperatura con lógica difusa

De esta manera, los resultados finales quedan en un bloque de 4 registros, uno para cada salida. Estos registros se utilizarán en otra rutina, la cual se encargara de generar el ángulo ( $\alpha$ ) de disparo para controlar la potencia de salida. En el listado 5.6 se muestra el código para el cálculo de la salida de los dispositivos de AC: resistencia y motor.

;\*\*\*\*\* FUNCION QUE EVALUA LA SALIDA \*\*\*\*\*

```

SALIDA:    PUSH AF
           PUSH BC
           PUSH DE
           PUSH HL
           LD IX,SINGS1      ;Lugar de los SINGLETON de salida
           LD HL,FUZOUT      ;Lugar donde se colocara FUZOUT
           EXX               ;En HL' se tendrá F(X)Z
           LD HL,FXZS1L
           EXX

SALCO:     LD A,(NUMSAL)      ;# de salidas
           LD C,A
           LD DE,00H         ;Inicia los registros para la suma
           LD A,(NUMCON)
           LD B,A            ;# de conjuntos
           LD A,(IY+0)       ;Apunta al lugar de Singletons
           LD (IX+VAR1L),A
           LD A,(HL)         ;Apunta al lugar de FUZOUT
           LD (IX+VAR2L),A
           CALL MULTI        ;SINGSij*FUZOUTij
           LD A,(IX+VAR3L)
           ADD A,B
           LD E,A
           LD A,(IX+VAR3H)
           ADC A,D            ;DE=SINGSij*FUZOUTij + DE
           LD D,A
           INC IY
           INC HL
           DJNZ SALBNO
           LD A,(NUMCON)
           CP 8              ;Se posiciona en el lugar correcto de
           JR Z,SALPOS      ;FUZOUT
           INC A
           INC HL
           JR SALCPA

SALPOS:   LD A,E            ;Pone los resultados parciales a
           EXX              ;partir de FXZS1L
           LD (HL),A
           INC HL
           EXX
           LD A,D
           EXX
           LD (HL),A
           INC HL
           EXX
           DEC C            ;Checa si se realizo para todas las salidas
           JR NZ,SALCO

           LD IY,FUZOUT      ;Hasta aquí se tiene el numerador y a
           LD HL,FXZS1L      ;continuación se realiza la suma
           EXX              ;del bloque para el denominador
    
```

**TESIS CON  
FALLA DE ORIGEN**

Control de temperatura con lógica difusa

	LD HL, CICLOS1	;Aquí comienza SUMATORIA DE Z1
	EXX	
	LD A, (NUMSAL)	
	LD C, A	;Se coloca el número de salida
SALB:	LD A, (NUMCON)	;# de conjuntos
	LD B, A	
	LD DE, 00H	
SALCAR:	LD A, (IX+0)	
	ADD A, E	
	LD E, A	
	LD A, D	
	ADC A, 00H	
	LD D, A	
	INC IX	
	DJNZ SALCAR	
	LD A, (NUMCON)	
SALCPA1:	CP 8	;Se coloca en la correcta posición
	JR Z, SALPOS1	;de FUZOUT
	INC A	
	INC IX	
	JR SALCPA1	
SALPOS1:	LD (IX+VAR1L), E	;Se pone el resultado en un registro
	LD (IX+VAR1H), D	;temporal para realizar las operaciones
	LD A, (HL)	
	LD (IX+VAR2L), A	
	INC HL	
	LD A, (HL)	
	LD (IX+VAR2H), A	
	INC HL	
	CALL DIV	
	LD A, (IX+VAR3L)	
	EXX	
	LD (HL), A	;CICLOS1-FXZS1/DE
	INC HL	
	EXX	
	DEC C	;Lo realizé para todas las salidas
	JR NZ, SALB	
	POP HL	
	POP DE	
	POP BC	
	POP AF	
	RET	

TESIS CON  
 FALTA DE ORIGEN

Listado 5.6 Ensamblador para evaluar la salida

Hasta este momento se tienen los resultados generados por la lógica difusa, a continuación se procede a acondicionar los valores para los actuadores de salida. Cabe mencionar que el acondicionamiento depende de la estructura del sistema, por lo que su desarrollo no es general.

Por último se tiene la función que controlará el ángulo de disparo, la cual comprende un rango establecido de FFH a 00H, con un equivalente aproximado de 0.7° por unidad hexadecimal. El diagrama de flujo se ilustra en la figura 5.8.

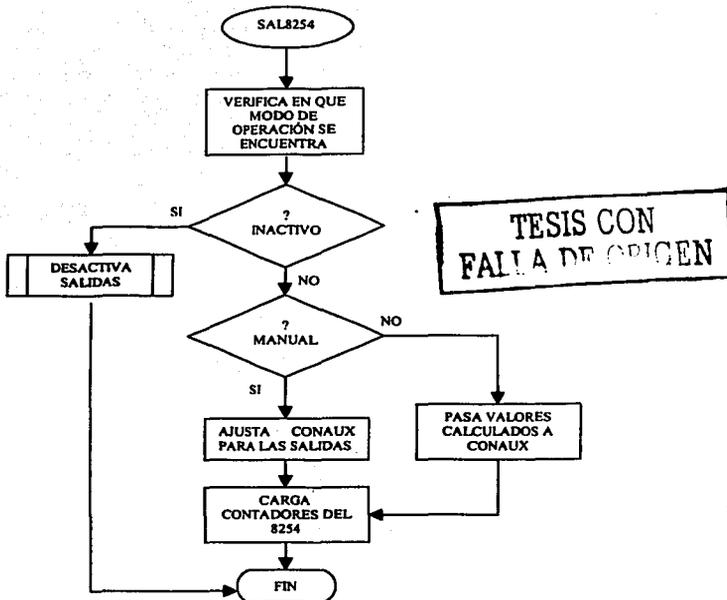


Diagrama 5.8 Descripción para controlar el ángulo de disparo

A continuación en el listado 5.7 se tiene el programa SAL8254 ensamblado. Cabe mencionar que el puerto de salida se denomina 8254. Como se mencionó en el capítulo cuatro, el 8254 es un timer que nos da la funcionalidad para poder hacer el disparo a un ángulo determinado por lo que de esa manera se libera carga de trabajo para el microprocesador y así también la complejidad de programación.

```

;***** FUNCION SAL8254 *****
SAL8254 :   PUSH AF
           PUSH BC
           PUSH DE
           PUSH IY
           LD A, (IX+B_M)
           CP 04H           ;Verifica si esta en inactivo
           JR NZ, CONAC
  
```

Control de temperatura con lógica difusa

```

CALL DES8254 ;Si est en inactivo anula las salidas
JR CONREG ;el sistema no se altere
CONAC: LD A, (IX+B_M) ;Verifica si esta manual
CP 02H
JR Z, CONSIG
LD B, 04H
LD IY, CICOS1
CONREC: LD A, (IY+0)
LD (IY+CONAUX), A
INC IY
DJNZ CONREC
CONSIG: LD IY, CICOS1+CONAUX
LD A, 0
LD B, (IY+0) ;Valor inicial para el cont3
SUB B
CP 0H
JR NZ, SALUNO
LD A, OFFH
SALUNO: OUT (P8254+2), A
LD A, 0
OUT (P8254+2), A
LD B, (IY+1) ;Valor inicial para el cont2
SUB B
CP 0H
JR NZ, SALDOS
LD A, OFFH
SALDOS: OUT (P8254+1), A
LD A, 0
OUT (P8254+1), A
CONREG: POP IY
POP DE
POP BC
POP AF
RET
;***** DES8254 *****
DES8254: PUSH AF
LD A, WORDCON1 ;Programa el Contador1 como generador de onda
OUT (P8254+3), A ;cuadrada a partir del CLK del sistema
LD A, DIV255 ;para que entren 255 divisiones en el periodo
OUT (P8254), A ;de media onda senosoidal de la fuente
LD A, 0 ;eléctrica
OUT (P8254), A
LD A, WORDCON2 ;Programa el Contador2 como diparado por señal
OUT (P8254+3), A ;de hardware
LD A, DIVINI ;Valor inicial para el cont2****!|||||
OUT (P8254+2), A
LD A, 0
OUT (P8254+2), A
LD A, WORDCON3 ;Programa el Contador3 como diparado por señal
OUT (P8254+3), A ;de hardware
LD A, DIVINI ;Valor inicial para el cont3****!|||||
OUT (P8254+1), A
LD A, 0
OUT (P8254+1), A
POP AF
RET

```

**TESIS CON  
FALLA DE ORIGEN**

## 5.2 COMUNICACIÓN CON EL USUARIO

La comunicación con el usuario es el medio por el cual se verifica que el sistema esté funcionando correctamente o bien para determinar parámetros de funcionamiento. Por ejemplo, poder determinar la temperatura a la que se desea controlar el sistema, actividad o inactividad del mismo, modo de uso: manual o automático, etc.

Para determinar esta parte del programa considérese algunos objetivos como son:

- Que la manipulación del sistema sea lo más sencillo posible
- El manejo de botones sea intuitivo
- Los despliegues sean cortos pero claros

Los objetivos anteriores son enfocados a usuarios que no necesariamente conozcan de Lógica Difusa, aquellos que sólo se limitan a controlar la operación del sistema.

Asimismo, es necesario una comunicación para modificar los parámetros de control. Esto, por la necesidad de examinar el sistema, así como para desarrollar la base de conocimiento en tiempo real sin la necesidad de estar grabando en memoria EPROM para cada prueba.

El sistema tiene cuatro modos de operación que son:

- 1.- AUTOMATICO
- 2.- MANUAL
- 3.- INACTIVO
- 4.- PROGRAMA

El modo AUTOMATICO es aquel que desarrolla el control con las herramientas de Lógica Difusa. En este modo se selecciona la temperatura deseada a través de los botones INTRO y SELEC, pero en el transcurso de la selección el control se desactiva.

El segundo modo es MANUAL, el cual se encarga de dar un ángulo de disparo determinado sin llevar a cabo cualquier tipo de control. Esto se implementa con el objetivo de realizar pruebas a los actuadores. Se puede variar los ángulos de disparo de las salidas en forma independiente.

El tercer modo permite poner INACTIVO el sistema, es decir, desactiva las salidas y se concreta a actualizar la temperatura de despliegue.

## Control de temperatura con lógica difusa

Por último, se considera el modo PROGRAMA, en el cual se requiere tener conocimientos básicos de Lógica Difusa, ya que en esta parte puede modificarse la base del conocimiento.

En el modo PROGRAMA se tiene un submenú en donde se puede escoger diferentes características como *cambiar despliegue*, *cambiar de valores predeterminados a programados* y *cambiar rangos*.

*Cambiar despliegue* alterna la forma de salida de información del sistema, por un lado se tiene solo el despliegue de la temperatura interna actual y por el otro se despliega tanto la temperatura interna actual como la temperatura externa del sistema, ya que como se mencionó en un principio, cuenta con dos sensores de temperatura.

*Cambiar valores predeterminados a programados* y viceversa es una forma de modificar el sistema, es decir, al definir la base de conocimiento en forma diferente de cómo se tiene en memoria ROM se puede pasar de uno a otro sin problema.

El funcionamiento del sistema puede ser modificado al cambiar la base de conocimiento, esto puede realizarse al *cambiar rangos* de algunos parámetros como:

- Número de Entradas.
- Número de Salidas.
- Número de Conjuntos.
- Conjuntos de Entrada.
- Conjuntos de Salida.
- Reglas.

Tanto para el número de entradas como para el número de salidas se tiene un tope de 4, es decir, por efecto de la palabra de control sólo pueden manejarse en el sistema 4 entradas y 4 salidas. Cabe mencionar que estas entradas deben implementarse en el programa ya que pueden ser variables físicas o bien comportamiento de ellas como son la derivada o la integral permitiendo, de esta manera, realizar control del tipo PD, PI o PDI.

El número de Conjuntos está limitado a 8 por la misma razón que las entradas. Este número de conjuntos es considerado tanto para los conjuntos de entradas como para los conjuntos de salidas.

Los conjuntos de entrada estarán limitados por el número de conjuntos previamente seleccionado y se podrán dar acceso a 4 Puntos de un conjunto, esto para dar mas flexibilidad al sistema ya que los conjuntos podrán ser trapezoidales y triangulares.

## **Control de temperatura con lógica difusa**

Los conjuntos de salida están también limitados por el número de conjuntos anteriormente fijado y estos conjuntos de salida tienen la forma de SINGLETON, por lo que solo se admite un número que corresponde a la magnitud, en porcentaje, del ángulo de disparo, que va desde 00H, que es el 0%, hasta el FFH que implica un 100%.

Por último se tienen las reglas que se refiere precisamente al valor correspondiente a una regla predeterminada. Se puede tener acceso a 255 reglas.

Para manejar los valores se tienen dos teclas que son las de “+” y “-”, ellas son las que permiten incrementar o decrementar un valor en particular. Estas teclas tienen la función de auto - incremento, es decir, si se mantiene presionada la tecla por un determinado tiempo, el incremento o decremento se realiza de manera acelerada.

En los modos de operación AUTOMATICO y MANUAL se tiene la opción de cambiar la temperatura deseada y el ángulo de disparo respectivamente.

En la presentación todo el listado del sistema operativo ya codificado se muestra en el apéndice C.

## CAPITULO 6

### RESULTADOS Y CONCLUSIONES

#### 6.1 INTRODUCCION

Sin duda, el control juega una parte muy importante en muchos de los procesos de hoy en día y su alcance es simplemente ilimitado, mientras más complejo sea el proceso a controlar, el sistema de control tendrá que cumplir con las necesidades requeridas: velocidad, exactitud, ser óptimo, práctico y económico; de esta manera, deben considerarse nuevas y evolucionadas técnicas que permitan cumplir con todas las necesidades presentes.

Actualmente una de las técnicas que ha evolucionado, que se adecua, en buena medida, a las demandas actuales es el *Control con Lógica Difusa (FLC)*, el cual se considera dentro de la clasificación de sistemas expertos, es decir, el desarrollo de ciertos aspectos de un sistema difuso depende, en gran parte, de la sensibilidad y experiencia humana, por ejemplo, el ajuste de un sistema de aire acondicionado de un recinto, en donde la persona que lo controla tiene la sensibilidad de controlar las condiciones de dicho sistema con base en la experiencia que se adquiere después de ajustarlo varias veces.

Durante el desarrollo del presente trabajo se percibió la importancia de tener un amplio conocimiento sobre el proceso a controlar, pues de ello depende el óptimo ajuste del sistema de control, no basta solamente con tener una buena arquitectura electrónica o una adecuada programación del kernel de la lógica difusa; en realidad lo más importante es tener una base de conocimiento sólida y eficaz para el proceso a controlar.

El planteamiento del presente proyecto fue pensado para que fuera un sistema que tuviera los recursos necesarios para poder ser una herramienta de diseño y ajuste de la base del conocimiento, esto por las necesidades mismas de ajustar y probar. Esto eleva en cierto grado la complejidad en el desarrollo del mismo, pero permitió, por otro lado, la realización de un sistema completo y didáctico, facilitando el ajuste de parámetros para su correcto funcionamiento.

La naturaleza de la estructura del sistema permitió el desarrollo de un sistema de control derivativo, con el firme propósito de seguir una entrada deseada hasta obtener el mínimo error posible.

#### 6.2 RESULTADOS

Los resultados obtenidos, los cuales se muestran en la figura 6.1, proyectan el comportamiento del control, pero cabe mencionar que fue necesario modificar, en diversas ocasiones, conjuntos y reglas para poder establecer una sintonización del sistema óptima y que cumpla con las necesidades del usuario. Esa es la particularidad de la lógica difusa, ya que un control convencional binario no permitiría realizar esa sintonización tan accesible sin evitar cálculos matemáticos e incluso inserción de etapas en el hardware.

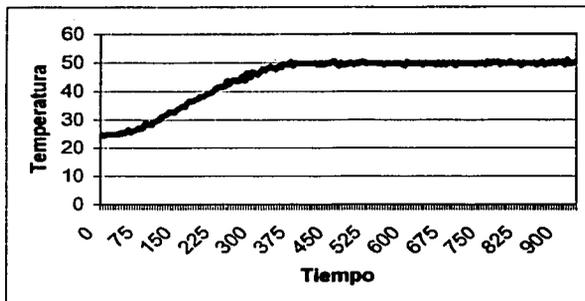


Figura 6.1.a) Respuesta de incremento de temperatura

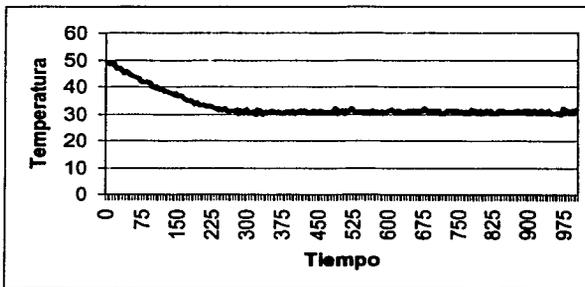


Figura 6.1.b) Respuesta del decremento de temperatura

Figura 6.1 Respuesta del control derivativo

En la figura 6.1 se muestra la respuesta del sistema de control derivativo donde se ilustra la parte ascendente y descendente de la misma.

**TESIS CON  
FALLA DE ORIGEN**

El sistema resulto ser amigable y manejable gracias al diseño que se planteo desde un principio, controla la temperatura con una pequeña oscilación de  $\pm 1^{\circ}\text{C}$ , pero en aplicaciones como las que se proponen este error es tolerable.

### 6.3 CONCLUSIONES

Para desarrollar un sistema de control con lógica difusa fue necesario considerar la importancia del fenómeno a controlar determinando ciertas características y de esa forma, elegir los requerimientos necesarios y la electrónica asociada. Si se requiere controlar un determinado proceso donde los cambios de las variables son lentas (como la temperatura de una pecera) no es necesario tener grandes recursos en el sistema de control, bastaría con un microcontrolador de 8 bits con una velocidad modesta. Asimismo, si el sistema a controlar no requiere de mucha exactitud o precisión, es suficiente con determinar pocos conjuntos de entrada y salida con ciertas reglas.

Cuando se planteó la idea de desarrollar un control difuso se pensó en que tuviera la facilidad de modificar su funcionamiento con el objetivo de hacer pruebas sin tener que emplear otras herramientas, por lo que la arquitectura del software esta diseñada para soportar cambios en los parámetros, tanto en los conjuntos de entrada como de salida y también en las reglas, además de aceptar diferentes modos de trabajo. Todo esto facilito, en gran medida, el ajuste de la base del conocimiento sin la necesidad de borrar y grabar memorias.

La ventaja que presenta este sistema difuso es que se puede realizar y mantener una sintonización del control y un comportamiento físico y funcional del mismo, ya que haciendo uso del control convencional serian necesarias diversas herramientas físicas y matemáticas para realizar lo que con la lógica difusa se haría sólo cambiando condiciones lingüísticas en las reglas correspondientes.

Asimismo, presenta una buena respuesta al incrementar la temperatura, esto gracias a los cálculos que se realizaron para determinar las características de los actuadores que se utilizan, pero cuando se requiere bajar la temperatura, el sistema únicamente lo puede hacer hasta llegar a la referencia establecida por la temperatura ambiente, de exigirse menores temperaturas a esta, deben considerarse otras alternativas que se mencionan más adelante. Por lo tanto, considérese este sistema como un eficiente elevador de temperatura, el cual mantiene condiciones estables partiendo de la línea base que determina la temperatura ambiente.

El software desarrollado para el funcionamiento del sistema permite manipular la base del conocimiento prácticamente en tiempo real; con la visualización establecida en el LCD se pueden manejar otros parámetros del comportamiento del sistema. Pero cabe mencionar que las modificaciones de la base del conocimiento se llevan a cabo en lenguaje de maquina, lo ideal sería que fueran con variables lingüísticas, pero esto demandaba más tiempo de desarrollo y más recursos de memoria.

El sistema difuso permite sacar provecho de la vida útil de los actuadores, ya que basta con cambiar los SINGLETONS para poder seguir operando a los actuadotes que

posiblemente hayan bajado su potencia por cuestiones de uso y sea más caro el cambiarlo que el de modificar los conjuntos de salida. Sin embargo, el cambio de los actuadores implica un repaso del capítulo 2, ya que tomando algunas consideraciones se realizaron los cálculos pertinentes para establecer las condiciones de operación de los mismos.

Las dimensiones físicas de la arquitectura de desarrollo es bastante manejable, pero podría reducirse más realizando una nueva distribución en algún montaje de doble cara como el que presentan algunas compañías comerciales al vender sus arquitecturas microcontroladoras. Sin embargo, cabe aclarar que la arquitectura de este sistema está diseñada de acuerdo a las necesidades que se fueron presentando, difícilmente se hubiera encontrado una arquitectura comercial que cumpliera con las exigencias de esta aplicación. A pesar de que algunos componentes de la arquitectura de desarrollo han existido en el mercado por largo tiempo y pueden ser considerados "viejos", el criterio empleado para su utilización tiene como base el desarrollar un sistema óptimo y práctico con los menores recursos posibles.

### **Mejoras en el proceso de control**

Con el objetivo de agilizar el decremento en la temperatura, considérese un sistema "serpentin" donde se hace fluir refrigerante. Esto lleva a considerar, dentro de la etapa de actuadores, sistemas de inyección y flujo que pasen por el ventilador.

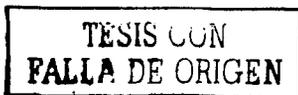
Podría considerarse una salida más donde se sence el flujo del refrigerante y con base en ese dato, determinar la cantidad de refrigerante que se inyecta. Aunque este rubro puede considerarse constante y dejar al ventilador como el actuador primario que controle el proceso de enfriado. Cabe mencionar que ambos actuadores pueden ser igualmente controlados.

De esta manera, podría agilizarse el decremento de la temperatura dentro de un recinto, el cual presentó un desarrollo lento en las pruebas realizadas.

Obviamente al sintonizar el sistema, variando condiciones de reglas y conjuntos, se obtiene una respuesta más estable y rápida del control, pero debido a que la única fuente externa que puede proporcionarme temperaturas bajas (tendencia negativa de calor) es la temperatura ambiente del propio aire, el sistema, en el proceso de bajar la temperatura, se vuelve un poco lento.

Todo lo contrario en al proceso de calentamiento, ya que para determinar las características de los actuadores participantes, se realizaron los cálculos pertinentes.

Así es como se sugiere utilizar un actuador para enfriamiento con el objetivo de agilizar el proceso de decremento de la temperatura. Esto implica modificar el sistema ya que se deberían tener 3 salidas y desde luego cambiar la base del conocimiento.



### **Otras aplicaciones del sistema**

La aplicación de la lógica difusa en este trabajo permite sustituir la mayor parte de la matemática que se desarrolla al plantear un sistema de control convencional con herramientas vagas e instintivas difundidas en la lógica difusa. Resumiendo cálculos algebraicos y obtención de ecuaciones de transferencia, lo difuso abunda en el manejo de reglas determinadas lingüísticamente y que describen, al igual que una ecuación de transferencia, el comportamiento físico de un sistema de control.

La importancia de la lógica difusa poco a poco se ha ido proyectando hacia un mejor desarrollo, según las aplicaciones estructuradas con la misma.

Se puede encontrar principios de lógica difusa en campos de investigación, medicina e industria con gran desarrollo creando tecnología vanguardista con diversas aplicaciones.

Dentro de los sistemas que regulan temperatura, se puede mencionar las incubadoras que se desarrollan para el tratamiento pediátrico de un recién nacido; el sólo hecho de establecer las condiciones deseadas en el interior de la misma, permiten al usuario una completa atención hacia otro tipo de factores, olvidándose de la temperatura.

Actualmente existen automóviles con climas inteligentes, los cuales, gracias a la lógica difusa, respetan un cierto rango de temperatura en su interior, manteniendo el confort del usuario, sin importar las condiciones externas de dicho vehículo. Siempre se desea una temperatura ideal y el control debe respetar dicho valor sin importar las perturbaciones externas.

Asimismo, edificios inteligentes se mantienen bajo un cierto rango de temperatura, con lo que se garantiza una buena estancia de las personas en el interior de los mismos.

No sólo es cuestión de confort, existen laboratorios de investigación, p.e. laboratorios de espectrometría gamma, física, química, donde se requiere mantener el sitio a ciertas condiciones de temperatura y humedad, equipos de instrumentación y elementos como el nitrógeno líquido requieren ciertas condiciones que necesitan ser controladas. La lógica difusa es una herramienta poderosa que puede permitir el control de dichos parámetros de estos sitios y más; con los debidos elementos es una forma muy poderosa de realizar control bajo cualquier condición y con características especiales.

## Apéndice A OPERACIÓN DEL LCD

### I. DESCRIPCIÓN

Este espacio presenta la operación del LCD en la aplicación expuesta, de esta manera sólo se presentan a detalle algunas características. Para complementar la información expuesta consultar el manual de operación del LCD HITACHI mod. LCD-II (HD44 780).

El LCD es un modulo controlador de matriz de puntos hechos con cristal liquido LSI. Puede manejar caracteres alfanuméricos y símbolos. Su control puede estar gobernado bajo microprocesadores o microcontroladores de 8-bits y 4-bits. Todas sus funciones requeridas por la matriz de puntos de cristal liquido están internamente provistas en sus controladores internos. El manejo del LCD puede realizarse con un mínimo de circuitos externos ya que cuenta con lo necesario para poder trabajar con él. Tiene además un consumo mínimo de potencia.

### II. CARACTERISTICAS

- Capaz de ser manejado por 4-bit y 8-bits
- RAM de despliegue de 80 caracteres máxima.
- ROM generador de caracteres
  - Caracteres de una matriz de 5x7 160
  - Caracteres de una matriz de 5x10 32
- Acción de lectura de RAM y ROM por el CPU
- Funciones consideradas:
  - Limpia display, retorno de cursor, encender/apagar display, encender/apagar cursor, blanquea caracteres del display, corrimiento del cursor, corrimiento del display.
- Reset interno automático al recibir energía.

TESIS CON  
FALLA DE ORIGEN

### III. CONEXIÓN LCD-CPU

El LCD cuenta con 14 pines de conexión con lo que se tiene todo lo necesario para poder trabar con él. Dos pines son para la alimentación, uno para ajustar el contraste de despliegue, tres de control y por último cuenta con ocho pines de datos. Los pines se encuentran configurados como se muestra en la figura A.1.

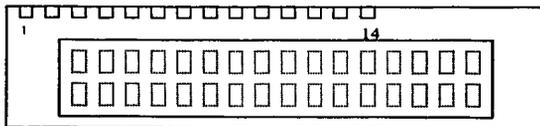


Fig. A.1 LCD

En la tabla A.1 se describen los pines.

PIN	NOMBRE	DESCRIPCIÓN
1	GND	TIERRA
2	V <sub>DD</sub>	ALIMENTACIÓN 5V
3	V <sub>O</sub>	REGULACIÓN DE CONTRASTE
4	RS	SELECTOR DE REGISTRO. Esta señal se encarga de separa el registro de instrucción (IR) y el registro de datos(DR).
5	R/W	LECTURA/ESCRITURA
6	E	HABILITAR
7	DB <sub>0</sub>	DATO 0
8	DB <sub>1</sub>	DATO 1
9	DB <sub>2</sub>	DATO 2
10	DB <sub>3</sub>	DATO 3
11	DB <sub>4</sub>	DATO 4
12	DB <sub>5</sub>	DATO 5
13	DB <sub>6</sub>	DATO 6
14	DB <sub>7</sub>	DATO 7

Tabla A.1 Descripción de Pines

### III.1 Contraste

La conexión del contraste se puede realizar conectando una resistencia variable a entre V<sub>DD</sub> y GND, donde el cursor estará conectado en V<sub>O</sub> tal como indica la figura A.2. Cabe mencionar que el contraste puede presentar variaciones ligeras en su funcionamiento debido a las variaciones de temperatura consideradas en el LCD, las cuales son originadas por la disipación de energía que presenta la estructura electrónica del mismo. De esta manera para eliminar dichas variaciones y obtener una invariabilidad en el funcionamiento es necesario hacer uso de componentes adicionales referidos en el manual de operación.

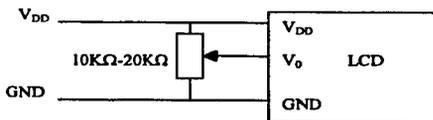


Fig. A.2 Regulación de contraste

Para la aplicación expuesta, la conexión del LCD se expone en la figura A.3. Cabe mencionar que la conexión esta ideada para ocupar el mínimo de hardware sin sacrificar el software. Las líneas GND y V<sub>DD</sub> se conectan a la alimentación de 5V, V<sub>O</sub> se conecta al cursor de la resistencia variable, RS se encuentra en la línea A<sub>0</sub> del microprocesador y R/W

TESIS CON  
FALLA DE ORIGEN

se conecta con A<sub>1</sub>, E se conecta con el Chip Selec respectivo y el bus de datos del LCD se conecta con el bus de datos del microprocesador.

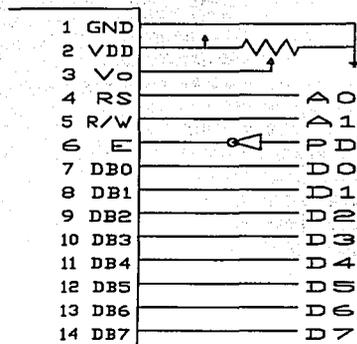


Fig. A.3 Conexión LCD-CPU

Como resultado de la configuración presentada en la figura A.3 se pueden considerar diversas operaciones las cuales son descritas en la tabla A.2.

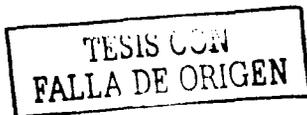
PD	A <sub>0</sub>	A <sub>1</sub>	OPERACIÓN
1	X	X	NO SELECCIONADO
0	0	0	ESCRITURA DE INSTRUCCIÓN
0	1	0	LECTURA DE BANDERA Y DIRECCIONES
0	0	1	ESCRITURA DE DATOS
0	1	1	LECTURA DE DATOS

Tabla A.2 Operaciones

#### IV. SOFTWARE

Una vez discutidos los detalles de hardware se procede al análisis del software ya que es necesario conocer las funciones del LCD para poder tener un aprovechamiento máximo del mismo.

Primero lo que se tiene que hacer después del reset es establecer la longitud de datos, número de líneas en el display y fuente de caracteres, control ON/OFF del display, limpiar el display y por ultimo el modo de trabajo del display. En el diagrama A.1 se muestra el procedimiento.



En la tabla A.3 se muestran las funciones y códigos del display.

FUNCIÓN	RS	R/V	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DESCRIPCIÓN
Limpia display	0	0	0	0	0	0	0	0	0	1	Limpia todo el display y retorna el cursor al inicio (dirección 0)
Retorna	0	0	0	0	0	0	0	0	1	*	Retorna el cursor al inicio (dirección 0) sin alterar el contenido en la DD RAM
Modo	0	0	0	0	0	0	0	1	VD	S	Determina el movimiento del cursor, determina el corrimiento o no del display
Control ON/OFF del display	0	0	0	0	0	0	1	D	C	B	Apaga o enciende a todo el display (D), enciende o apaga al cursor (C), y blanquea la posición del cursor.
Corrimiento del cursor y del display	0	0	0	0	0	1	S/C	R/L	*	*	Recorre el cursor o el display sin afectar el contenido de la DD RAM
Función del LCD	0	0	0	0	1	DL	N	F	*	*	Longitud de datos (DL), número de líneas en el display (L) y fuente de caracteres (F)
Pone dirección CG RAM	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Se establece la dirección de la CG RAM para poder escribir o leer de ella.
Pone dirección DD RAM	0	0	1	ADD	Se establece la dirección de la DD RAM para poder escribir o leer de ella.						
Lectura de la bandera BUSY y dirección	0	1	BF	AC	Lectura de la bandera de ocupado (BUSY) y también el valor del contador de dirección en curso						
Escritura de datos al CG o DD RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Escritura de datos al CG o DD RAM
Lectura de datos al CG o DD RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Lectura de datos al CG o DD RAM

Tabla A.3 Tabla de funciones

Cabe mencionar que los caracteres del display corresponden al código ASCII a partir de la posición 20H hasta la 7DH.

Por otra parte, el LCD permite la definición de 8 posiciones cuya ubicación será de la localidad 00H hasta la 07H. Los caracteres de estas posiciones se definen al cargar los datos en las localidades apartadas para ellos en la CG RAM.



Ejemplo para desplegar "HOLA" a partir de la conexión propuesta y utilizando el microprocesador Z-80.

```
.ORG PROG          ;Inicio del programa

COMLCD EQU *
.DBYTE 380CH,0601H ;Comandos para programar el LCD

HOLA EQU *
.DBYTE 0FE4FH,5045H,5241H,52FEH,454EH,0FE4DH,4F44H,4FFEH
;Letrero

LD BC,LCD+400H ;Carga el periférico y #de datos
;para el LCD
LD HL,COMLCD ;Lugar de comandos para el LCD
SIG: CALL LET_LIS ;LCD listo para recibir
OUTI ;Has una transferencia
JR NZ,SIG
LD BC,LCD+1001H ;Cargar el periférico y #de datos para el
;LCD
LD HL,HOLA ;Lugar del letrero para el LCD
SIG: CALL LET_LIS ;LCD listo para recibir
OUTI ;Has una transferencia
JR NZ,SIG
FIN: JR FIN ;Fin del programa
LET_LIS: PUSH AF
ESP: IN A,(LCD+2) ;Lee la bandera
AND 80H ;Verifica si esta limpia
JR NZ,ESP
POP AF
RET ;retorna
```

TESIS CON  
FALLA DE ORIGEN

## Apéndice B

### SITEMA DE DESARROLLO

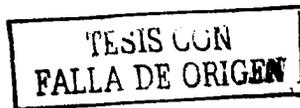
En este apartado se expone la realización de un sistema de desarrollo cuyo funcionamiento facilite la manipulación y ejecución del proyecto. Cabe mencionar que este sistema se diseñó para tener el máximo aprovechamiento del microprocesador y poder visualizar su secuencia en el procesamiento.

Primeramente, para llevar a cabo esta herramienta se identificaron las principales características de funcionalidad con el objetivo de establecer las condiciones estructurales de la misma. A continuación se mencionan las características del sistema de desarrollo:

- Microprocesador Z80A
- Frecuencia de trabajo 4MHz
- Comunicación paralela con la PC
- Cuatro puertos I/O
- Cuatro selectores de memoria
- Acceso a todas las direcciones, datos y líneas de control del Z80
- 4Kbytes de EPROM
- 16Kbytes de RAM

Una vez identificadas las características se procede a la conexión del sistema para esto contamos con la siguiente lista de materiales:

- Microprocesador Z80
- 2716 (EPROM)
- 62256 (RAM)
- 8255 (PPI)
- 74LS139
- 74LS04
- 5 resistencias 4.7K $\Omega$
- 2 resistencias 330 $\Omega$
- 1 resistencia 10K $\Omega$
- Diodo 1N914
- Capacitor de 10 $\mu$ F
- Cristal 4MHz
- Botón
- Conector DB-25
- Conector tipo Peine paso 100 de 50 pines
- Cabezales



El conector cuenta con 50 pines conectados a todas las líneas de control, direcciones y datos necesarios. En la figura B.1 se muestra la configuración del conector, como también se indicaran entre paréntesis el número de pin y de CI donde va conectado.



Fig. B.1 Configuración de pines

Los valores de los puertos de I/O y de los selectores de memoria son como siguen:

- I/O0 00H (Este puerto corresponde a la PPI 8255 por lo que no es externo)
- I/O1 40H Uso externo
- I/O2 80H Uso externo
- I/O3 C0H Uso externo
- MS1 0000H (Corresponde a la EPROM)
- MS2 4000H (Corresponde a la RAM)
- MS3 8000H Uso externo
- MS4 C000H Uso externo

**TESIS CON  
FALLA DE ORIGEN**

Concluido lo que corresponde a la parte del hardware del sistema se procede ahora al software el cual tiene una gran importancia ya que ser en realidad quien determine el comportamiento de nuestro sistema. El software se compone de dos partes importantes: la primer parte es un programa escrito en ensamblador para el Z80 del sistema armado, grabado en la EPROM y la segunda es un programa escrito en lenguaje C para la PC. Al primer programa que esta en ensamblador lo llamaremos TDESZ80.ASM y para el segundo escrito en lenguaje C lo llamaremos TDESZ80.CPP. Para la realización de ambos programas se definen las funciones que se necesitan.

## CC Y CM

Estas funciones se encargaran de cargar datos a la memoria RAM en una dirección específica. Cargar Código (CC) es específica para enviar un bloque de datos por el puerto paralelo de la PC donde esta última a su vez obtendrá estos datos de un archivo tipo INTEL-HEX. Cargar Memoria(CM) es la misma función que CC pero la diferencia radica en la PC en donde ahora se pone la dirección, el tamaño del bloque y el dato a cargar desde los comandos del programa TDESZ80.EXE.

## CP

Cargar Periférico (CP) se encarga de enviar un dato al periférico (puerto I/O) indicado.

## DR

Desplegar Registros (DR) se encarga de mandar los contenidos de los registros principales y auxiliares a la PC donde son desplegados en pantalla.

## DM

Desplegar Memoria (DM) se encarga de mandar un bloque de ocho contenidos de memoria a partir de la dirección indicada y ser desplegados en pantalla.

## DP

Desplegar Periférico (DP) se encarga de mandar el dato obtenido del periférico especificado y desplegarlo en pantalla.

## P

Parar programa (P) detiene la ejecución de algún programa. Reset por comando.

## E

Ejecutar programa (E) ocurre un salto hacia una dirección dada desde la PC y comienza o ejecutar las instrucciones a partir de ahí.

## S

Este comando afecta solo a la ejecución de la PC e indica que salga del programa, para el sistema de desarrollo no tiene efecto.

## L

Limpiar pantalla. Solo afecta a la pantalla de la PC.

Para el reconocimiento de estos comandos se definen sus claves como se indica a continuación:

```
- CC Y CM 01h
- DM      02h
- DP      04h
- CP      08h
- DR      10h
- E       20h
- P       40h
```

A continuación se presentan los programas tanto para el Z80 como para la PC.

Ensamblador para el Z80

```
;*****FUNCION INICIO*****
.ORG 0000H
P8255 EQU 00H
SPINI EQU 4200H
LUGARSP EQU 4000H
LD SP,SPINI ;CARGA EL STAK POINTER
LD A,0F8H ;PROGRAMA AL 8255
OUT (P8255+3),A
LD A,09H
OUT (P8255+3),A
IM 1
ABINT: EI ;Lugar donde brinca de SI (ES A)
HALT
JR ABINT
;*****FUNCION DE SERVICIO DE INTERRUPCION*****
INT: DI
PUSH IY
PUSH IX
PUSH HL
PUSH DE
PUSH BC
PUSH AF
LD (LUGARSP),SP
CALL ENTRADA ;LLAMA A ENTRADA
CP 01H
JR NZ,NCC
CALL CC ;LLAMA A LA FUNCIÓN CC
JR BRIN
NCC: CP 02H ;COMPARA PARA LA FUNCIÓN DM
JR NZ,NDM
CALL DM ; LLAMA A LA FUNCIÓN DM
JR BRIN
NDM: CP 04H ;COMPARA PARA LA FUNCIÓN DP
JR *+5
.ORG 0038H
JR INT ;INTERUPCIÓN
JR NZ,NDP
CALL DP ; LLAMA A LA FUNCIÓN DP
JR BRIN
```

**TESIS CON  
FALLA DE ORIGEN**

Control de temperatura con lógica difusa

```

NDP:      CP 08H          ;COMPARA PARA LA FUNCIÓN CP
          JR NZ,NCP
          CALL RCP       ; LLAMA A LA FUNCION CP
          JR BRIN
NCP:      CP 10H         ;COMPARA PARA LA FUNCIÓN DR
          JR NZ,NDR     ; LLAMA A LA FUNCION DR
          CALL DR
          JR BRIN
NDR:      CP 20H         ;COMPARA PARA LA FUNCIÓN EX
          JR NZ,NEX     ; LLAMA A LA FUNCIÓN EX
          CALL EXE
          JR BRIN
NEX:      CP 40H         ;COMPARA PARA LA FUNCIÓN ST
          JR Z,ST
BRIN:     POP AF
          POP BC
          POP DE
          POP HL
          JR *+5
          .ORG 0066H
          JP 4400H       ;*****INTERRUPCION NO ENMASCARABLE*****
          POP IX
          POP IY
          EI
          RET
ST:       POP AF
          POP BC
          POP DE
          POP HL
          POP IX
          POP IY
          INC SP
          INC SP
          JP ABINT
;*****FUNCTION CARGAR CODIGO (CC)*****
CC:       CALL ENTRADA  ;LLAMA A LA FUNCIÓN ENTRADA
          LD B,A
          CALL ENTRADA  ;LLAMA A LA FUNCIÓN ENTRADA
          LD C,A
BANPE:   CALL ENTRADA  ;LLAMA A LA FUNCIÓN ENTRADA
          LD H,A
          CALL ENTRADA  ;LLAMA A LA FUNCIÓN ENTRADA
          LD L,A
          LD L,A
          LD D,H
          LD E,L
          CALL ENTRADA  ;LLAMA A LA FUNCIÓN ENTRADA
          LD (HL),A
          LDI
          JP PE,BANPE
          RET
;*****FUNCTION ENTRADA*****
ENTRADA:  IN A,(P8255+2) ;SUBROUTINA ENTRADA
          BIT 3,A
          JR Z,ENTRADA
          IN A,(P8255)
          RET

```

**TESIS CON  
FALLA DE ORIGEN**

Control de temperatura con lógica difusa

```

;*****FUNCION SALIDA*****
SALIDA:  PUSH AF          ;SUBROUTINA SALIDA
SALID:   IN A, (P8255+2)
         BIT 7,A
         JR Z, SALID
         POP AF
         OUT (P8255),A
         RET
;*****FUNCION DESPLEGAR MEMORIA (DM)*****
DM:      CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD H,A
         CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD L,A
         LD B,08H
         LD A,(HL)
SIGA:    INC HL
         CALL SALIDA      ;LLAMA A LA FUNCIÓN SALIDA
         DJNZ SIGA
         RET
;*****FUNCION CARGA PERIFERICO (CP)*****
RCP:     CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD C,A
         CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         OUT (C),A
         RET
;*****FUNCION DESPLIEGA PERIFÉRICO (DP)*****
DP:      CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD C,A
         IN A,(C)
         CALL SALIDA      ; LLAMA A LA FUNCIÓN SALIDA
         RET
;*****FUNCION DE EXECUCIÓN (EX)*****
EXE:     LD SP,SPINI      ;CARGA EL STAK POINTER
         CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD H,A
         CALL ENTRADA     ;LLAMA A LA FUNCIÓN ENTRADA
         LD L,A
         EI
         JP (HL)
;*****FUNCION DESPLEGAR REGISTROS (DR)*****
DR:      LD HL,(LUGARSP)
         LD B,08H
         LD A,(HL)
AHL:     INC HL
         CALL SALIDA      ;LLAMA A LA FUNCIÓN SALIDA
         DJNZ AHL
         EX AF,AF'
         CALL SALIDA      ;LLAMA A LA FUNCIÓN SALIDA
         PUSH AF
         EX (SP),HL
         LD A,L
         CALL SALIDA      ;LLAMA A LA FUNCIÓN SALIDA
         EX (SP),HL
         POP AF
         EX AF,AF'
         EXX
         LD A,C

```

TESIS CON  
FALLA DE ORIGEN

## Control de temperatura con lógica difusa

```
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,B
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,E
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,D
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,L
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,H
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
LD A,I
CALL SALIDA ;LLAMA A LA FUNCIÓN SALIDA
EXX
RET
```

## Programa en lenguaje C para la PC

```
/*ESTE PROGRAMA ES PARA LA TARGETA DE DESARROLLO TDES_Z80 QUE TIENE EL
SISTEMA
TDESZ80.ASM Y ES CAPAS DE CARGAR ARCHIVOS TIPO CML Y TIPO INTEL.HEX*/
```

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<ctype.h>
#include<stdlib.h>
#include<process.h>
#include<bios.h>
#define CC 0x01 //Cargar código
#define DM 0x02 //Desplegar memoria
#define DP 0x04 //Desplegar periférico
#define CP 0x08 //Cargar periférico
#define DR 0x10 //Desplegar Registros
#define EX 0x20 //Executar Programa
#define ST 0x40 //Parar programa
```

```
FILE *pf;
unsigned con=0,con2=0,exe=0x4200,ban=1,colum=1;
char nom[50]; //variables globales
```

```
void lissal(unsigned salida)
```

```
{
    unsigned est;long inter=0; //función de salida
    do{est=inp(0x379)&0x40;inter++;}while(est!=0&&inter<100000);
    if(inter==100000)
    {
        printf("\nNO RESPONDE\n"); outp(0x378,0x00);sound(800);
        delay(500);nosound();delay(1000);exit(0);
    }
    else
    {
        outp(0x378,salida);outp(0x37a,inp(0x37a)|0x01);
        outp(0x37a,inp(0x37a)&0xfe);
    }
}
unsigned lissent(void)
```

TESIS CON  
FALLA DE ORIGEN

```

{
    unsigned est; long inter=0; //función de entrada
    do{est=inp(0x379)&0x10; inter++;}while(est!=0&&inter<100000);
    if (inter==100000)
    {
        printf("\nNO RESPONDE\n"); outp(0x378,0x00); sound(800);
        delay(500); nosound(); delay(1000); exit(0);
    }
    else
    {
        outp(0x37A, inp(0x37a) | 0x08); est=inp(0x378);
        outp(0x37A, inp(0x37a) & 0xf7);
    }
    return (est);
}

void cc(void)
{
    // función que carga desde un disco un
    unsigned dat, dat2, i, ac=1, ban=0, dir=0x4200, dirH, con_hex=0, b_c=0;
    // archivo que sea en ASCII escrito desde
    // algún editor.
    printf(" Nombre del archivo: ");
    gets(nom);
    if (nom[0]!='\x0') return;
    pf=fopen(nom, "r"); con=0;
    while(!ferror(pf) && !feof(pf) && pf!=NULL)
    {
        add: dat=fgetc(pf);
        if (con==0 && dat==58) {b_c=1; goto hex1;} //indica que es un
        //archivo *.hex
        if (!isxdigit(dat))
        {
            if (dat==34)
                while(fgetc(pf)!=34);
            if (dat==36) ban=1;
        } //indica que es una dirección
        else
        {
            if (dat>96) dat=dat-32; //ASCII - > HEXADECIMAL
            if (dat>64) dat=dat-55;
            else dat=dat-48;
            if (ac!=1)
            {
                dat=dat*16+dat; ac=1;
                if (ban==0)
                {
                    lissal(CC);
                    lissal(0); lissal(1);
                    lissal(dir/256); lissal(dir&0xff);
                    lissal(dat);
                    gotoxy(5, 24); printf("Numero de byte
                    ", con, con);
                    con++;
                    dir++;
                }
                if (ban==1) { dirH=dat; ban=2; goto add; }
                if (ban==2) { exe=dir=dirH*256+dat; ban=0; }
            }
        }
    }
}

```

**TESIS CON  
FALLA DE ORIGEN**

```

        do
        {
            dat=fgetc(pf);
        }
        while(dat!=10);
    }
    else
    {
        ac=2;dat2=dat;
    }
}
goto salta; hex1:
while(!ferror(pf)&&!feof(pf)&&pf!=NULL)
{
    hex: dat=fgetc(pf);
    if(!isxdigit(dat))
    {
        if(dat=='58'){b_c=1;goto hex;}
        goto hex2;
    }
    if(dat>96)dat=dat-32;
    if(dat>64)dat=dat-55;
    else dat=dat-48;
    if(ac!=1)
    {
        dat=dat*16+dat;ac=1;
        if(ban==0&&b_c==0&&con_hex!=0)
        {
            lissal(CC);
            lissal(0x00);lissal(0x01);
            lissal(dir/256);lissal(dir&0xff);
            lissal(dat);
            gotoxy(5,24);printf("Numero de byte transmitido:
%XH  %i          ",con,con);
            con++;
            dir++;
            con_hex--;
        }
        if(ban==1){ dirH=dat;ban=2;goto hex;}

if(ban==2){exe=dir-dirH*256+dat;ban=0;dat=fgetc(pf);dat=fgetc(pf);}
if(b_c==1){con_hex=dat;b_c=0;ban=1;if(dat==0)goto salta;}
        else
        {
            ac=2;dat2=dat;
        }
        hex2:
    }
    salta:
    if(ferror(pf)||pf==NULL)
    {
        error: printf("\nERROR DE LECTURA");
        sound(800);delay(500);nosound();
        delay(1000);clearerr(pf);exit(0);
    }
}

```

TESIS CON  
FALLA DE ORIGEN

```

    }
    delay(500);
}

void cm(void)
{ //función Cargar Memoria
  unsigned b,c,d;
  printf(" Dir ini: $");scanf("%x",&b);gets(nom);
  gotoxy(25,24);printf(" No de loc: $");scanf("%x",&c);gets(nom);
  gotoxy(45,24);printf(" Dato:
  $");scanf("%x",&d);gets(nom);lissal(CC);
  if(c>0){lissal((c/256));lissal((c&0xff));}else return;
  while(c>0)
  {
    lissal((b/256));lissal((b&0xff));
    lissal(d);b++;c--;
  }
}

void cp(void)
{ //función Cargar Perif,rico
  unsigned p,d;
  printf(" Perif,rico: $");scanf("%x",&p);gets(nom);
  gotoxy(30,24);printf(" Dato: $");scanf("%x",&d);gets(nom);
  lissal(CP);
  lissal(p);
  lissal(d);
}

void dm(void)
{
  unsigned i,p,fi,co;unsigned d; // función Desplegar Memoria
  printf(" Dirección: $");scanf("%x",&d);gets(nom);
  lissal(DM);
  lissal((d/256));
  lissal((d&0xff));outp(0x37a,inp(0x37a)|0x20);
  if(colum==1||colum==2)fi=2;else fi=12;
  if(colum==1||colum==3)co=2;else co=20;
  for(i=0;i<8;i++)
  {
    gotoxy(co,fi);p=d/256;if(p<16)printf("%0%X",p);else
    printf("%X",p);
    p=d&0xff;if(p<16)printf("0%XH - ",p);else printf("%XH - ",p);
    p=lissent();fi++;d++;
    if(p<16)printf("$0%X ",p);else printf("$%X ",p);
  }
  if(colum==4)colum=1;else colum++;
  outp(0x37a,inp(0x37a)&0xde);outp(0x37a,inp(0x37a)&0xf7);
  outp(0x378,0x00);
}

void dp(void)
{
  unsigned p,d;//función Desplegar Perif,rico
  printf(" Perif,rico: $");scanf("%x",&p);gets(nom);
  lissal(DP);
  lissal(p);outp(0x37a,inp(0x37a)|0x20);
  d=lissent();
  gotoxy(76,20);if(p<16)printf("$0%X ",p);else printf("$%X ",p);
  gotoxy(67,21);
}

```

Control de temperatura con lógica difusa

```

        if (d<16)printf("Dato: $0%X ",d);else printf("Dato: $%X ",d);
        outp (0x37a, inp (0x37a) &0xde); outp (0x37a, inp (0x37a) &0xf7);
        outp (0x378, 0x00);
    }
    void dr(void)
    {
        unsigned rL,rH; //función Desplegar Registros
        lissal (DR); outp (0x37a, inp (0x37a) |0x20);
        rL=lissent (); rH=lissent (); gotoxy (40, 3);
        if (rH<16) printf ("A:0%X ", rH); else printf ("A:%X ", rH);
        if (rL<16) printf ("F:0%X ", rL); else printf ("F:%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (40, 5);
        if (rH<16) printf ("B:0%X ", rH); else printf ("B:%X ", rH);
        if (rL<16) printf ("C:0%X ", rL); else printf ("C:%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (40, 7);
        if (rH<16) printf ("D:0%X ", rH); else printf ("D:%X ", rH);
        if (rL<16) printf ("E:0%X ", rL); else printf ("E:%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (40, 9);
        if (rH<16) printf ("H:0%X ", rH); else printf ("H:%X ", rH);
        if (rL<16) printf ("L:0%X ", rL); else printf ("L:%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (42, 11);
        if (rH<16) printf ("IX:0%X", rH); else printf ("IX:%X", rH);
        if (rL<16) printf ("0%X ", rL); else printf ("%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (42, 13);
        if (rH<16) printf ("IY:0%X", rH); else printf ("IY:%X", rH);
        if (rL<16) printf ("0%X ", rL); else printf ("%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (42, 16);
        if (rH<16) printf ("PC:0%X", rH); else printf ("PC:%X", rH);
        if (rL<16) printf ("0%X\n", rL); else printf ("%X\n", rL);
        rL=lissent (); rH=lissent (); gotoxy (60, 3);
        if (rL<16) printf ("A':0%X ", rL); else printf ("A':%X ", rL);
        if (rH<16) printf ("F':0%X ", rH); else printf ("F':%X ", rH);
        rL=lissent (); rH=lissent (); gotoxy (60, 5);
        if (rH<16) printf ("B':0%X ", rH); else printf ("B':%X ", rH);
        if (rL<16) printf ("C':0%X ", rL); else printf ("C':%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (60, 7);
        if (rH<16) printf ("D':0%X ", rH); else printf ("D':%X ", rH);
        if (rL<16) printf ("E':0%X ", rL); else printf ("E':%X ", rL);
        rL=lissent (); rH=lissent (); gotoxy (60, 9);
        if (rH<16) printf ("H':0%X ", rH); else printf ("H':%X ", rH);
        if (rL<16) printf ("L':0%X ", rL); else printf ("L':%X ", rL);
        rL=lissent (); gotoxy (62, 11);
        if (rL<16) printf ("I:0%X ", rL); else printf ("I:%X ", rL);
        outp (0x37a, inp (0x37a) &0xde); outp (0x37a, inp (0x37a) &0xf7);
        outp (0x378, 0x00);
    }
    void ex(void)
    { //función Ejecutar programa
        printf (" Ejecutar en $"); scanf ("%x", &exe); gets (nom);
        lissal (EX);
        lissal ((exe/256));
        lissal ((exe&0xff)); ban=0;
    }
    void main(void)
    { // programa maestro
        char op='n';
        outp (0x37a, inp (0x37a) &0xde);

```



```

outp(0x37a,inp(0x37a)&0xf7);
outp(0x378,0x00);
inicio:
clrscr();
gotoxy(36,1);printf("TDES-Z80");
gotoxy(65,20);printf("Perif,rico");
dr();
do
{
gotoxy(65,23);
if(ban==1)printf("SIN EJECUCION");
else printf("EJECUTANDO");
gotoxy(1,24);printf("Z80> ");
op=getche();if(bioskey(1)!=0)getch();
if(op=='c' || op=='C')
{
op=getche();if(bioskey(1)!=0)getch();
if(op=='c' || op=='C')cc();
if(op=='m' || op=='M')cm();
if(op=='p' || op=='P')cp();
}
if(op=='d' || op=='D')
{
op=getche();
if(op=='m' || op=='M')dm();
if(op=='p' || op=='P')dp();
if(op=='r' || op=='R')dr();
}
if(op=='e' || op=='E')ex();
if(op=='p' || op=='P'){lissal(ST);ban=1;}
if(op=='l' || op=='L')goto inicio;
if(op=='h' || op=='H')
{
clrscr();
printf("Cargar código          CC\n");
printf("Cargar memoria          CM\n");
printf("Cargar periférico        CP\n");
printf("Desplegar memoria        DM\n");
printf("Desplegar perif,rico     DF\n");
printf("Desplegar registros      DR\n");
printf("Ejecutar programa        E \n");
printf("Parar programa          P \n");
printf("Limpiar pantalla        L \n");
printf("Salir                    S \n");
getch();goto inicio;
}
if(op=='s' || op=='S'){printf("Seguro? (s/n)");op=getch();}
gotoxy(1,24);
printf(" ");
}
while(op!='s' && op!='S');
clrscr();
}

```

TDES Z80  
FALLA DE ORIGEN

Con esto ésta tarjeta de desarrollo se obtiene un gran dominio para el desarrollo del sistema que se realizo y para cualquier otro que se base en Z-80 ya que se ideo como un sistema abierto y no dedicado a una sola aplicación.

Para poder realizar aplicaciones se contó con el ensamblador ASMZ80 que genera archivos con el formato INTEL-HEX.

El diagrama de TDESZ80 se muestra en la siguiente hoja.



## APENDICE C

En este apartado se enlista el programa desarrollado para el presente trabajo, el cual ya cuenta con las rutinas de la lógica difusa, el control del LCD, control de actuadores y de teclado.

Se compilo con el programa ASMZ80.COM el cual general el presente listado.

Al final del apéndice se encuentra el diagrama completo del sistema.

Control de temperatura con lógica difusa

```

0000: ;*****
0000: ;*Este programa se encarga de la interfaz con el usuario y el control del *
0000: ;*teclado, es el programa principal y de aqui parten los demas subprogramas *
0000: ;*****
0000: ;//////////////////// Lugar de las constantes del sistema //////////////////////
0000:
0000: ;ORG 0800H
0000: ORG 0800H EQU *
0000: 38 0C 06 01 .DEYTE 380CH,0601H;Comandos para programar el LCD
0004: COIDENTER EQU * ;Datos para la generacion de caracteres especiales del LCD
0004: 01 01 01 05 .DEYTE 0101H,0105H,091FH,0804H,0609H,0906H,0000H,0000H
0006: 09 1F 06 04
000C: 06 09 09 06
0010: 00 00 00 00
0014:
0014: ;Estos valores contienen el numero de salidas, entradas y conjuntos del
0014: ;sistema
0014: SINCENOP EQU *
0014: 02 02 05 .BYTE 02H,02H,05H
0017:
0017: ;Estos son los puntos de los conjuntos de entrada. En este caso se cuentan
0017: ;con 5 conjuntos por entrada y son dos entradas por que son 40 datos
0017: CUMPLEIP EQU *
0017: 00 00 2A 54 .DEYTE 0000H,2A54H,2A54H,5480H,5480H,80A0H,80A0H,0A0B0H,0A0B0H,0FFFFH
001B: 2A 54 54 80
001F: 54 80 80 AB
0023: 80 AB AB D6
0027: AB D6 FF FF
002B: 00 00 2A 54 .DEYTE 0000H,2A54H,2A54H,5480H,5480H,80A0H,80A0H,0A0B0H,0A0B0H,0FFFFH
002F: 2A 54 54 80
0033: 54 80 80 AB
0037: 80 AB AB D6
003B: AB D6 FF FF
003F: ;Estos son los puntos de los singletons de salida, son dos salidas
003F: ;y 5 singletons por cada una. Da un total de 10 datos
003F: SINGSLIP EQU *
0043: FF .BYTE 00H,3FH,7FH,0B0H,0FFFH
0044: 4D 6A 87 AA .BYTE 4DH,6AH,87H,0A0H,0C2H
0048: C2
0049:
0049: ;En este lugar se colocan las reglas para evaluar
0049: REGLASIP EQU * ;Las reglas temdran cuando el bit7 = 1
0049: 00 08 20 2C .DEYTE 0008H,202CH,0108H,202BH,0208H,212BH,0308H,222AH,0408H,232BH
004D: 01 08 20 2B
0051: 02 08 21 2B
0055: 03 08 22 2A
0059: 04 08 23 29
006D: 00 09 20 2C .DEYTE 0009H,202CH,0109H,212BH,0209H,222AH,0309H,232BH,0409H,242BH
0061: 01 09 21 2B
0065: 02 09 22 2A
0069: 03 09 23 29
007D: 04 09 24 29
0071: 00 0A 20 2C .DEYTE 000AH,202CH,010AH,212BH,020AH,222AH,030AH,232BH,040AH,242BH
0075: 01 0A 21 2B
0079: 02 0A 22 2A
007D: 03 0A 23 29
0081: 04 0A 24 29
0085: 00 0B 21 2B .DEYTE 000BH,212BH,010BH,222AH,020BH,232BH,030BH,242BH,040BH,242BH
0089: 01 0B 22 2A
008D: 02 0B 23 29
0091: 03 0B 24 29
0095: 04 0B 24 28
0099: 00 0C 22 2A .DEYTE 000CH,222AH,010CH,232AH,020CH,242BH,030CH,242BH,040CH,242BH
009D: 01 0C 23 2A
00A1: 02 0C 24 29
00A5: 03 0C 24 28
00A9: 04 0C 24 28
00AD: FF .BYTE 0FFH ;Fin de las reglas
00AB: ;Los siguientes datos son los letreros que se utilizan en el sistema
00AB: MODO EQU *

```

YESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

O0AR: 20 4F 50 45      .BYTE * OPERAR EN MODO *
O0B2: 52 41 52 20
O0B6: 45 4E 20 4D
O0BA: 4F 44 4F 20
O0BE:
O0B8: 20 20 20 41      AUTOMATI EQU *
O0BA: 20 20 20 41      .BYTE * AUTOMATICO *
O0C2: 55 54 4F 4D
O0C6: 41 54 49 43
O0CA: 4F 20 20 20
O0CC: ;.DEBYTE OPEFESH, OES4SH, OPESSH, 32FESH, OPEFESH, OPEFESH, OPEFESH, OPEFESH ;DEBUC
O0CE: AUTOCINCO EQU *
O0D0: 20 20 20 01      .BYTE * ",01H,"C","AUTO",",01H,"C" *
O0D4: 43 20 41 55
O0D8: 54 4F 20 20
O0DA: 20 01 43 20
O0DE: MANUAL EQU *
O0E0: 20 20 20 20      .BYTE * MANUAL *
O0E4: 20 4D 41 4E
O0E8: 55 41 4C 20
O0EA: 20 20 20 20
O0EC: MANUINCIO EQU *
O0F0: 20 20 20 01      .BYTE * ",01H,"C","MANU",",01H,"C" *
O0F4: 43 20 4D 41
O0F8: 4E 55 20 20
O0FA: 20 01 43 20
O0FC: INACTIVO EQU *
O0FE: 20 20 20 20      .BYTE * INACTIVO *
O102: 49 4E 41 43
O106: 54 49 56 4F
O10A: 20 20 20 20
O10C: INANCINCO EQU *
O110: 20 20 20 01      .BYTE * ",01H,"C","INAC",",01H,"C" *
O114: 43 20 49 4E
O118: 41 43 20 20
O11A: 20 01 43 20
O11C: PROGRAMA EQU *
O120: 20 20 20 20      .BYTE * PROGRAMA *
O124: 50 52 4F 47
O128: 52 41 4D 41
O12A: 20 20 20 20
O12C: FRINCII EQU *
O130: 54 65 6D 70      .BYTE "Temp. Act.",",01H,"C"
O134: 2E 20 41 63
O138: 74 2E 20 20
O13A: 20 20 01 43
O13C: ;.DEBYTE OPEFESH, 4SPESH, OPESSH, 31FESH, OPEFESH, OPEFESH, OPEFESH, 01A3H ;DEBUC
O13E: FRINCII EQU *
O142: 54 20 69 6E      .BYTE "T inc.",",",",T est."
O146: 74 2E 20 20
O14A: 20 20 54 20
O14C: 65 7E 74 2E
O14E: OMBELAR EQU *
O152: 20 20 20 43      .BYTE * OMBELAR DE *
O156: 41 4D 42 49
O15A: 41 52 20 44
O15C: 45 20 20 20
O15E: INESPLITE EQU *
O162: 20 20 20 44      .BYTE * INESPLITE *
O166: 45 53 50 4C
O16A: 49 45 47 55
O16C: 45 20 20 20
O16E: RANCOE EQU *
O172: 20 20 20 20      .BYTE * RANCOE *
O176: 20 52 41 4E
O17A: 47 4F 53 20
O17C: 20 20 20 20
O17E: FREPRO EQU *
O182: 20 20 50 52      .BYTE * FRED.",",7EH,"PROG." *
O186: 45 44 2E 7E
O18A: 50 52 4F 47
O18C: 2E 20 20 20

```

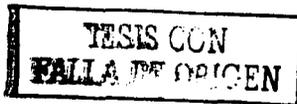
TESIS CON  
 FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

OCBE:          FROFEE   EQU *
OCBE: 20 20 50 52   .BYTE " FROG.", "7EH", "FRED.  "
OC92: 4F 47 2E 7E
OC96: 50 52 45 44
OC9A: 2E 20 20 20
OC9E:          CMFNU1  EQU *
OC9E: 53 61 6C 69   .BYTE "Salida 1:  \t C"
OC92: 64 61 20 31
OC96: 3A 20 20 20
OC9A: 25 20 43 54
OC9E:          CMFNU2  EQU *
OC9E: 28 2D 29 20   .BYTE "(-) (+) ("7FH," ("7EH,")
OC92: 28 2D 29 20
OC96: 20 28 7F 29
OC9A: 20 28 7E 29
OC9E:          CMFNU3  EQU *
OC9E: 28 2D 29 20   .BYTE "(-) (+) ("0FH," ("7EH,")
OC92: 28 2B 29 20
OC96: 20 28 00 29
OC9A: 20 28 7E 29
OC9E:          LAUTO   EQU *
OC9E: 54 65 6D 70   .BYTE "Temperatura: ", "01H", "C"
OC92: 65 72 61 74
OC96: 75 72 61 3A
OC9A: 20 20 01 43
OC9E:          LENTRA  EQU *
OC9E: 23 20 45 4E   .BYTE "# ENTRADAS:  "
OC92: 54 52 41 44
OC96: 41 53 3A 20
OC9A: 20 20 20 20
OC9E:          LSALED  EQU *
OC9E: 23 20 53 41   .BYTE "# SALIDAS:  "
OC92: 4C 49 44 41
OC96: 53 3A 20 20
OC9A: 20 20 20 20
OC9E:          LCENJU  EQU *
OC9E: 23 20 43 4F   .BYTE "# CONJUNTOS:  "
OC92: 4E 4A 55 4E
OC96: 54 4F 53 3A
OC9A: 20 20 20 20
OC9E:          AJENR   EQU *
OC9E: 41 6A 75 73   .BYTE "Ajusta Entradas "
OC92: 74 61 20 45
OC96: 6E 74 72 61
OC9A: 64 61 73 20
OC9E:          AJEALI  EQU *
OC9E: 41 6A 75 73   .BYTE "Ajusta Salidas "
OC92: 74 61 20 53
OC96: 61 6C 69 64
OC9A: 61 73 20 20
OC9E:          AJREEL  EQU *
OC9E: 41 6A 75 73   .BYTE "Ajusta Reglas "
OC92: 74 61 20 20
OC96: 52 65 67 6C
OC9A: 61 73 20 20
OC9E:          AJCEN1  EQU *
OC9E: 45 20 43 20   .BYTE "E C P1:  P2:  "
OC92: 20 50 31 3A
OC96: 20 20 20 50
OC9A: 32 3A 20 20
OC9E:          AJCEN2  EQU *
OC9E: 50 33 3A 20   .BYTE "P3:  P4:  ", "0FH," ", "7EH
OC92: 20 20 50 34
OC96: 3A 20 20 20
OC9A: 20 00 20 7E
OC9E:          AJEAL   EQU *
OC9E: 53 20 20 53   .BYTE "S SINGLETON :  "
OC92: 49 4E 47 4C
OC96: 45 54 4F 4E
OC9A: 20 3A 20 20
OC9E:          AJREO   EQU *

```



Control de temperatura con lógica difusa

```

0066: 52 45 47 4C      ,BYTE "REGLA  VAL:  "
0072: 41 20 20 20
0076: 20 56 41 4C
007A: 3A 20 20 20
007E:          LODNREG  EQU *
007E: 28 2D 29 20      ,BYTE "(-) (+) (" ,OH,") SAL"
0082: 28 2B 29 20
0086: 20 28 00 29
008A: 20 53 41 4C
008E:
008E:          ;//////////////////// Definición de variables //////////////////////
008E:
008E:          BASEFUZ  EQU 1200H      ;Lugar de memoria RAM del Kernel
008E:          BLKFROM  EQU 200H      ;Lugar de variables
008E:
008E:          FUZIN    EQU BASEFUZ    ;Las entradas van desde 0000H a 000FH
008E:
008E:          FUZOUT   EQU BASEFUZ+20H;Las salidas van desde 0020H a 003FH
008E:
008E:          FZSILL   EQU BASEFUZ+40H;Suma de u(z)z de las salidas
008E:          ;Va desde 0040H a 0047H
008E:
008E:          CICLOG1  EQU BASEFUZ+48H;Lugar donde se ponen los resultado
008E:          ;Va desde 0048H a 004EH
008E:          ;De 004CH a 004EH esta CONVALX
008E:
008E:          BLKENT   EQU BASEFUZ+50H;Bloque de entradas que va desde
008E:          ;Va desde 0050H hasta 0053H
008E:
008E:          CIMLES   EQU BASEFUZ+54H;Lugar de las pendientes calculadas
008E:
008E:          TEMSEL   EQU BASEFUZ+BLKFROM;Temperatura seleccionada
008E:          NUMSAL   EQU BASEFUZ+BLKFROM+1;# de salidas
008E:          NUMENT   EQU BASEFUZ+BLKFROM+2;# de entradas
008E:          NUMCON   EQU BASEFUZ+BLKFROM+3;# de conjuntos (anteriores como con
008E:          CLPUN1   EQU BASEFUZ+BLKFROM+4;Lugar de los puntos
008E:          SINGSEL   EQU BASEFUZ+BLKFROM+8H;Singletons
008E:          REGLAS   EQU BASEFUZ+BLKFROM+0A0H;Reglas
008E:          RSSPAL   EQU REGLAS+100H
008E:
008E:          ;-----
008E:          VAR1L    EQU 0          ;-----Estas variables se ocupan en la
008E:          VARGH    EQU 1          ;----- multiplicación y en la
008E:          VAPZL    EQU 2          ;----- división.
008E:          VARGH    EQU 3          ;-----
008E:          VARGL    EQU 4          ;-----Resultado BAJO
008E:          VARGH    EQU 5          ;-----Resultado ALTO
008E:          ;-----
008E:          GRN1     EQU 6          ;Variable de uso general
008E:
008E:          VART1    EQU 7          ;-----
008E:          VART2    EQU 8          ;-----Variables de tiempo de espera-----
008E:
008E:          ;B_M=1H AUTOMATICO B_M=2H MANUAL B_M=4H INACTIVO B_M=8H PROGRAM#
008E:          B_T      EQU 9          ;Bandera de teclado
008E:          B_M      EQU 10         ;Bandera de modo
008E:
008E:          PCONT1   EQU 11         ;Bandera de control 1
008E:          PCONT2   EQU 12         ;Bandera de control 2
008E:
008E:          TBLAT    EQU 13         ;Variable tecla temporal
008E:
008E:          ;Contadores para la entrada de datos
008E:          CONENT   EQU 14         ;Contador de entrada
008E:          CONVAL   EQU 15         ;Contador de salida
008E:          CONCON   EQU 16         ;Contador de conjuntos
008E:          CONPUN   EQU 17         ;Contador de puntos
008E:
008E:          ABRAND   EQU 18         ;Variable para retardar incremento o decremento
008E:          DIR      EQU 19         ;Indica el lugar de despliegue del LCD
008E:          HEKA     EQU 20         ;Bandera que indica si la operación es en hexa o decim
008E:          TOPS     EQU 21         ;Indica hasta que valor llegara alguna variable

```

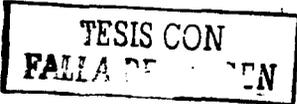


Control de temperatura con lógica difusa

```

O00E:          CONF2   EQU 22           ;Variable que retarda la lectura de entradas
O00E:          ERROR   EQU 23           ;Variable en la cual se guarda el error
O00E:          ERRORS  EQU 24           ;Variable de signo del error
O00E:          ;//////////////////////////////////// Definición de constantes //////////////////////////////////////
O00E:          P4       EQU 3           ;Punto cuatro
O00E:          FUZH    EQU 65H          ;12H;Parte alta de FUZZIN y FUZOUT
O00E:          FILTRAT EQU 0FH          ;Filtro de teclado
O00E:          MENOS   EQU 1           ;Valor correspondiente a la tecla menos
O00E:          MAS     EQU 2           ;Valor correspondiente a la tecla mas
O00E:          ENTER   EQU 4           ;Valor correspondiente a la tecla enter
O00E:          SELEC   EQU 8           ;Valor correspondiente a la tecla selección
O00E:          LD472A  EQU 1           ;Código para limpiar el LCD
O00E:          LDM_DCS EQU 000H         ;Código para pasar a la línea dos del LCD
O00E:          CONXUX  EQU 4           ;Desplazamiento para auxiliares
O00E:          VALBP   EQU MOOD-GABEOP;# de datos a pasar a la RAM
O00E:          RETLL   EQU 10H          ;Retorna un lugar el cursor a la izquierda
O00E:          FACTOR  EQU 64H          ;Factor para convertir de Hexa a Dec.
O00E:          DIV255  EQU 74H          ;82H ;Divisor de frecuencia para el con1 del 8254
O00E:          DIV74H  EQU 0F6H          ;EI valor inicial para el con2 y con3 del 8254
O00E:          WORDCON1 EQU 3EH          ;Palabra de control para el con1 del 8254
O00E:          WORDCON2 EQU 0B4H          ;Palabra de control para el con2 del 8254
O00E:          WORDCON3 EQU 7AH          ;Palabra de control para el con3 del 8254
O00E:          PB254   EQU 80H          ;Salida (Número de periférico para el 8254)
O00E:          TECLA   EQU 40H          ;Número correspondiente al Teclado
O00E:          LCD     EQU 00H          ;Número correspondiente al LCD
O00E:          ACC1    EQU 2000H        ;Valor correspondiente al ACC1
O00E:          ACC2    EQU 3000H        ;Valor correspondiente al ACC2
O00E:          INDEXX  EQU 1100H        ;Lugar de variables
O00E:          ;***** INICIO *****
O00E:          ;ORG 000H
O000:          ;
O000:          ;
O000:          LD SP,10FH              ;Apuntador de pila
O003:          ;
O003:          LD IX,INDEXX            ;Lugar de variables y registros a utilizar
O007:          ;
O007:          LD (IX-PCINT1),00H;Inicia banderas de control
O00B:          LD (IX-PCINT2),00H
O00F:          LD (IX-ERRORS),00H
O013:          LD (IX-ERROR),00H
O017:          LD (IX-B_M),01H;Inicia registro B_M = 01H
O01B:          CALL DESB254
O01E:          ;
O01E:          CALL BLKROMP           ;Pasa el bloque de EPROM a la RAM
O021:          ;
O021:          LD BC,LCD+400H          ;Cargar el periférico y #de datos para el LCD
O024:          LD HL,CMDLCD            ;Lugar de comandos para el LCD
O027:          CALL LET_LIS           ;LCD listo para recibir
O02A:          OUTI                    ;Has una transferencia
O02C:          OR NZ,SIG
O02E:          CALL LET_LIS           ;Prepara la carga de nuevos caracteres
O031:          LD A,40H                ;al LCD correspondientes a los códigos
O033:          OUT (LCD),A            ; 00H y 01H
O035:          ;
O035:          LD BC,LCD+100H;Carga los nuevos caracteres
O038:          LD HL,CDBENTER
O03B:          CALL LET_LIS
O03E:          OUTI
O040:          OR NZ,SIGDOS
O042:          ;
O042:          ;
O042:          LD A,(D<+B_M)          ;Carga al registro B_T con
O045:          LD (IX<+B_T),A          ;lo que tiene B_M
O048:          LD (IX-PCINT2),00H; TPRG = 0
O04C:          CALL LET_PRIN          ;Llama al letrero principal
O04F:          IN A,(TECLA)           ;Obtiene el dato del teclado
O051:          AND FILTRAT            ;quitando los cuatro ultimos bits

```



Control de temperatura con lógica difusa

```

0053: 28 33          JR Z,CON          ;Si A=0 -> CON
0055: 1D 77 0D        LD (DX+TECLAT),A;Si a=0 guarda el valor de la tecla
0058: 1D C3 0C 7E     BIT 7, (DX+PCINT2);Esta en PROGRAM?
005C: 28 05          JR Z,NPRFRO      ;si es así llama a la función programa
005E: CD 38 02       CALL NPRFRO      ;si no entonces compara con la siguiente
0061: 18 1C           JR TECO
0063: 1D 7E 0A        NPRFRO: LD A, (DX+B_M) ;Compara si es AUTOMATICO
0066: FE 01           CP 01H
0068: 20 05           JR NZ,NRPUITO
006A: CD EF 00        CALL FRUITO
006D: 18 10           JR TECO
006F: FE 02         NRPUITO: CP 02H          ;Compara si es MANUAL
0071: 20 05           JR NZ,NRPNANU
0073: CD 53 01      CALL RPNANU
0076: 18 07           JR TECO
0078: FE 04         NRPNANU: CP 04H          ;Esta comparación corresponde a INACTIVO pero
007A: 20 03         JR NZ,TECO       ;como solo afecta a la bandera se procederá
007C: CD EF 00      CALL FRUITO      ;con AUTOMATICO
007F:           ;
007F: CD F2 05      TECO: CALL CONTROL ;Llama a control
0082: 18 40           IN A, (TECLA)   ;Espera a que se desactive la tecla
0084: E6 0F           AND FELIBRAT
0086: 20 F7           JR NZ,TECO
0088:           ;
0088: CD F2 05      CON: CALL CONTROL
008B: 1D C3 0C 4E     BIT 0, (DX+PCINT2)
008F: 28 BE           JR Z,INDATO
0091: 1D 35 07        DEC (DX+VART1)  ;***** VARIABLES DE ESPERA *****
0094: 20 B9           JR NZ,INDATO    ;Sirven para dejar un determinado
0096: 1D 35 08        DEC (DX+VART2) ;tiempo el letrero y esperando también
0099: 20 B4           JR NZ,INDATO    ;una respuesta de parte del usuario
009B: 18 A5           JR LA
009D:           ;***** BLKDATP *****
009D:           ;
009D: CD           BLKDATP: FUSH BC
009E: D5           FUSH DE
009F: E5           FUSH HL
00A0: 21 14 0B        LD HL,SABNOP   ;-- Esta función se encarga de --
00A3: 11 01 14       LD DE,NUMPAL   ;-- pasar el bloque predeterminad
00A6: 01 2B 00       LD EC,SINGELP-SABNOP;-- nado de datos grabados en --
00A9: ED B0         LDIR ;-- la EPROM a la RAM. --
00AB: 21 3F 0B        LD HL,SINGELP ;-----
00AD: 11 85 14       LD DE,SINGELP
00B1: 01 0A 00       LD EC,REGLASP-SINGELP
00B4: ED B0         LDIR
00B6: 21 49 0B        LD HL,REGLASP
00B9: 11 A6 14       LD DE,REGLASP
00BC: 01 65 00       LD EC,MODD-REGLASP
00BE: ED B0         LDIR
00C1: E1           POP HL
00C2: D1           POP DE
00C3: C1           POP EC
00C4: CD 9D 06      CALL PENDIEN   ;Calcula las pendientes predeterminadas
00C7: C9           RET
00C8:           ;
00C8:           ;***** BLKRAM *****
00C8:           ;
00C8: CD           BLKRAM: FUSH BC
00C9: D5           FUSH DE
00CA: E5           FUSH HL
00CB: 21 01 14       LD HL,NUMPAL   ;-- Esta función se encarga de --
00CE: 11 A6 15       LD DE,RESPAL   ;-- pasar el bloque de desplie --
00D1: 01 A5 01       LD EC,RESPAL-NUMPAL;-- que de RAM a otra parte de --
00D4: ED B0         LDIR ;-- la RAM. --
00D6: E1           POP HL
00D7: D1           POP DE
00D8: C1           POP EC
00D9: C9           RET
00DA:           ;
00DA:           ;***** RWMEK *****

```

**TESIS CON  
FALLA DE ORIGEN**

Control de temperatura con lógica difusa

```

000A:
000A: C5          RAMELK:   PUSH BC
000B: D6          PUSH DE
000C: E5          PUSH HL
000D: 21 A6 15    LD HL, R2SPAL ;-- Esta función se encarga de --
000E: 11 01 14    LD DE, NUMESAL ;-- pasar el bloque de la RAM --
000F: 01 A5 01    LD BC, R2SPAL-NUMESAL;-- a la RAM de despliegue --
0010: ED B0      LEI R1 ;-----
0011: E1          POP HL
0012: D1          POP DE
0013: 00EA: C1      POP BC
0014: 00EB: CD 50 D6  CALL PENDIEN ;Calcula las pendientes programadas
0015: 00EE: C9          RET
00EF:
00EF: ;***** PRUTO *****
00F0: ED 7E 0D    PRUTO:   LD A, (IX+TECLAT);Recupera la tecla y verifica
00F1: FE 04      CP ENTER ;si se trata de la tecla ENTER
00F2: 28 14      JR Z, PRUTOCCN ;entonces salta
00F3:
00F4: FE 08      PRUTOSEL: CP SELEC ;si se trata de la tecla SELEC
00F5: C0          RET NZ ;continúa de lo contrario regresa.
00F6: ED CB 09 06 RLC (IX+B,T) ;Se pone en el siguiente modo pero
00F7: DD CB 0C 06 SET 0, (IX+PCINT2);sin activarlo. Activa el tiempo.
00F8: CD 32 05    CALL LET_B_T ;despliega el modo a seleccionar
00F9: 06 05      LD B, 05H
00FA: CD EA 05    CALL FON_TIE
00FB: C9          RET
00FC: ED CB 0C 46 PRUTOCCN: BIT 0, (IX+PCINT2)
00FD: 28 1F      JR Z, S1PAUTO
00FE:
00FF: ED 7E 09    PRUTOGIG: LD A, (IX+B,T) ;Coloca el nuevo modo activo
0100: FE 08      CP 08H ;excepto si es PROGRAMA
0101: 2B 0B      JR Z, VAPROG ;salta
0102: ED 77 0A    LD (IX+B,M), A
0103: DD CB 0C 86 PRUTOREG: RES 0, (IX+PCINT2);Limpia tiempo
0104: CD 1A 05    CALL LET_PRIN
0105: C9          RET
0106: 06 05      VAPROG: LD B, 05H ;Aquí se pone activo el tiempo de espera
0107: CD EA 05    CALL FON_TIE ;además de que se pone el letrero
0108: DD CB 0C FE SET 7, (IX+PCINT2);correspondiente a la selección
0109: CD A6 05    CALL LETIEC ;esta también en modo PROGRAMA
0110: C9          RET
0111: ED 36 14 00 S1PAUTO: LD (IX+HEXA), 0 ;No se trata de un valor hexadecimal
0112: 3E 01      LD A, LIMPIA ;Limpia el LCD para desplegar
0113: D3 00      OUT (LCD), A ;Los letreros correspondientes
0114: 21 C9 0C    LD HL, LAUTO
0115: CD 96 05    CALL DESTIAR
0116: 3E CD      LD A, LIN_DOS
0117: D3 00      OUT (LCD), A
0118:
0119: ED 36 13 8D LD (IX+DIR), 8EH;DIRAM para los dígitos
0120: 21 EE 0C    LD HL, CPANNU
0121: CD 96 05    CALL DESPLEG
0122:
0123: ED 21 00 14 LD HL, TEMSEL ;Lugar donde se tiene la temperatura
0124: CD EF 01    CALL TECLACCN ;seleccionada
0125:
0126: 18 C7      JR PRUTOREG
0127:
0128: ;***** PRANU *****
0129:
0130: ED 7E 0D    PRANU:   LD A, (IX+TECLAT);Recupera la tecla y verifica
0131: FE 04      CP ENTER ;si se trata de la tecla ENTER
0132: 28 02      JR Z, PRANUSIG ;entonces salta
0133: 18 9A      JR PRUTOSEL
0134:
0135: ED CB 0C 46 PRANUSIG: BIT 0, (IX+PCINT2);Verifica si se encuentra en selección
0136: 28 02      JR Z, S1PANNU
0137: 18 AC      JR PRUTOGIG
0138:

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

0164: DD 36 OF 31 RPNANUL: LD (DX+CONSAL),31H;Pone el valor de 1 al contador de salida
0168: DD 36 14 00 LD (DX+HECO),0 ;No se trata de un valor hexadecimal
016C: 3E 01 LD A,LIMPIA ;Limpia el LCD para desplegar
016E: DD 00 OUT (LCD),A ;los letteros correspondientes
0170: 21 9E 0C LD HL,CMPANUL
0174: CD 96 05 CALL DESPLIEG
0176: 3E 00 LD A,LIN DOS
0178: DD 00 OUT (LCD),A
017A:
017A: DD 36 13 8B LD (DX+DIR),8BH;DIRAM para los dígitos
017E: 21 AE 0C LD HL,CMPANUZ
0181: CD 96 05 CALL DESPLIEG
0184:
0184: DD 21 4C 12 LD HL,CICLOSI+CONANU;Lugar donde se tiene los ciclos de las salid
0187: 3A 01 14 LD A,(NUMERAL)
018A: C6 31 AND A,31H ;Este tipo de comparación se realiza
018C: 47 LD B,A ;para que no exceda el límite de salidas
018D: DD 7E 0F LD A,(DX+CONSAL)
0190: BE CP B
0191: 28 87 JR Z,FNUTOREG
0193:
0193: 3E 87 RPNANUZ: LD A,87H ;DIRAM para el numero
0195: DD 00 OUT (LCD),A ;Actualiza DIRAM el numero actual
0197: CD 89 09 CALL LET_LIS
019A: DD 7E 0F LD A,(DX+CONSAL)
019E: DD 01 01 OUT (LCD+1),A
019F: CD 89 09 CALL LET_LIS
01A2: CD B1 01 CALL TECLACN
01A5:
01A5: DD 7E 0D LD A,(DX+TECLAT);Al regresar de la función
01A6: FE 04 CP ENTER ;compara si se trata de regresar
01A8: 2D 0D JR NZ,RPNUI ;un lugar
01AC: DD 7E 0F LD A,(DX+CONSAL)
01AF: FE 31 CP 31H
01B1: 28 E0 JR Z,RPNUIZ
01B3: DD 35 0F DEC (DX+CONSAL);Regresa un lugar atras
01B6: 2B DEC HL
01B7: 1B DA JR RPNUIZ
01B9: DD 34 0F RPNUIZ: INC (DX+CONSAL);En este caso se adelanta un lugar
01BC: 23 DEC HL
01BD: 18 CB JR RPNUI
01BF:
01BF: ;***** TECLACN *****
01BF:
01BF: CD 99 08 TECLACN: CALL DESO254 ;Desactiva las salidas ya que se cambiar algún
01C2: DD 46 0A LD B,(DX+M) ;parámetro
01C3: DD 36 0A 04 LD (DX+M),04H
01C9: 3E 0E LD A,0EH ;Se activa el cursor
01CB: DD 00 00 OUT (LCD),A ;para resaltar el número que
01CD: CD 89 09 CALL LET_LIS ;cambiar
01D0: DD 7E 14 LD A,(DX+HECO);Verificas si se trata de un número
01D3: FE 00 CP 0 ;hexadecimal o decimal
01D5: CD EF 08 CALL Z,DEBAND ;Se coloca el primer número de memoria
01D8: CA 3A 09 CALL NZ,DEBDA ;ya sea decimal o hexadecimal
01DB: DD 00 00 TECCERO: IN A,(TECLA) ;Espera para la desactivación de cualquier
01DD: BE 0F AND FILTRAT ;tecla
01DF: 20 FA JR NZ,TECCERO
01E1:
01E1: DD 36 12 0A TECLACD: LD (DX+ARRANI),0AH;Inicia el arranque
01E5:
01E5: CD 00 0A TECLAIN: CALL TIEMPO 1
01E8: DD 00 00 IN A,(TECLA) ;Obtiene el dato del teclado
01EA: BE 0F AND FILTRAT ;quitando los cuatro últimos bits
01EC:
01EC: FE 08 CP SELEC ;Checa si se trata de alguna tecla de control
01EE: 2E 3A JR Z,TECREF ;si es así retorna al program anterior
01F0: FE 04 CP ENTER
01F2: 28 36 JR Z,TECREF
01F4:
01F4: DD BE 0D CP (DX+TECLAT);Comprueba si la tecla continúa activa
01F7: 20 0C JR NZ,TECLATA

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

01F9:
01F9: ID 7E 12          LD A, (IX+ARRANI);Compara si arranque esta en cero
01FC: FE 00            CP 0
01FE: 28 08            JR Z,TEDECCIN
0200:
0200: ID 35 12          DEC (IX+ARRANI)
0203: 18 ED            JR TECLAIV
0205:
0205: ID 77 0D          TECACIAA: LD (IX+TECLAT),A;Guarda la tecla actual
0208: ID 7E 0D          TEDECCIN: LD A, (IX+TECLAT)
020B: FE 02            CP MAS
020D: 28 06            JR Z,TECMAS ;Realiza la operación correspondiente
020F: FE 01            CP MENOS ;de incremento o decremento
0211: 20 CE            JR NZ,TECLADO
0213: 35              DEC (HL)
0214: 35              DEC (HL)
0215: 34              TCMAS: INC (HL)
0216: ID 7E 14          LD A, (IX+HECA)
0219: FE 00            CP 0
021B: CC EF 08          CNL Z,DEINCO
021E: 04 3A 09          CNL NZ,DECA
0221: ID 7E 06          LD A, (IX+GENI) ;Esta variable indica si el ultimo digito
0224: FE 00            CP 0 ;cambia ya que si no es así se debe repetir
0226: 20 ED            JR NZ,TEDECCIN ;la operación de forma autónoma para que
0228: 18 EB            JR TECLAIV ;exista un cambio en el despliegue
022A: ID 77 0D          TECEG: LD (IX+TECLAT),A
022D: 1E 0C            LD A, 0FH
022F: D3 00            OUT (LCD),A
0231: CD 89 09          CNL LET LIS
0234: ID 70 0A          LD (IX+B_M),B
0237: C9              RET
0238:
0238: ;***** PROG *****
0238:
0238: ID 7E 0D          PPROG: LD A, (IX+TECLAT);Recupera la tecla presionada
023B: FE 04            CP ENTER
023D: 28 0C            JR Z,PPROSIG
023F: FE 08            CP SELEC ;Compara si se trata de SELEC
0241: CD              RET NZ
0242: CD A6 05          CNL LETTEC ;Despliega entonces las siguientes
0245: 06 05            LD B,05H ;siguientes opciones
0247: CD EA 05          CNL FON_TIE
024A: C9              RET
024B:
024B: ;En esta parte del programa se maneja un
024B: ID CB 0C 76          PPROSIG: BIT 6, (IX+PCINT2);subruti derivado del modo programa.
024F: CB CB 04          JP Z,PRNG ;Verifica que se trata del menú o submenú
0252:
0252: ;programa
0252:
0252: ;Entra en el modo de subprograma
0252: ID 36 0C 00          LD (IX+PCINT2),0;Quita banderas
0256: ID 36 14 FF          LD (IX+HEGA),OFFH
025A: 3E 01            LD A, LIMPRA ;Limpia el LCD para desplegar
025C: D3 00            OUT (LCD),A ;los letreros correspondientes
025E: CD 89 09          CNL LET LIS
0261: 3E 00            LD A, LIN_DOS
0263: D3 00            OUT (LCD),A
0265: 21 AE 0C          LD HL,CMMNIZ
0268: CD 96 05          CNL DESPLIEG
026B:
026B: PFOECA: LD A, 02H ;Regresa al LCD al inicio del letrero
026D: D3 00            OUT (LCD),A ;sin cambiar la DERRAM
026F: CD 89 09          CNL LET LIS
0272: ID 36 15 04          LD (IX+TOPE),4 ;Se tendrá un tope de 4
0276: ID 36 13 8C          LD (IX+DIR),8CH
027A: ID 7E 0C          LD A, (IX+PCINT2);Verifica en que opción de cambio de
027D: FE 00            CP 0 ;encuentra
027F: 20 CE            JR NZ,PCOERO
0281: 21 EE 0C          LD HL,LETRA ;Se encuentra en entradas
0284: CD 96 05          CNL DESPLIEG ;y prepara lo necesario
0287: 21 02 14          LD HL,NUMENT ;para poder hacer los

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

028A: 18 20          JR PPTCCON      ;cambios
028C: FE 01          PPNCCERO: CP 1
028E: 20 0B          JR NZ, PPNOUNO ;Se encuentra en salidas
0290: 21 E8 0C          LD HL, LESALID ;y prepara lo necesario
0293: CD 96 05          CALL DESPLIEB ;para poder hacer los
0296: 21 01 14          LD HL, NUMPAL ;cambios
0299: 18 11          JR PPTCCON
029B: FE 02          PPNOUNO: CP 2
029D: 20 29          JR NZ, PPNCCOS ;Se encuentra en conjuntos
029F: DD 36 15 08       LD (IX+TOPE) ,8 ;donde el tope es de 8 y
02A3: 21 FE 0C          LD HL, LCONJU  ;prepara lo necesario para
02A6: CD 96 05          CALL DESPLIEB ;poder hacer los cambios
02A9: 21 03 14          LD HL, NUMCON
02AC:
02AC: CD BF 01          PPTCCON: CALL TECLCAN ;En este apartado se lleva a cabo
02AF: DD 7E 0D          LD A, (IX+TECLAT) ;el proceso de cambio de las variables
02B2: FE 04          CP ENTER      ;o bien la navegación entre ellas
02B4: 28 06          JR Z, PPNENOS
02B6: DD 34 0C          PPNAS: INC (IX+PCONT2)
02B9: DD 34 0C          PPNENOS: INC (IX+PCONT2)
02BC: DD 7E 0C          LD A, (IX+PCONT2)
02BE: FE 00          CP 0
02C1: 28 A8          JR Z, PPNBCA
02C3: DD 35 0C          DEC (IX+PCONT2)
02C5: 18 A3          JR PPNBCA
02C8:
02C8: FE 03          PPNCCOS: CP 3 ;En este caso el proceso cambia ligeramente
02CA: 20 0E          JR NZ, PPNOTRES ;ya que es el proceso de cambio de conjuntos
02CC: 21 0E 0D          LD HL, AJENTR ;de entrada
02CE: CD 96 05          CALL DESPLIEB
02D0: 18 17          JR PPTCCDIF
02D4: FE 04          PPNOTRES: CP 4
02D6: 20 08          JR NZ, PPNDA
02D8: 21 1E 0D          LD HL, ANSALI ;Cambio de conjuntos de salida
02DB: CD 96 05          CALL DESPLIEB
02DE: 18 0B          JR PPTCCDIF
02E0: FE 05          PPNDA: CP 5
02E2: CD BF 04          JR NZ, PPNREG
02E5: 21 2E 0D          LD HL, AREG   ;Cambio de reglas
02E9: CD 96 05          CALL DESPLIEB
02EB:
02EB: CD 99 08          PPTCCDIF: CALL DESB254 ;Desactiva salida
02EE: DB 40          PPTCCOUT: IN A, (TECLA) ;Bajera para la desactivación de cualquier tecla
02F0: E6 0F          AND FILTRAT
02F2: 20 FA          JR NZ, PPTCCOT
02F4:
02F4: CD 00 0A          PPTCCIN: CALL TIEMPO_1
02F7: DB 40          IN A, (TECLA) ;Obtiene el dato del teclado
02F9: E6 0F          AND FILTRAT ;quitando los cuatro últimos bits
02FB:
02FB: FE 08          CP SELEC     ;Checa si se trata de alguna tecla de control
02FD: 28 B7          JR Z, PPNAS ;si no es así incrementa o decreceta
02FF: FE 04          CP ENTER
0301: 28 B9          JR Z, PPNENOS
0303: FE 02          CP NMS
0305: 28 06          JR Z, PPNJUS
0307: FE 01          CP MNOS
0309: 28 02          JR Z, PPNJUS
030B: 18 B7          JR PPTCCIN
030D:
030D: DD 7E 0C          PPNJUS: LD A, (IX+PCONT2)
0310: FE 05          CP 5
0312: 20 5D          JR NZ, PPNOM ;Verifica si se trata de las reglas
0314: DD 36 0F 00       LD (IX+CONSA),00H ;Pone el valor de 0 al contador de reglas
0318: DD 36 14 FF       LD (IX+HEXA),0FFH ;Se trata de un valor hexadecimal
031C: DD 36 15 FF       LD (IX+TOPE),0FFH ;No tiene un tope
0320: 2E 01          LD A, LINDIA ;Limpia el LCD para desplegar
0322: D3 00          CUI (LCD),A ;los letreros correspondientes
0324: 21 6E 0D          LD HL, AJREG
0327: CD 96 05          CALL DESPLIEB
032A: 3E 0D          LD A, LIN DOS

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

032C: D3 00          OUT (LCD),A
032E:
032E: 21 7E 0D      LD HL,CONREG
0331: CD 96 05      CALL DESPLREG
0334:
0334: 21 A6 14      LD HL,REGLAS ;Lugar donde se tienen las reglas
0337:
0337: 3E 86          PPREG2: LD A,86H ;DORAM para los dígitos
0339: D3 00          OUT (LCD),A
033B: CD 89 09      CALL LET_LIS
033E: LD 7E 0F      LD A,(IX+CONSAL)
0341: DE F0          AND 0FH ;Aquí se realiza el despliegue del corteo
0343: 0F             RRCA ;que se lleva en hexadecimal
0344: 0F             RRCA
0345: 0F             RRCA
0346: 0F             RRCA
0347: CD 7B 09      LD A,CONVEX
034A: LD 7E 0F      LD A,(IX+CONSAL)
034D: DE 0F          AND 0FH
034F: CD 7B 09      CALL CONVEX
0352:
0352: LD 36 13 8E      LD (IX+DIR),8EH;Se prepara para hacer el cambio de reglas
0356: CD BF 01      CALL TECLACION ;Cambia la regla actual
0359:
0359: LD 7E 0D          LD A,(IX+TECLAT);Al regresar de la función
035C: FE 08          CP SELEC ;compara si se trata de salir de reglas
035E: 20 03          JR NZ,PPREG2 ;NOCA: se debe de terminar poniendo
0360: C3 1A 01      JP PPUOTREG ;como fin el código FFH
0363: DD 34 0F      PPREG3: INC (IX+CONSAL);En este caso se adelanta un lugar
0366: 23             INC HL
0367: LD 7E 0F      LD A,(IX+CONSAL)
036A: FE 00          CP 0 ;Se pueden acceder hasta 256 reglas
036C: CA 1A 01      JP Z,P2UTOREG ;
036F: 1B 05          JR PPREG2
0371:
0371: FE 04          PFCIDM4: CP 4
0373: C2 11 04      JP NZ,PPCIDM ;Se trata de cambiar los conjuntos de salida
0376: DD 36 0E 31   LD (IX+CONENT),31H;Pone el valor de 1 al contador de salida
037A: DD 36 0F 31   LD (IX+CONSAL),31H;Pone el valor de 1 al contador de singleton
037E: DD 36 15 FF   LD (IX+TOPE),0FFH;No tiene tope
0382: DD 36 14 FF   LD (IX+HEQA),0FFH;Se trata de un valor hexadecimal
0386: 3E 01          LD A,LIMETA ;Limpia el LCD para desplegar
0388: D3 00          OUT (LCD),A ;los letreros correspondientes
038A: 21 5E 0D      LD HL,AUSAL
038D: CD 96 05      CALL DESPLREG
0390: 3E 00          LD A,LIN DOS
0392: D3 00          OUT (LCD),A
0394:
0394: DD 36 13 8E      LD (IX+DIR),8EH;DORAM para los dígitos
0398: 21 AE 0C      LD HL,CONREG
039B: CD 96 05      CALL DESPLREG
039E:
039E: 21 85 14      LD HL,SINGE1 ;Lugar donde se tiene los SINGLETON
03A1:
03A1: 3A 03 14      PPSAL1: LD A,(NUMCON) ;Se compara para no rebasar el
03A4: C6 31          AND A,31H ;el número de conjuntos
03A6: 47             LD B,A
03A7: DD 7E 0E      LD A,(IX+CONENT)
03AA: B8             CP B
03AB: 20 14          JR NZ,PPSAL2
03AD: DD 36 0E 31   LD (IX+CONENT),31H;Respectivo a contador de SINGLETON
03B1: DD 34 0F      INC (IX+CONSAL)
03B4: 3A 01 14      LD A,(NUMERAL)
03B7: C6 31          AND A,31H ;Este tipo de comparación se realiza
03B9: 47             LD B,A ;para que no exceda el límite de salidas
03BA: DD 7E 0F      LD A,(IX+CONSAL)
03BD: B8             CP B
03BE: CA 1A 01      JP Z,P2UTOREG
03C1:
03C1: 3E 81          PPSAL2: LD A,81H ;DORAM para el número
03C3: DD 00          OUT (LCD),A ;Actualiza DORAM el número actual

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

03C5: CD 89 09      CNLL LET LIS      ;de conjunto y de salida.
03C8: ED 7E 0F      LD A, (IX+CONSAL)
03CB: D3 01        OUT (LCD),A
03CD: CD 89 09      CNLL LET LIS
03D0: 3E 8C        LD A,8CH
03D2: D3 00        OUT (LCD),A
03D4: CD 89 09      CNLL LET LIS
03D7: ED 7E 0E      LD A, (IX+CONENT)
03DA: D3 01        OUT (LCD),A
03DC: CD 89 09      CNLL LET LIS
03DF:
03E1: CD EF 01      CNLL TECLACON ;Para cambiar el valor del conjunto
03E2:
03E4: ED 7E 0D      LD A, (IX+TECLAT);Al regresar de la función
03E5: FR 04        CP ENTER ;compara si se trata de regresar
03E7: 20 22        JR NZ,PPSAL3 ;un lugar
03E9: ED 7E 0E      LD A, (IX+CONENT)
03EC: FR 31        CP 31H
03EE: 28 06        JR Z,PPSALA
03F0: ED 35 0E      DEC (IX+CONENT);Regresa un lugar atrás
03F3: 2B          DEC HL
03F4: 1B CB        JR PPSAL2
03F6: ED 7E 0F      PPSALA: LD A, (IX+CONSAL)
03F9: FR 31        CP 31H
03FB: 28 04        JR Z,PPSAL2
03FD: ED 35 0F      DEC (IX+CONSAL)
0400: 1A 03 14      LD A, (INMCON)
0403: CE 30        AND A,30H
0405: ED 77 0E      LD (IX+CONENT),A
0408: 2B          DEC HL
0409: 1B B6        JR PPSAL2
040B:
040E: ED 34 0E      PPSAL3: INC (IX+CONENT);En este caso se adelanta un lugar
0410: 23          INC HL
0412: 1B 90        JR PPSALI
0414:
0416: ED 36 0E 31 PPSO3: LD (IX+CONENT),31H;Pone el valor de 1 al contador de entradas
0418: ED 36 10 31 LD (IX+CONCON),31H;Pone el valor de 1 al contador de conjuntos
0419:
041B: ED 36 15 FF      LD (IX+TOPE),OFFH;No tiene tope
041D: ED 36 14 FF      LD (IX+HECO),OFFH;Se trata de un valor hexadecimal
0421: 3E 01          LD A,LIMPIA ;limpia el LCD para desplegar
0423: D3 00          OUT (LCD),A ;los letreros correspondientes
0425: 21 3E 0D      LD HL,AJCSN1
0428: CD 96 05      CNLL DESPLEG
042B: 3E CD          LD A,LIN DOS
042D: D3 00          OUT (LCD),A
042F:
0432: 21 4E 0D      LD HL,AJCSN2
0432: CD 96 05      CNLL DESPLEG
0435: 21 04 14      LD HL,CLPLE1
0438:
043B: ED 36 11 31 PPSN1: LD (IX+CONPUN),31H;Inicia con 1 el contador de puntos
043C: 1A 03 14      LD A, (INMCON)
043F: CE 31          AND A,31H
0441: 47            LD B,A
0442: ED 7E 10      LD A, (IX+CONCON);Compara el valor de conjuntos
0445: B8            CP B
0446: 20 14          JR NZ,PPSNE2
0448: ED 36 10 31 LD (IX+CONCON),31H;Reestablece a 1 el contador de conjuntos
044C: ED 34 0E      INC (IX+CONENT);Incrementa contador de entradas
044F: 3A 02 14      LD A, (INMENT)
0452: CE 31          AND A,31H ;Este tipo de comparación se realiza
0454: 47            LD B,A ;para que no exceda el límite de entradas
0455: ED 7E 0E      LD A, (IX+CONENT)
0458: B8            CP B
0459: CA 1A 01      JP Z,FMOTORB
045C:
045C: 3E 81          PPSNE2: LD A,81H ;IERRM para el número
045D: D3 00          OUT (LCD),A ;Actualiza IERRM el número actual
0460: CD 89 09      CNLL LET LIS

```



Control de temperatura con lógica difusa

```

0463: DD 7E 0E          LD A, (IX+CONENT);Despliega el número de entrada
0466: D3 01             OUT (LCD+1),A
0468: CD 89 09          CALL LET_LIS
046B: 3E 83             LD A,83H
046D: D3 00             OUT (LCD),A
046F: CD 89 09          CALL LET_LIS
0472: DD 7E 10          LD A, (IX+CONCON);Despliega el número de conjunto
0475: D3 01             OUT (LCD+1),A
0477: CD 89 09          CALL LET_LIS
047A: CD F7 04          CALL FONFUN ;Despliega los puntos correspondientes
047D: DD 36 13 88       LD (IX+DIR),88H;Prepara el lugar para el primer punto
0481: 18 26              JR PFEPS
0483: DD 7E 11 PFEFG:   LD A, (IX+CONFUN)
0486: FE 32             CP 32H
0488: 20 06             JR NZ, *+8
048A: DD 36 13 8E       LD (IX+DIR),8EH;Prepara el lugar para el segundo punto
048E: 18 19              JR PFEPS
0490: FE 33             CP 33H
0492: 20 06             JR NZ, *+8
0494: DD 36 13 C3       LD (IX+DIR),0C3H;Prepara el lugar para el tercer punto
0498: 18 0F              JR PFEPS
049A: FE 34             CP 34H
049C: 20 06             JR NZ, *+8
049E: DD 36 13 C9       LD (IX+DIR),0C9H;Prepara el lugar para el cuarto punto
04A2: 18 05              JR PFEPS
04A4: DD 34 10          INC (IX+CONCON)
04A7: 18 8F              JR PFEHL
04A9: ;
04A9: CD BF 01 PFEFS:  CALL TECLACON ;Cambia el punto correspondiente
04AC: ;
04AC: DD 7E 0D          LD A, (IX+TECLAT);Al regresar de la función
04AF: FE 04             CP ENTER ;compara si se trata de avanzar
04B1: 20 06             JR NZ, PFEHD ;si conjunto
04B3: 23              INC HL ;Rancho que fue Enter se avanza un punto
04B4: DD 34 11          INC (IX+CONFUN)
04B7: 18 CA              JR PFEFG
04B9: ;
04B9: DD 34 10          PFEFD:  INC (IX+CONCON);En este caso se adelanta un conjunto
04BC: DD 7E 11          LD A, (IX+CONFUN)
04BF: 23 PFEHFL:       INC HL ;Aquí se realiza el avance de puntos
04C0: 3C              INC A ;para pasar a otro punto
04C1: FE 35             CP 35H
04C3: 20 FA             JR NZ, PFEHFL
04C5: C3 38 04          JP PFEHL
04CB: ;
04CB: DD CB 0C 6E PFEFG: BIT 5, (IX+PCONT2);En este caso se procesa parte del menú programa
04CC: 28 38              JR Z, PFCMD
04CE: DD CB 0B 1E       RR (IX+PCONT1)
04D2: 3F              CCF
04D3: DD CB 0B 16       RL (IX+PCONT1)
04D7: DD CB 0B 46       BIT 0, (IX+PCONT1)
04DB: 04 CB 00          CALL NZ, HELGRAM
04DE: CA 3D 00          CALL NZ, HELGCHP
04E1: CA 1A 00          CALL Z, RWMSLK
04E4: 38 09              JR PFEIG
04E6: DD CB 0B 16 PFCMD: RL (IX+PCONT1)
04EA: 3F              CCF
04EB: DD CB 0B 1E       RR (IX+PCONT1)
04EF: DD 36 0C 00 PFEIG: LD (IX+PCONT2),00H;Quita banderas
04F3: CD 1A 05          CALL LET_FUN
04F6: C9              RET
04F7: ;
04F7: ;***** Función que despliega los puntos de un conjunto *****
04F7: ;
04F7: DD 36 13 88 FONFUN: LD (IX+DIR),88H;En este caso se ponen los lugares
04FB: CD 3A 09          CALL IHESA ;correspondientes de los puntos
04FE: 23              INC HL ;y se despliegan
04FF: DD 36 13 8E       LD (IX+DIR),8EH
0503: CD 3A 09          CALL IHESA ;No hay cambio alguno en ellos
0506: 23              INC HL
0507: DD 36 13 C3       LD (IX+DIR),0CH

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

0508: CD 3A 09      CALL DHEGA
0508: 23           INC HL
050F: DD 36 13 C9   LD (DX+DIR),0C9H
0513: CD 3A 09      CALL DHEGA
0516: 2B           DEC HL
0517: 2B           DEC HL
0518: 2B           DEC HL
0519: C9           RET
051A:
051A: ,*****FUNCION LEITERO PRINCIPAL*****
051A:
051A: DD 36 16 01 LET_PRIN: LD (DX+CONFP2),1
051E: 21 3E OC      LD HL,PRINC12 ;Lugar del letrero principal 2
0521: DD CB 08 7E   BIT 7,(DX+FRONT1);Verifica el despliegue actual
0525: 20 03         JR NZ,FRINC12
0527: 21 2E OC      LD HL,PRINC11 ;Lugar del letrero principal 1
052A: DD 7E 0A      PRINC12: LD A,(DX+B_M)
052D: DD 77 09     LD (DX+B_T),A
0530: 18 03         JR SALIDA
0532:
0532: ,*****FUNCION LEITERO B_T*****
0532:
0532: 21 AE 08 LET_B_T: LD HL,MODO ;Lugar del letrero de modo
0535: F5           PUSH AF
0536: 3E 01         LD A,LDMPIA ;Limpia el LCD
0538: DD 00         OUT (LCD),A
053A: CD 96 05      CALL DESPLIEG
053D: 3E CD      LD A,LIN_DOS ;Pasa a la línea dos del LCD
053F: DD 00         OUT (LCD),A
0541: DD 7E 09     LD A,(DX+B_T) ;Carga en A el reg. B_T
0544: DD CB OC 46   BIT 0,(DX+FRONT2)
0548: 20 06         JR NZ,SISELETC
054A: DD CB 08 7E   BIT 7,(DX+FRONT1);En este lugar se verifica el tipo de
054E: 20 2D         JR NZ,NOSELETC ;despliegue actual
0550: FE 01         CP 01H ;En el siguiente proceso
0552: 20 05         JR NZ,SIG1 ;corresponde al primer tipo
0554: 21 BE 08 AUTO: LD HL,AUTOMBTI ;de despliegue donde solo se
0557: 18 1F         JR DEFP ;tiene la temperatura interna en el LCD
0559: FE 02         CP 02H
055B: 20 05         JR NZ,SIG2
055D: 21 DE 08     LD HL,MANUAL
0560: 18 16         JR DEFP
0562: FE 04         CP 04H
0564: 20 05         JR NZ,SIG3
0566: 21 FE 08     LD HL,INACTIVO
0569: 18 0D         JR DEFP
056B: FE 08         CP 08H
056D: 28 06         JR Z,SIG4
056F: DD 36 09 01  LD (DX+B_T),01H
0573: 18 EF         JR AUTO
0575: 21 1E OC     SIG4: LD HL,PROGRAMA
0578: CD 96 05     DESP: CALL DESPLIEG
057B: F1           POP AF
057C: C9           RET
057D: FE 01         NOSELETC: CP 01H ;En este caso se procesan los letreros
057F: 20 05         JR NZ,SIGLIND ;que corresponden al segundo tipo
0581: 21 CE 08     LD HL,AUTOLIND ;de despliegue donde se visualiza
0584: 18 F2         JR DEFP ;tanto la temperatura interna como externa
0586: FE 02         CP 02H
0588: 20 05         JR NZ,SIGLIND
058A: 21 BE 08     LD HL,MANULIND
058D: 18 E9         JR DEFP
058F: FE 04         SIGLIND: CP 04H
0591: 21 CE OC     LD HL,INACTIND
0594: 18 E2         JR DEFP
0596:
0596: ,*****FUNCION DESPLIEG*****
0596:
0596: C5           DESPLIEG: PUSH BC ;-----
0597: 01 01 10     LD BC,LCD*100H;-- Esta función se encarga --
059A: CD 89 09     OTRA: CALL LET_ILS ;-- de mandar los datos de --

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

059D: ED A3          OUTI ;-- un letrero la LCD.  --
059F: 20 F9          JR NZ,OTRA ;-----
05A1: CD 89 09       CALL LET_LIS
05A4: C1             FOF BC
05A5: C9             RET
05A6:
05A6: ;***** FUNCION LETTEC *****
05A6:
05A6: 21 4E 0C LETTEC: LD HL,CMBIAR ;Lugar del letrero "CAMBIAR DE"
05A9: 3E 01          LD A,LIMPIA ;Limpia el LCD
05AB: D3 00          OUT (LCD),A
05AD: CD 96 05       CALL DESPLIEG
05B0: 3E 0C          LD A,LIN DOS ;Pasa a la linea dos del LCD
05B2: D3 00          OUT (LCD),A
05B4: DD CB 0C 76     BIT 6,(IX+PCNT2);-----
05B8: 20 0F          JR NZ,TN91 ;-- En las siguientes comparaciones --
05BA: DD CB 0C 6E     BIT 5,(IX+PCNT2);-- se realiza la filtración de --
05BE: 20 1F          JR NZ,TN91 ;-- Letreros para el menu PROGRAMA --
05C0: DD CB 0C F6     SET 6,(IX+PCNT2);-- en forma ciclica. --
05C4: 21 6E 0C       LD HL,RANDOS ;-----
05C7: 18 1D          JR TTELET
05C9: DD CB 0C EE TN91: SET 5,(IX+PCNT2)
05CD: DD CB 0C B6     RES 6,(IX+PCNT2)
05D1: 21 8E 0C       LD HL,PROFRE
05D4: DD CB 0B 46     BIT 0,(IX+PCNT1)
05D8: 20 0C          JR NZ,TTELET
05DA: 21 7E 0C       LD HL,FREERO
05DD: 18 07          JR TTELET
05DF: DD CB 0C AE TNV1: RES 5,(IX+PCNT2)
05E3: 21 5E 0C       LD HL,DESPLIE
05E6: CD 96 05       TTELET: CALL DESPLIEG
05E9: C9             RET
05EA:
05EA: ;***** FUNCION CON TIEMPO *****
05EA:
05EA: DD 36 07 50 FCN_TIE: LD (IX+VART1),50H;Esta función se encarga de poner el tiempo
05EE: DD 70 08          LD (IX+VART2),B;necesario para retardar un letrero desplegado,
05F1: C9             RET ;con el registro B se puede extender el tiempo.
05F2:
05F2: ;***** FUNCION CONTROL *****
05F2:
05F2: CD FF 05 CONTROL: CALL ACTENT ;Llama a actualizar entradas
05F5: CD E0 06         CALL FUZY ;Llama a convertir valores reales a difusos
05F8: CD 6E 07         CALL EREGLAS ;Llama a evaluar reglas
05FB: CD A4 07         CALL SALIDA ;Llama a determinar el estado de las salidas
05FE: C9             RET
05FF:
05FF: ;***** ACTUALIZA EL VALOR DE LAS ENTRADAS *****
05FF:
05FF: DD 35 16 ACTENT: DEC (IX+CNFF2);Aqui se utiliza un retardo para definir
0602: CD          RET NZ ;una constante de tiempo
0603: CD 45 08         CALL SNA/254
0606: DD 36 16 40     LD (IX+CNFF2),40H;para la actualización de las entradas
060A: F5             PUSH AF
060B: C5             PUSH BC
060C: D5             PUSH DE
060D: E5             PUSH HL
060E: 21 52 12        LD HL,BLKENT*2
0611: 3A 00 20         LD A,(INCA) ;Actualiza el valor de temperatura, el
0614: 77             LD (HL),A ;valor de temperatura lo pone BLKENT*2
0615: 47             LD B,A ;B=Tactual
0616: 3A 00 14        LD A,(TMESEL);A=Tuelec
0619: 90             SUB B ;A=Tuelec-Tactual
061A: 5F             LD E,A ;E=Error2
061B: 16 00           LD D,0H ;D=Se encarga de definir que signo tiene Error2
061D: 3A 00 02        LD A,ACTUNO ;Se compara si hay acarreo
061F: 16 FF           LD B,D,OFH ;Si D=FFH nos indica que es negativo E2 por lo
0621: CD 83 06 ACTUNO: CALL AJUSTA ;tanto Tactual*Tuel.
0624: 32 50 12        LD (BLKENT),A ;A=BLKENT
0627:
0627: 7B             LD A,E ;A=Error2

```

**TESIS CON FALLA DE ORIGEN**

Control de temperatura con lógica difusa

```

0628: ID 4E 17          LD C, (IX+ERROR) ;O>Error1
062B: 91                SUB C                ;A>Error2-Error1
062C: 07                RLCA
062D: 07                RLCA
062E: 07                RLCA
062F: ID 73 17          LD (IX+ERROR),B;Se pasa el Error2->Error1
0631: F5                PUSH AF
0633: ID 7E 18          LD A, (IX+ERROR);Retira el signo del error anterior
0636: BA                CP D
0637: 28 14            JR Z,ACTDOS ;Si es así salta, si no entonces compara
0639: FE 00            CP 0
063B: 20 08            JR NZ,ACTCUA ;si no es así entonces salta
063D: F1                POP AF
063E: 17                SCF ;pone la bandera de acarreo en alto
063F: ID 36 18 FF       LD (IX+ERRORS),OFFH;Se pone la bandera de negativo
0643: 18 09            JR ACTRES
0645: F1                ACTCUA: POP AF ;
0646: E7                OR A ;Con esto se logra limpiar la bandera de acarreo
0647: ID 36 18 00       LD (IX+ERRORS),OH;pone la el signo como positivo
064B: 18 01            JR ACTRES
064D: F1                ACTDOS: POP AF
064E: CD 83 06          ACTRES: CALL AJUSTA ;Se pone la derivada del error
0651: 32 51 12          LD (BLUNT+1),A
0654:
0654: 3A 00 30          LD A, (ADC2) ;Actualiza el valor de ADC2
0657: 32 53 12          LD (BLUNT+3),A
065A:
065A: ED C1 0C 46       BIT 0, (IX+PCONT2)
065E: 20 1E            JR NZ,ACTFIN
0660: ED CB 0B 7E       BIT 7, (IX+PCONT1)
0664: 20 09            JR NZ,ACTLET
0666: ED 36 13 8C       LD (IX+DIR),8CH;Actualiza despliegue de letreiro 1
066A: CD EF 08          CALL DEINBCD
066D: 18 0F            JR ACTFIN
066F:
066F: ID 36 13 C2          ACTLET: LD (IX+DIR),0C2H;Actualiza despliegue de letreiro 2
0673: CD EF 08          CALL DEINBCD
0676: ID 36 13 CC       LD (IX+DIR),0CCH
067A: 23                INC HL
067B: CD EF 08          CALL DEINBCD
067E:
067E: ACTFIN: LD (IX+TOPE),OFFH ;*****ES PARA DEBUG
067E: ;LD HL,BLUNT ;*****ES PARA DEBUG
067E: ;LD (IX+DIR),80H ;*****ES PARA DEBUG
067E: ;CALL DEBGA ;*****ES PARA DEBUG
067E: ;INC HL ;*****ES PARA DEBUG
067E: ;LD (IX+DIR),0C0H ;*****ES PARA DEBUG
067E: ;CALL DEBGA ;*****ES PARA DEBUG
067E:
067E: E1                POP HL
067F: D1                POP DE
0680: C1                POP EC
0681: F1                POP AF
0682: C9                RET
0683: ;***** FUNCION QUE AJUSTA LOS VALORES DE LAS ENTRADAS*****
0683: ;***** PRGA COMO REFERENCIA "A" Y "B" RESULTADO EN "A"*****
0683: 4F                AJUSTA: LD C,A
0684: 38 0B            JR C,AJUNB
0686: D6 80            SUB 80H ;Checa que el valor no exceda de 80H
068B: 30 04            JR NC,AJUNO
068A: 79                AJDOS: LD A,C
068B: 06 80            ADD A,80H
068D: C9                RET
068E: 3E FF           AJUNO: LD A,OFFH
0690: C9                RET
0691: C6 80           AJUNB: ADD A,80H
0693: 4F                LD C,A
0694: D6 80            SUB 80H
0696: 38 03            JR C,AJUNRES
0698: 3E 00           LD A,OH
069A: C9                RET

```

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

069B: 79          AJUTRES: LD A,C
069C: C9          RET
069D:
069D:              ;***** PROGRAMA QUE CALCULA LAS PENDIENTES DE LOS CONJUNTOS *****
069D: 3A 02 14      PENDIENT: LD A,(NUMENT) ;Calcula el número de pendientes que
069D: 07            FICR, resta dado por el número de entradas
069D: ED 77 00      LD IX,(VAR1L),Nº número de conjuntos.
069D: 3A 03 14      069A: LD A,(NUMCON) ;pendientes = conjuntos*2*entradas
069D: DD 77 02      LD (IX+VAR2L),A
069D: CD CF 09      CALL MEXIT
069D: DD 46 04      LD B,(IX+VAR3L);# de pendientes -> B
069D: FD 21 54 12   069B: LD IX,(MEXI) ;Lugar donde estarán las pendientes calculadas
069D: 21 05 14      LD HL,(MEXI)+1 ;Lugar de lo partes de los conjuntos
069D: DD 36 02 FF   06B7: LD (IX+VAR2H),OFFH;inicia los valores de la división
069D: DD 36 03 00   06B8: LD (IX+VAR2H),OH
069D: DD 36 01 00   06B9: LD (IX+VAR2H),OH
069D: 7E            LOOP1: LD A,(HL) ; A = P2 * P4
069D: 2B            DEC HL ; HL se coloca en el punto anterior
069D: 96            SUB (HL) ;VALOR DEL DIVIDENDO EN A
069D: ED 77 00      LD (IX+VAR4L),A;Pasa el valor a la división
069D: CD 52 09      CALL DIV
069D: DD 7E 04      LD A,(IX+VAR3L); A = (P2 - P1)/FF * A = (P4 - P3)/FF
069D: FE 00         CP OH ;Compara si los puntos son iguales el resultado
069D: 20 02         JR NZ,PENDCERO ;ser FFH
069D: 3E FF         LD A,OFFH
069D: FD 77 00      PENDCERO: LD (IX+0),A ;Se coloca el resultado donde corresponde
069D: 23            INC HL ;-----
069D: 23            INC HL ;Se coloca en el sig. conjunto
069D: 23            INC HL ;-----
069D: FD 23         INC IX ;Siguiente lugar de pendiente
069D: 10 E4         DINEZ LOOP1 ;? están todas las pendientes
069D: C9            RET ; SI -> retorna
069D:
069D:              ;***** PROGRAMA QUE EFECTUA LA FUSIFICACIÓN *****
069D: 75            FUZY: FUEH AP ;-----
069D: C5            PUSH BC ; Retiene valores
069D: D5            PUSH DE ; anteriores
069D: E5            PUSH HL ;-----
069D: FD 21 04 14   069A: LD IX,(MEXI) ;PUNTO 1 DEL CONJUNTO 1 Y ENTRADA 1
069D: 21 54 12      LD HL,(MEXI) ;Lugar de las pendientes calculadas
069D: 11 00 12      LD DE,(FUZIN) ;Lugar en donde se pone el valor difuso
069D: D9            EOX
069D: 21 50 12      LD HL,(LURET) ;HL' Tendrá la dir para el BLOQUE DE ENTRADAS
069D: D9            EOX
069D: 3A 02 14      06F2: LD A,(NUMENT) ; # de entradas
069D: 4F            FUZYB: LD C,A
069D: D9            EOX
069D: 7E            LD A,(HL)
069D: 23            INC HL
069D: D9            EOX
069D: DD 77 06      06F8: LD (IX+GEN1),A ;Variable que utiliza valós para calcular
069D: 3A 03 14      LD A,(NUMCON)
069D: 0701: 47            LD B,A ;# de conjuntos
069D: CD 24 07      FUZYVAL: CALL VALUA ;Función que realiza el calculo correspondiente
069D: 12            LD (DE),A ;En A se encuentra el Min(P1,P2,FF)
069D: 13            INC DE ;lo pone en FUZIN e incrementa el lugar
069D: FD 23         INC IX ;-----
069D: FD 23         INC IX ;IX se ubica en el conjunto
069D: FD 23         INC IX ;siguiente
069D: 10 F1         DINEZ FUZYVAL ;Checa si ya se realizó para los conjuntos
069D: 3A 03 14      0711: LD A,(NUMCON)
069D: FE 08         FUZYCFA: CP 8 ;Se posiciona en el lugar correcto de
069D: 28 04         JR Z,FUZYFOS ;los resultados en FUZIN
069D: 3C            INC A
069D: 13            INC DE
069D: 18 FB         JR FUZYCFA
069D: 0D            FUZYFOS: DEC C ;Checa si ya se realizó para las entradas
069D: 20 D8         JR NZ,FUZYB

```

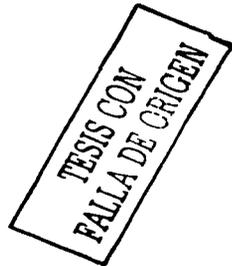
TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

```

071P: R1          FOP HL          ;-----
0720: D1          FOP DE          ;Recupera los valores anteriores
0721: C1          FOP BC          ;de los registros
0722: P1          FOP AF          ;-----
0723: C9          RET ;Regresa
0724:
0724:
;..... FUNCION VALIA .....
0724: C5          VALIA:  FUEH BC
0725: D5          FUEH DE
0726: DD 06 06    LD D, (IX+06H) ;Recupera valor de la entrada
0729: FD 78 03    LD A, (IX+P4) ;Se coloca en el punto 4
072C: 92          SUB D          ; A = P4 - ADC
072D: 38 38      JR C, VALNEG ; Si es negativo Termina
072E: 5F          LD B, A          ; A -> E
0730: 78          LD A, D
0731: FD 96 00    SUB (IX+0) ; A = ADC - P1
0734: 38 31      JR C, VALNEG ; Si es negativo Termina
0736: 06 02      LD B, 02H ;Por ser dos pendientes por conjunto
0738: DD 36 01 00 LD (IX+VARGH), 02H; Variables de la mult. sin utilizar
073C: DD 36 03 00 LD (IX+VARGH), 03H; Variables de la mult. sin utilizar
0740: DD 77 00    VAL2:  LD (IX+VARGL), A; Valor del incremento
0741: 78          LD A, 04H ;Valor de la pendiente
0744: DD 77 02    LD (IX+VARGL), A
0747: CD CF 09    ONLJ MULTI ;Realiza la multiplicación
074A: DD 78 05    LD A, (IX+VARGH) ;-----
074D: FE 00      CP 0          ;Verifica si el resultado es > FFH
074E: 28 04      JR Z, VAL0 ;-----
0751: DD 36 04 FF VAL0:  LD (IX+VARGL), 0FFH; Si es > FFH coloca FFH
0755: DD 48 04    LD C, (IX+VARGL); Resultado -> C
0758: 21          INC HL          ;Sig. pendiente
0759: 78          LD A, E
075A: 59          LD E, C
075B: 10 E3      DINEZ VAL2 ;? para las dos pendientes
075D: 47          LD B, A
075E: 93          SUB E          ;-----
075F: 30 03      JR NC, MENE ; Obtiene el valor
0761: 78          LD A, B          ; mínimo de los
0762: 18 07      JR VALFIN ; productos de los
0764: 78          MENE:  LD A, E          ; incrementos y
0765: 18 04      JR VALFIN ; pendientes
0767: 38 00      VALNEG: LD A, 00H ; Valor cero si in incremento es (-)
0769: 23          INC HL
076A: 23          VALFIN: FOP DE
076B: D1          FOP BC
076D: C9          RET
076E:
;..... FUNCION QUE EVALUA REGLAS .....
076E:
076E: F5          REGLAS: FUEH AF
076F: C5          FUEH BC
0770: D6          FUEH DE
0771: E5          FUEH HL
0772:
0772: 06 20      LD B, 20H ;Este bloque se encarga de borrar las
0774: 21 20 12    LD HL, FUZOUT ;reglas evaluadas en ocasiones anteriores
0777: 38 00      LD A, 0
0779: 77          BORRARREG: LD (HL), A
077A: 23          INC HL
077B: 10 FC      DINEZ BORRARREG
077D:
077D: 26 65      LD H, FUEH ;Lugar de FUZIN y FUZOUT (Parte alta)
077F: 21 26 14    LD DE, REGLAS ;Dirección donde están las reglas
0782: 06 FF      REGAL:  LD B, 0FFH ;Establece como mínimo el valor FFH
0784: 1A          REGIC2: LD A, (DE)
0785: C8 7F      BIT 7, A ;Carga la regla en curso
0787: 28 05      JR Z, REGIO ;? fin de las reglas
0789: E1          JR Z, REGIO ;NO continúa con la evaluación
078A: D1          FOP HL
078B: C1          FOP BC

```



Control de temperatura con lógica difusa

```

078C: F1          FOP AP
078D: C9          REP
078E: 13          RSIG:  INC DE      ;Se posiciona a la siguiente regla
078F: 6F          LD L,A      ;Se completa la dirección (HL)
0790: C8 6F      BIT 5,A      ;? antecedente o consecuente
0792: 28 08      JR Z,RES0   ;Salta si es antecedente
0794: 79          LD A,C
0795: FE 00      CP 0
0797: 28 E9      JR Z,RSAL   ;En consecuente por lo que se pone el resultado
0799: 77          LD (HL),A   ;Siguiente regla
079A: 18 E5      JR RSAL
079C: 7E          RES0:  LD A,(HL) ;SE REALIZA LA Operación AND PARA LOGICA DIFUSA
079D: 90          SUB B      ;La operación AND en La lógica difusa es el
079E: 30 01      JR NC,RS0  ;obtener el mínimo y esto lo realiza
07A0: 46          LD B,(HL) ;este bloque
07A1: 48          RIG:  LD C,B
07A2: 18 E0      JR RSIG2
07A4:
07A4: ;..... FUNCION QUE EVALUA LA SALIDA .....
07A4:
07A4:
07A4: PS          SALIDA:  PUSH AP
07A5: C5          PUSH BC
07A6: D5          PUSH DE
07A7: E5          PUSH HL
07A8: FD 21 85 14 LD IX,SINGL ;Lugar de los SINGLETON de salida
07AC: 21 20 12 LD HL,PUZOUT ;Lugar donde se colocara PUZOUT
07AF: D9          EXX ;En HL se tendrá F(X)Z
07B0: 21 40 12 LD HL,FXZSSL
07B1: D9          EXX
07B4: 3A 01 14 LD A,(NUMAL)
07B7: 4F          LD C,A      ;# de salidas
07B8: 21 00 00 SALCO:  LD DE,00H ;Inicia los registros para la suma
07B9: 3A 03 14 LD A,(NUMCON)
07BC: 47          LD B,A      ;# de conjuntos
07BE: 7E 00 00 SALENO:  LD A,(IX+0) ;Apunta al lugar de Singletons
07C2: ED 77 00 LD (IX+VARIL),A
07C5:
07C5: 7E          LD A,(HL) ;Apunta al lugar de FUZOUT
07C6: ED 77 02 LD (IX+VARZL),A
07C9: CD CF 09 CALL MULAT ;SINGSIj*FUZOUTij
07CC: ED 7E 04 LD A,(IX+VARGL)
07CF: 83          ADD A,E
07D0: 5F          LD E,A
07D1: ED 7E 05 LD A,(IX+VARGH)
07D4: 8A          ADD A,D
07D6: 57          LD D,A      ;DE-SINGSIj*FUZOUTij + DE
07D8: FD 23          INC IX
07DB: 23          INC HL
07DE: 10 E4          DRCZ SALENO
07E0: 3A 03 14 LD A,(NUMCON)
07E2: FE 08          SALCPA:  CP 8 ;Se posiciona en el lugar correcto de
07E4: 28 04          JR Z,SALFOS ;FUZOUT
07E5: 3C          INC A
07E6: 23          INC HL
07E8: 18 F8          JR SALCPA
07E9: 7B          SALFOS: LD A,E ;Pone los resultados parciales a
07EB: D9          EXX ;partir de FXZSSL
07ED: 77          LD (HL),A
07EE: 23          INC HL
07E9: D9          EXX
07EB: 7A          LD A,D
07EC: D9          EXX
07ED: 77          LD (HL),A
07EE: 23          INC HL
07EF: D9          EXX
07F0: 0D          DEC C ;Checa si se realizó para todas las salidas
07F1: 20 C5      JR NZ,SALCO
07F3:
07F3: FD 21 20 12 LD IX,PUZOUT ;Hasta aquí se tiene el numerador y a
07F7: 21 40 12 LD HL,FXZSSL ;continuación se realiza la suma
07FA: D9          EXX ;del bloque para el denominador

```

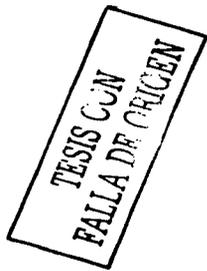


Control de temperatura con lógica difusa

```

07FB: 21 48 12          LD HL,CYCLOSI ;Aquí comienza SUMATORIA DE ZI
07FC: D9                ISX
07FF: 3A 01 14          LD A,(NUMSAL)
0802: 4F                LD C,A          ;Se coloca el numero de salida
0803: 3A 03 14          SALB: LD A,(NUMCON) ;# de conjuntos
0806: 47                LD B,A
0807: 11 00 00          LD DE,00H
080A: FD 7E 00          SALCON: LD A,(IY+0)
080D: 83                ADC A,B
080E: 5F                LD E,A
080F: 7A                LD A,D
0810: CE 00            ADC A,00H
0812: 57                LD D,A
0813: FD 23            INC IX
0815: 10 F3            DEXE SALCON
0817: 3A 03 14          SALCPA: LD A,(NUMCON)
081A: FE 08            CP 8          ;Se coloca en la correcta posición
081C: 28 05            JR Z,SALFOS1 ;de FUZZUT
081E: 3C                INC A
081F: FD 23            INC IX
0821: 1B F7            JR SALCPA
0823: ED 73 00          SALFOS1: LD (IX+VARI1),E;Se pone el resultado en un registro temporal
0826: ED 72 01          LD (IX+VARI0),D;para realizar las operaciones correspondientes
0829: 7E                LD A,(HL)
082A: ED 77 02          LD (IX+VARI2),A
082D: 23                INC HL
082E: 7E                LD A,(HL)
082F: ED 77 03          LD (IX+VARI2H),A
0832: 23                INC HL
0833: CD 52 09          CNLL DIV
0836: ED 7E 04          LD A,(IX+VARI1)
0839: D9                ISX
083A: 77                LD (HL),A    ;CYCLOSI=FUZZ1/DE
083B: 23                INC HL
083C: D9                ISX
083D: CD                INC C        ;Lo realiza para todas las salidas
083E: 20 C3            JR NZ,SALB
0840:
0840: ;LD (IX+TOR0),OFFH ;*****ES PARA DEBUG
0840: ;LD HL,CYCLOSI ;*****ES PARA DEBUG
0840: ;LD (IX+DIR),87H ;*****ES PARA DEBUG
0840: ;CNLL DHE0A ;*****ES PARA DEBUG
0840: ;INC HL ;*****ES PARA DEBUG
0840: ;LD (IX+DIR),0CH ;*****ES PARA DEBUG
0840: ;CNLL DHE0A ;*****ES PARA DEBUG
0840:
0840: EI                POP HL
0841: DI                POP DE
0842: CI                POP BC
0843: FI                POP AF
0844: CS                RST
0845:
0845: ;*****FUNCION SALB254 *****
0845:
0845: PS                SALB254: PUSH AF
0846: CS                PUSH BC
0847: DS                PUSH DE
0848: FD ES            PUSH IX
084A: ED 7E 0A          LD A,(IX+B_M)
084D: FE 04            CP 04H        ;Verifica si esta en inactivo
084F: 20 05            JR NZ,CORNC
0851: CD 99 08          CNLL DESB254 ;Si esta en inactivo anula las salidas
0854: 1B 3D            JR CORNRS0 ;el sistema no se altere
0856:
0856:
0856: ED 7E 0A          CORNC: LD A,(IX+B_M) ;Verifica si esta manual
0859: FE 02            CP 02H
085B: 28 10            JR Z,CORNSIG
085D: 06 04            LD B,04H
085F: FD 21 48 12          LD IX,CYCLOSI
0863: FD 7E 00          CORNRS: LD A,(IY+0)

```



Control de temperatura con lógica difusa

```

0866: FD 77 04          LD (IX+CONRUX),A
0869: FD 23             INC IX
086B: 10 FE           DIZC CONREC
086D:
086D:
086D: FD 21 4C 12 CONSIG: LD IX,CICLOSL+CONRUX
0871: 3E 00           LD A,0
0873: FD 46 00       LD B,(IX*0) ;Valor inicial para el cont2
0876: 90             SUB B
0877: FE 00           CP OH
0879: 20 02         JR NZ,SALINO
087B: 3E FF         LD A,OFFH
087D: D3 82       SALINO: OUT (P8254*2),A
087F: 3E 00         LD A,0
0881: D3 82       OUT (P8254*2),A
0883:
0883: FD 46 01       LD B,(IX*1) ;Valor inicial para el cont1
0886: 90             SUB B
0887: FE 00           CP OH
0889: 20 02         JR NZ,SALDOS
088B: 3E FF         LD A,OFFH
088D: D3 81       SALDOS: OUT (P8254*1),A
088F: 3E 00         LD A,0
0891: D3 81       OUT (P8254*1),A
0893:
0893: FD E1         CONREG: FOP IX
0895: D1             FOP DE
0896: C1           FOP BC
0897: F1         FOP AF
0898: C9         RET
0899:
0899: ;***** DESR254 *****
0899: F5         DESR254: PUSH AF
089A: 3E 3E       LD A,MRDCIN0 ;Programa el Contador1 como generador de onda
089C: D3 83       OUT (P8254*3),A;comenzada a partir de la frecuencia del sistema
089E: 3E 74       LD A,DIV255 ;para que entren 255 divisiones en el periodo
08A0: D3 80       OUT (P8254),A ;de media onda senooidal de la fuente
08A2: 3E 00       LD A,0
08A4: D3 80       OUT (P8254),A
08A6:
08A6: 3E BA       LD A,MRDCIN2 ;Programa el Contador2 como disparado por señal
08A8: D3 83       OUT (P8254*3),A;de hardware
08AA: 3E FF         LD A,DIVINI ;Valor inicial para el cont2****(111111)
08AC: D3 82       OUT (P8254*2),A
08AE: 3E 00       LD A,0
08B0: D3 82       OUT (P8254*2),A
08B2:
08B2: 3E 7A       LD A,MRDCIN3 ;Programa el Contador2 como disparado por señal
08B4: D3 83       OUT (P8254*3),A;de hardware
08B6: 3E FF         LD A,DIVINI ;Valor inicial para el cont2****(111111)
08B8: D3 81       OUT (P8254*1),A
08BA: 3E 00       LD A,0
08BC: D3 81       OUT (P8254*1),A
08BE: F1         FOP AF
08BF: C9         RET
08C0:
08C0: ;*****FUNCIÓN DE TRANSFORMACIÓN DE BINARIO A DECIMAL*****
08C0: ;***** EN ASCII*****
08C0:
08C0: ID 77 00     BINECO: LD (IX+VARLL),A;Factor uno
08C3: ID 70 02     LD (IX+VARZL),B;Factor dos
08C5: CD CF 09     CALL MLTIT
08C9: ID 7E 05     LD A,(IX+VARRH);A = A*B
08CC: ID 36 05 00 LD (IX+VARRH),00H
08D0: D6 0A     BINSIG: SUB 10
08D2: 38 05     JR C,MENLO
08D4: ID 34 05     INC (IX+VARRH)
08D7: 2B FF     JR BINSIG
08D9: C5 2A     MENLO: ADD A,30H;10 ;Valor en ASCII
08DB: ID 77 04     LD (IX+VARLL),A
08DE: ID 7E 05     LD A,(IX+VARRH)

```

**TESIS CON  
FALLA DE ORIGEN**

Control de temperatura con lógica difusa

```

08E1: FE 00          CP 0
08E3: 20 04          JR NZ,BINND
08E5: 3E FE          LD A,0FFH
08E7: 18 02          JR BINNDLR          ;Las unidades en ASCII se encuentran en
08E9: C9 30          BINND: ADD A,30H          ;(IX+VAR3L), las decimas en (IX+VAR3H)
08EB: ID 77 05          BINCLAR: LD (IX+VAR3H),A
08ED: C9          RET
08EF:
;***** PROGRAMA QUE REALIZA EL DESPLIEGUE DE DATOS *****
08F0:
08F2: FE          DEINBCD: PUSH AF          ;La rutina de despliegue de datos requiere
08F4: C5          ;la dirección del dato en los registros HL
08F6: 7E          LD A,(HL)          ;Los datos son por dos dígitos que se ponen
08F8: 06 64          LD B,FACTOR          ;en el LCD
08FA: CD CD 08          CNL DEINBCD          ;El factor es por la cual hay que multiplicar
08FC:              ;el número para usar toda la escala
08FE: ID 7E 13          LD A,(IX+DIR)          ;Se accede a la dirección correspondiente
0900: ID 00          OUT (LCD),A          ;y se lee el dato actual, esto sirve
0902: CD 89 09          CNL LET LIS          ;para que posteriormente se compare tanto
0904: DB 03          IN A,(LCD+3)          ;el número leído como el calculado
0906: 57          LD D,A              ;y de esa manera verificar si hubo cambio
0908: CD 89 09          CNL LET LIS          ;en los dígitos.
090A: ID 46 04          LD B,(IX+VAR1L)
090C: 58          LD E,B
090E: BE          CP B
0910:
0912: 3E 10          LD A,HEX1L
0914: DB 00          OUT (LCD),A
0916: CD 89 09          CNL LET LIS
0918:
091A:
091C: 20 06          JR NZ,DEINSDI
091E: ID 36 06 FF          LD (IX+GENL),OFFH;Esto indica que no hubo cambio
0920: 18 04          JR DEINSDI
0922:
0924:
0926: ID 36 06 00 DEINSDI: LD (IX+GENL),0H;Indica que sí hubo cambio en el dígito
0928:
092A: 7E          DEINSDI: LD A,B
092C: ID 01          OUT (LCD+1),A          ;Realiza el despliegue de dígitos
092E: CD 89 09          CNL LET LIS
0930: 3E 10          LD A,HEX1L
0932: ID 00          OUT (LCD),A
0934: CD 89 09          CNL LET LIS
0936: ID 00          OUT (LCD),A
0938: CD 89 09          CNL LET LIS
093A: ID 7E 05          LD A,(IX+VAR3H)
093C: ID 01          OUT (LCD+1),A
093E: CD 89 09          CNL LET LIS
0940:
0942:
0944: C1          POP EC
0946: F1          POP AF
0948: C9          RET
094A:
;***** DECA *****
094C:
094E: FE          DECA: PUSH AF
0950: ID 7E 15          LD A,(IX+TOPE)          ;Verifica si no tiene tope
0952: FE FF          CP 0FFH
0954: 2E 14          JR Z,ELIERS
0956: 7E          LD A,(HL)
0958: FE 00          CP 0
095A: 20 04          JR NZ,DMAVO          ;Si tiene tope compara para no
095C: 36 01          LD (HL),1          ;sobrepasarlo
095E: 18 0B          JR ELIERS
0960:
DMAVO: SUB 1
0962: ID 96 15          SUB (IX+TOPE)
0964: 3E 04          JR C,ELIERS
0966: ID 7E 15          LD A,(IX+TOPE)
0968: 77          LD (HL),A
096A:
096C:
096E: ID 7E 13          ELIERS: LD A,(IX+DIR)          ;Aquí separa la parte alta y baja para su
0970: ID 00          OUT (LCD),A          ;transformación a ASCII

```



Control de temperatura con lógica difusa

```

095B: CD 89 09      CALL LET_LIS ;con despliegue en hexadecimal
095E: 7E           LD A, (HL)
095F: EG F0       AND OFOH
0961: 0F         RRCA
0962: 0F         RRCA
0963: 0F         RRCA
0964: 0F         RRCA
0965: CD 7B 09      CALL CONVHEX
0968:
0968: 7E           LD A, (HL)
0969: EG 0F       AND OFH
096B: CD 7B 09      CALL CONVHEX
096E:
096E: 3E 10       LD A, RETLL
0970: DD 00       OUT (LCD),A
0972: CD 89 09      CALL LET_LIS
0975: DD 36 06 00   LD (DX+GEN1),0 ;Por default hay cambio de dígito
0979: F1         POP AF
097A: C9         RET
097B:
097B: ;***** CONVHEX *****
097B: FE 0A       CONVHEX: CP 10 ;Se encarga de convertir el binario a
097D: 38 02       JR C,HEXOMEN10 ;ASCII para deslignar en hexadecimal
097F: C5 07       ADD A, 7
0981: C5 30       HEXOMEN10: ADD A,30H
0983: DD 01       OUT (LCD+1),A
0985: CD 89 09      CALL LET_LIS
0988: C9         RET
0989:
0989: ;*****FUNCIÓN DE LETRERO LISTO*****
0989: F5         LET_LIS: PUSH AF ;Esta función se dedica a esperar
098A: DB 02       ESP: IN A, (LCD+2) ;ejecutar algún comando o recibir
098C: EG 80       AND B0H ;datos verificando la bandera BUSY
098E: 20 FA       JR NZ,ESP ;del mismo.
0990: F1         POP AF
0991: C9         RET
0992:
0992: ;*****
0992: ;PROGRAMA QUE EFECTUA UNA DIVISION ENTERA DE 16BITS X 16BITS Y UN RESULTADO DE
0992: ;***** 16 BITS *****
0992:
0992: F5         DIV: PUSH AF
0993: C5         PUSH BC
0994: DB         PUSH DE
0995: EG         PUSH HL
0996: 21 00 00   LD HL,00H ;Serán usados para guardar el resultado parcial
0999: 06 11     LD B,11H ;Numero de bits más uno (16 bits + 1)
099B: DD SE 02   LD E, (DX+VAR2L);Obtiene el valor del dividendo
099E: DD 56 03   LD D, (DX+VAR2H)
09A1: CB 13     DIV1: RL B ;Rota el valor bajo del dividendo con carry
09A3: CB 12     RL D ;Rota el valor alto del dividendo con carry
09A5: 10 0B     DNEG DIV2 ;? fin de bits
09A7: DD 73 04   LD (DX+VAR3L),E;Guarda el valor final a
09AA: DD 72 05   LD (DX+VAR3H),D;La variable VAR3
09AD: E1         POP HL ;Regresa valores principales
09AE: DD         POP DE
09AF: C1         POP BC
09B0: F1         POP AF
09B1: C9         RET ;Retorna de la subrutina
09B2: CB 15     DIV2: RL L ;Rota el valor del dividendo parcial
09B4: CB 14     RL H
09B6: 7D         LD A,L ;Recupera el valor del dividendo parcial
09B7: DD 96 00   SUB (DX+VAR1L) ;y réstalo al divisor
09B8: 6F         LD L,A
09B9: 7C         LD A,H
09BC: DD 9B 01   SEC A, (DX+VAR1H)
09BE: 67         LD H,A
09C0: 30 0A     JR NC,DIVNSUM ;? existe acarreo
09C2: 7D         LD A,L ;Si hay acarreo por lo que

```



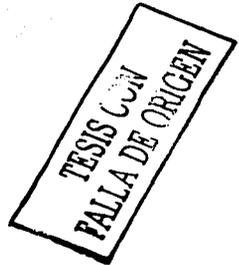
Control de temperatura con lógica difusa

```

09C3: LD R6 00          AED A, (IX+VARLL); se regresa al valor inicial
09C4: CP                LD L,A
09C7: 7C                LD A,H
09C8: LD R6 01          ADC A, (IX+VARUH)
09C8: 67                LD H,A
09C9: 3F                DIVANGUM: CCF ;Complementa el acarreo
09C9: 18 IZ            JR DIV1          ;y repite para el siguiente bit
09CF:
;*****
09CF:
;PROGRAMA QUE EFECTUA UNA MULTIPLICACION DE BITS X BITS Y UN RESULTADO DE*
09CF:
;***** 16BITS *****
09CF:
MULTI: PUSH AF          ;Se guarda el contenido de AF
09D0: CS                PUSH BC
09D1: D5                PUSH DE
09D2: E5                PUSH HL
09D3: 11 00 00          LD DE,00H          ;El resultado acumulado es igual a cero
09D6: LD 7E 00          LD A, (IX+VARLL)   ;Lugar del multiplicando
09D9: FE 00              CP 0H              ;Compara si no es cero, si lo es
09DB: 28 18            JR Z,MULRESUL      ;el resultado es cero
09DC: 4F                LD C,A             ;Multiplicando -> C
09DE: LD 7E 02          LD A, (IX+VARZL)   ;Lugar del multiplicador
09E1: FE 00              CP 0H              ;Compara si no es cero, si lo es
09E3: 28 10            JR Z,MULRESUL      ;el resultado es cero
09E5: 67                LD H,A             ;Multiplicador -> H
09E6: 06 08            LD B,08H          ;C_B = 8 (contador de bit)
09E8: CB 1C            MULTI: RR H        ;Mueve al multiplicador a la derecha con carry
09E9: 30 02            JR NC,MULANGUM     ;Checa el bit en curso
09EC: 7A                LD A,D             ;El bit en curso es "1" y se suma
09ED: 81                AED A,C            ;al resultado acumulado
09EE: 57                LD D,A
09EF: CB 1A            MULANGUM: RR D      ;Rotar el resultado acumulado
09F1: CB 1B            RR E
09F2: 10 F3            DIME MULA          ;Se realiza para los ocho bits?
09F5: LD 73 04          MULRESUL: LD (IX+VARGL);EL RESULTADO ACUMULADO ESTA EN DE
09F8: LD 72 05          LD (IX+VARGH),D;Y SE PASA A UN LUGAR DE MEMORIA
09FA: E1                POP HL             ;Regresa los valores de los registros
09FC: D1                POP DE
09FE: C1                POP BC
09FF: F1                POP AF             ;B,C,D,E,H,L,A,F
09FF: C9                RET               ;Regresa de la multiplicación
;*****TIEMPO 1 Duración APROX. 50ms*****
0A00: CS
TIEMPO 1: PUSH BC
0A01: 0E 5A            LD C,5AH          ;Tiempo que se utiliza para eliminar
0A03: 06 00            SUGS: LD B,00H    ;los rebotes de los botones.
0A05: 10 FE            BMS: DIME BMS
0A07: 00                DMC C
0A08: 20 F9            JR NC,SUGS
0A0A: C1                POP BC
0A0B: C9                RET

```

Line #	Identifier	Name	Value
959	ACTCDA		0645
963	ACTCDS		0646
921	ACTCMT		05FF
984	ACTFIN		067E
978	ACTLET		066F
964	ACTRES		064E
939	ACTUMO		0621
189	ADCI		2000
190	ADCD		3000
90	AJCSH1		000E
92	AJCSH2		004E
1003	AJCS		068A
84	AJNTR		000E
86	AJRES		006E
88	AJRESL		002E
94	AJSL		003E
86	AJSLJ		001E



Control de temperatura con lógica difusa

1008	AJUNEG	0691
1006	AJUNO	068E
999	AJUSTA	0683
1014	AJUTRES	069B
158	ARRANI	0012
837	AUTO	0554
43	AUTOINCO	0E3E
40	AUTOPATI	0E8E
103	BASEFUZ	1200
1357	BINECD	0E00
1374	BINECAP	0E8B
1373	BINEB	0E8D
1262	BINEIG	0E8D
268	BLADATP	009D
117	BLAKNT	1250
104	BLAKROM	0200
291	BLAKRAM	00C8
145	BM	000A
1557	BWE	0A05
1143	BORRAREB	0779
144	BT	0009
120	CINLEL	1254
126	CLPLEL	1404
20	CLPLELP	0E17
70	CNPAWUJ	0C3E
72	CNPAWUZ	0C3E
74	CNPAWJ	0C3E
60	CNMBLAR	0C4E
113	CYCLOS1	1248
10	COCENTER	0E04
8	COLECD	0E00
257	CON	0E8B
1292	CONFC	0E56
177	CINDALX	0004
155	CINCCIN	0010
153	CINSENT	000E
162	CINFF2	0016
156	CINFUN	0011
1297	CINREDC	0E63
1323	CINRSG	0E93
154	CINSGAL	000F
1303	CINSGIG	0E6D
913	CONTROL	05F2
1462	CONVHEX	097B
1379	DEBMECD	0E8F
1403	DEBNECD1	0919
1405	DEBNECD2	091D
1330	DISR254	0E99
852	DIRP	057B
62	DISPLATE	0C3E
869	DISPLATEG	0596
1423	DIHGA	093A
159	DIR	0013
1483	DIRV	0992
1491	DIRV1	09A1
1501	DIRV2	09B2
181	DIRV255	0074
182	DIRVINI	00FF
1516	DIWNCBOM	09CC
1438	ELIBRE	0956
1432	DMWD	094B
173	ENTER	0004
1135	ERRCLAS	076E
163	ERRCOR	0017
164	ERRORS	001B
1473	ESP	09BA
180	FACTOR	0064
147	FCOINT1	000B
148	FCOINT2	000C
170	FILORAT	000F
169	FUZZH	0065

TESIS CON  
FALLA DE COMIEN

Control de temperatura con lógica difusa

106	FUZIN	1200
108	FUZOUT	1220
1050	FUZY	0620
1062	FUZYB	06F7
1078	FUZYCTA	0714
1083	FUZYFOS	071C
1069	FUZYVAL	0702
110	FUZVAL	1240
138	GMBL	0006
160	HGA	0014
1465	HMBRNLO	0981
51	INDACTIVO	000E
49	INDACTIVO	08FE
231	INDACTO	004F
192	INDGCK	1100
227	LA	0042
76	LAUTO	003E
188	LCD	0000
82	LOCOMO	00FE
98	LOCOMO	007E
78	LMBIRA	000E
823	LMBIR	0532
1472	LELIS	0989
812	LETRZIN	051A
880	LETRIC	05N6
175	LIMPZA	0001
176	LINDOS	00C0
1031	LOCP	06C0
80	LOCALD	080E
45	MMAL	080E
47	MMALIND	080E
172	MAS	0002
1366	MERLO	08D9
1124	MERE	0764
171	MEROS	0001
38	MICO	080E
1537	MILL	0988
1542	MELANOSUM	098F
1545	MELRESUL	098F
1523	MELT	09CF
855	MEMERLTC	0570
244	MERUTO	006F
248	MERREJ	0078
239	MERPCO	0063
125	MENCIN	1403
124	MEMENT	1402
123	MENRAL	1401
871	OTRA	059A
168	PA	0003
186	PE254	0800
120	PNOTO	008F
132	PNOTOCEN	010A
339	PNOTOCEN	011A
324	PNOTOSBL	00F6
335	PNOTOSIG	0110
1019	PNOTOSUM	0690
1040	PNOTOSUM	069E
366	PNW01	0153
389	PNW01	0187
396	PNW02	0193
413	PNW03	0189
371	PNW0401G	015C
794	PONCUN	04F7
907	PONTEX	058A
577	PPALJE	0300
785	PPCAND	048E
506	PPCHCA	026B
695	PPCOP	0411
623	PPCOPM	0371
711	PPH01	0438
727	PPH02	045C

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

767	PEPND	04B9
758	PEPNS	04A9
742	PEPNE	04B3
769	PEPNE.L	04BF
535	PEPNS	02B6
537	PEPNSC	02BC
775	PENE	04CB
553	PEPND	02B0
518	PEPNDERO	02BC
543	PEPNDOS	02CB
548	PEPNDRES	02DA
524	PEPNDUNO	02B8
595	PEPND2	0337
616	PEPND3	0363
480	PEPND	0238
491	PEPNDSTG	024B
642	PEPND1	03A1
657	PEPND2	03C1
691	PEPND3	040B
681	PEPND4	03F6
788	PEPND	04BF
531	PEPNDEN	02AC
558	PEPNDIP	02B9
563	PEPNDIN	02F4
559	PEPNDOT	02B2
66	PEPND	0C7E
55	PEPND1	0C2E
58	PEPND2	0C3E
817	PEPND3	0C2A
53	PEPND4	0C1E
68	PEPND5	0C8E
305	PEPND6	0CDA
64	PEPND7	0C5E
1167	PEPND	079C
128	PEPND	14A6
30	PEPND	08A9
129	PEPND	15A6
179	PEPND	0010
1171	PEPND	07A1
1149	PEPND	0782
1158	PEPND	07B8
1150	PEPND	0784
15	PEPND	0B14
1281	PEPND	0B45
1231	PEPND	0B03
1190	PEPND	07BF
1187	PEPND	07B8
1224	PEPND	0B0A
1206	PEPND	07BE
1243	PEPND	0B1A
1319	PEPND	0B8D
1176	PEPND	07FA
1211	PEPND	07B6
1248	PEPND	0B23
824	PEPND	0535
1310	PEPND	0B7D
174	PEPND	0008
835	PEPND	0550
213	PEPND	0027
839	PEPND	0559
859	PEPND	0586
843	PEPND	0562
863	PEPND	058F
847	PEPND	056B
851	PEPND	0575
1556	PEPND	0A03
222	PEPND	003B
375	PEPND	0164
127	PEPND	14B5
25	PEPND	0B3F
347	PEPND	012F

TESIS CON  
FALLA DE ORIGEN

Control de temperatura con lógica difusa

252	TECO	007F
454	TECOADAA	0205
429	TECOBRO	01DB
455	TECOBCTN	0208
187	TECLA	0040
433	TECLAD	0181
419	TECLADON	018F
435	TECLAIN	0185
150	TECLAT	000D
902	TECLET	058E
462	TECOMS	0215
471	TECOMO	022A
122	TIBRES	1400
1554	TIBRESOL	0A00
893	TIN91	05C9
900	TIN91	05DF
161	TOPE	0015
1114	VALO	0795
1106	VAL2	0740
178	VALAP	009A
1129	VALFIN	0768
1126	VALFINO	0767
1093	VALJIA	0724
342	VAPROG	0122
132	VARELI	0001
131	VARELL	0000
134	VAREH	0003
133	VAREZL	0002
136	VAREH	0005
135	VARELL	0004
140	VARETI	0007
141	VARETI	0008
183	WORECONI	003E
184	WORECONO	00BA
185	WORECONO	007A

No. of lines : 1561  
No. of errors : 0

TESIS CON  
FALLA DE ORIGEN

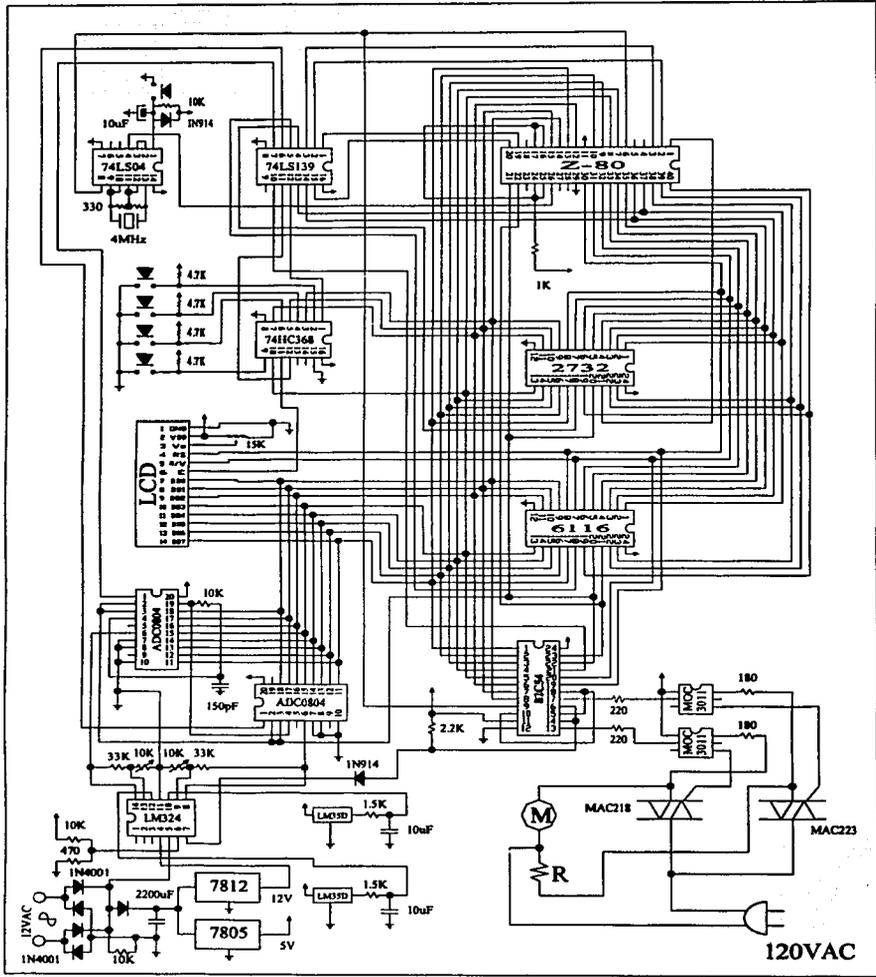


Diagrama completo del sistema

TESIS CON  
FALLA DE COMPLETACION

## BIBLIOGRAFÍA

- a) G. PITA, Edward. Acondicionamiento de aire. Principios y sistemas. Ed. CECSA; 2ª edición; México 1994.
- b) HERNÁNDEZ GORÍBAR, Eduardo. Fundamentos de aire acondicionado y refrigeración. Ed. Limusa ; 1ª edición; México 1973.
- c) W. COTTEL, L. Aire acondicionado y refrigeración para regiones tropicales. Ed. Limusa ; 1ª edición; México 2000.
- d) OGATA, Katsuhiko. Ingeniería de control moderna. Ed. Prentice Hall; 2a edición; New Jersey 1990.
- e) M. PASSINO, Kevin. Fuzzy Control. Ed. Addison-Wesley; 1ª edición; U.S.A 1998.
- f) KOSKO, Bart. Fuzzy Engineering. Ed. Prentice Hall; 1ª edición; New Jersey 1998.
- g) CIARCIA, Steve. Construya una microcomputadora basado en el Z80. Ed. McGraw Hill/ Interamericana de México, S.A. de C.V. ; 1ª edición; México 1998.
- h) W. COFFRON, James. Z80 Aplications. Ed. SYBEX; U.S.A. 1983.
- i) FISHER, Marvin J. Power Electronics. Ed. PWS-KENT PUBLISHING COMPANY; 1ª edición ; Boston 1991.
- j) W. Hart, Daniel. Introduction to power electronics. Ed. Prentice Hall; 1ª edición; New Jersey 1997.
- k) TEXAS INSTRUMENTS. The TTL Logic Databook. U.S.A. 1988.
- l) NATIONAL SEMICONDUCTOR. Memory Databook ; U.S.A. 1990

