

# **ESCUELA POLITECNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

**DISEÑO Y CONSTRUCCION DE UN PROTOTIPO CON  
MICROCONTROLADORES DE LA FAMILIA MSC-51, PARA EL  
USO EN UN CONTROLADOR ELECTRÓNICO DE AGUA USANDO  
LÓGICA DIFUSA.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO EN  
ELECTRONICA Y TELECOMUNICACIONES**

**William Rodrigo Fiallos Ortega.**  
william\_fiallos@hotmail.com

**Geovanny Francisco Rivera Vega.**  
georiv7@hotmail.com

**Director: Ing. Pablo Salinas.**  
psalinas58@hotmail.com

**Quito, octubre, 2008.**

## DECLARACION

Nosotros William Rodrigo Fiallos Ortega y Geovanny Francisco Rivera Vega, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o certificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

---

William Fiallos Ortega

---

Geovanny Rivera Vega.

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por William Fiallos Ortega y Geovanny Rivera Vega, bajo mi supervisión.

---

Ing. Pablo Salinas.  
DIRECTOR DE PROYECTO

## AGRADECIMIENTOS

- A mis padres Gustavo e Hilda, a mis hermanos Alex, Danny y a mi sobrino Alex Javier por todo su apoyo y aliento en todos y cada uno de los momentos difíciles de mi vida.
- A la Escuela Politécnica Nacional y especialmente a mi querida Facultad de Eléctrica.
- Al Ing. Pablo Salinas por brindarnos sincero interés y dedicación en la supervisión de este proyecto de titulación.
- A los amigos Swigers que son muchos y que han estado presentes en todos los buenos y malos momentos, dentro y fuera de universidad.
- A todas las personas que de una u otra manera han estado a mi lado aportando en algo a mi vida y han echo posible la realización de esta tesis.

**William Fiallos O.**

### **DEDICATORIA**

- Para mis padres Gustavo e Hilda, mis hermanos Alex, Danny y al gordo por todo su amor y la fuerza que me han dado, gracias por estar siempre conmigo por que son lo más importante para mi.
- A mis queridos abuelitos Papa Viche y a mi mama Fanny que siempre están en mi corazón.
- A mi tío Flavio que nos hace mucha falta a todos y que pese a la distancia lo llevamos en el corazón.

**William Fiallos O.**

## **DEDICATORIA**

Dedicado a mi Madre, a mi Hermano, a mi Abuelita y a mi Padre que ya no se encuentra con nosotros pero siempre me acompaña.

**Geovanny Francisco Rivera Vega**

## **AGRADECIMIENTOS**

- Agradezco profundamente a mi Madre Enriqueta Vega por su apoyo en todas las etapas de mi vida y por el gran esfuerzo que hizo para educarme..
- A mi Padre Gabriel Rivera quien siempre me ayudo aunque no se encuentra con nosotros pero estoy seguro que desde el cielo siempre me estará brindando su ayuda.
- A mi Hermano Diego Rivera que con su cariño siempre me cuida y me enseña a ser una persona perseverante en todo lo que me proponga,
- A mi Abuelita María Vega a quien agradezco por ser mi segunda madre y comprenderme en momentos en los que la vida nos da la espalda.
- A mis tíos y primos en especial a Eduardo Vega, Zoila Herrera, Christian Vega, Janeth Vega, por su apoyo y cariño brindado.
- Y a mis amigos y amigas incondicionales Christian Verdesoto, Renato Martínez, Gabriela Clerque, Jenny Torres, Paola Mayorga, Alejandra Alvear, William Fiallos, Edwin Yépez, Renán Ponce, Danilo Montalvo, Fabricio Prado, Efraín Herrera, Mauro Espinosa, Carlos Quishpe, Julio Trujillo, Juan Pancho, Santiago Maldonado, Santiago Zaldumbide, Jaime Velasco, por estar siempre juntos y brindarme su apoyo en las buenos y malos momentos que nos presenta la vida.
- En fin agradezco a todas las personas que me conocen a las que aprecio sinceramente e incondicionalmente.
- A los Ingenieros de la Escuela Politécnica Nacional, en especial al Ing. Pablo Salinas por el apoyo y amistad brindados durante el desarrollo del proyecto de titulación.
- A todos aquellos que conforman la Facultad de Ingeniería Electrónica y Telecomunicaciones

**Geovanny Francisco Rivera Vega**

Quito, 7 de Marzo 2006.

Señor MSc.  
Miguel Hinojosa  
Coordinador de carrera  
Ingeniería Eléctrica y Telecomunicaciones  
Presente:

De mi consideración:

Nosotros, William Fiallos Ortega y Geovanny Rivera Vega solicitamos a usted y por intermedio a la Subcomisión Académica Permanente de la Carrera de Ingeniería en Electrónica y Telecomunicaciones se designe tribunal para la calificación de mi proyecto de titulación que esta dirigido por el Ingeniero Pablo Salinas y cuyo titulo es "DISEÑO Y CONSTRUCCION DE UN PROTOTIPO CON MICROCONTROLADORES DE LA FAMILIA MSC-51, PARA EL USO EN UN CONTROLADOR ELECTRÓNICO DE AGUA USANDO LÓGICA DIFUSA".

Atentamente,

---

---



William Fiallos Ortega  
098920612

Geovanny Rivera Vega.  
084322622

## **RESUMEN**

El objeto del presente proyecto, es diseñar y construir el prototipo de un controlador de temperatura utilizando lógica difusa, como también de optimizar los recursos naturales como el agua potable y la energía eléctrica, así como también llevar nuevas opciones a lo que son los consumidores que cada vez buscan artefactos más inteligentes, prácticos y eficientes.

En la primera parte de este proyecto, se establecen la parte teórica donde se encuentran conceptos básicos de lo que es la lógica difusa para entender de mejor manera y familiarizarse con la misma para así poder cumplir con los objetivos planteados.

En función de esto, se realiza un planteamiento y estudio de las posibles alternativas con el fin de solucionar el problema del diseño y así seleccionar los dispositivos mas adecuados para el proyecto.

Dentro de este esquema de selección, se procede al diseño de este controlador de temperatura utilizando lógica difusa, este diseño esta basado en este mecanismo de control que tiene dicho sistema difuso.

Las pruebas de operación realizadas al prototipo construido que arrojan resultados, mismos que serán analizados en la parte final de este trabajo lo que permitirá citar conclusiones y recomendaciones para futuros proyectos.

El diseño de este control de temperatura permite tener un ahorro con lo que tiene que ver con el consumo del agua potable en un porcentaje de 37.83 % al igual que el ahorro de lo que es la energía eléctrica que en estos tiempos en el país es muy necesario, por lo que el costo de fabricación del mismo puede ponerse al alcance

CAPITULO 1 .....	15
DESCRIPCIÓN GENERAL Y CARACTERÍSTICAS FUNDAMENTALES DE LA LÓGICA DIFUSA.....	15
1.1 INTRODUCCIÓN.....	15
1.2 ANTECEDENTES DE LA LÓGICA DIFUSA.....	15
1.3 BASES DE LA LÓGICA DIFUSA.....	17
1.3.1 CONJUNTOS DIFUSOS.....	18
1.3.1.1 INTRODUCCIÓN.....	18
1.3.1.2 OPERACIONES SIMPLES CON CONJUNTOS DIFUSOS.....	20
1.3.1.3 OTRAS OPERACIONES CON CONJUNTOS DIFUSOS .....	24
1.3.1.3.a POTENCIA DE ORDEN $m$ .....	25
1.3.1.3.b PRODUCTO CARTESIANO .....	25
1.3.1.3.c DIFERENCIA LIMITADA.....	25
1.3.1.3.d SUMA LIMITADA.....	26
1.3.1.3.e SUMA ALGEBRAICA .....	26
1.3.1.3.f PRODUCTO ALGEBRAICO .....	26
1.3.1.4 OPERADORES DIFUSOS BASICOS .....	28
1.3.1.4.a T-CONORMAS .....	28
1.3.1.4.b T-NORMAS .....	29
1.3.1.5 OPERADORES COMPENSATORIOS.....	34
1.3.2. NÚMEROS DIFUSOS .....	37
1.3.2.1 INTERVALOS DE CONFIANZA PARA NUMEROS DIFUSOS.....	39
1.3.2.2. NÚMEROS DIFUSOS NORMALES Y CONVEXOS CON $\alpha$ CORTES	40
1.3.2.3. NUMEROS DIFUSOS TRIANGULARES .....	43
1.3.2.4. NUMEROS DIFUSOS TRAPEZOIDALES.....	44
1.3.2. SISTEMAS DE INFERENCIA DIFUSOS .....	45
1.3.3.1 IMPLICACIÓN DIFUSA .....	45
1.4 SISTEMAS COMERCIALES QUE UTILIZAN LA TÉCNICA DIFUSA.....	47
1.4.1. ÁREAS DE APLICACIÓN QUE TIENE LA LÓGICA DIFUSA.....	48
1.5 VENTAJAS DEL SISTEMA DE CONTROL DIFUSO. ....	48
CAPITULO 2 .....	51
ANÁLISIS DEL PROYECTO.....	51
2.1 SISTEMAS DE CONTROL NORMALES.....	51
2.1.1 SISTEMA DE CONTROL DE LAZO ABIERTO.....	52
2.1.2 SISTEMA DE CONTROL DE LAZO CERRADO.....	53
2.1.3 SISTEMAS DE CONTROL PROPORCIONALES INTEGRALES DERIVATIVOS (PID). ....	54
2.2 SISTEMAS DE CONTROL DIFUSOS.....	55
2.2.1 SECCIÓN DE FUSIFICACIÓN .....	56
2.2.2 SECCIÓN DE REGLAS DIFUSAS .....	56
2.2.3 SECCIÓN DE DEFUSIFICACIÓN.....	58
2.2.2.1 MÉTODO DEL CENTRO DE GRAVEAD (COG) O DEL CENTRO DEL ÁREA (COA). ....	59
2.2.2.2 MÉTODO DE EL CENTRO MÁXIMO (COM).....	59

2.2.3 METODO DE DISEÑO .....	60
2.2.4 FORMANDO EL CONJUNTO SOLUCIÓN DE SALIDA .....	68
2.3 IDENTIFICACIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA PARA EL PROTOTIPO. ....	69
2.3.1. TEMPERATURA.....	70
2.3.2 FLUJO DEL AGUA.....	73
CAPITULO 3 .....	73
DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DE CONTROL DE TEMPERATURA.	74
3.1 DISEÑO DEL PROTOTIPO.....	74
3.2 PARÁMETROS DE DISEÑO .....	74
3.2.1 CONTROL DE LA TEMPERATURA.....	74
3.2.2 SELECCIÓN DE PARTES Y PIEZAS. ....	76
3.2.2.1 El ATMEGA8 .....	76
3.2.2.2 TIPO DE ENSAMBLADOR .....	78
3.2.2.3 TRIAC .....	78
3.2.2.4 DISPLAY (LCD).....	80
3.2.2.5 El DS-1820 .....	81
3.2.2.6 EL OPTOTRIAC. (MOC 3020).....	83
3.2.2.7 EL TANQUE.....	84
3.2.2.8 EL ELEMENTO RESISTIVO.....	84
3.2.2.9 FUENTE REGULADA.....	84
3.3 IDENTIFICACIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA DEL CONTROLADOR ELECTRONICO DE TEMPERATURA DIFUSO.....	85
3.3.1 CALCULO DEL GRADO DE PERTENECÍA DE LAS VARIABLES .....	87
3.3.2 CREACION DE LAS REGLAS DIFUSAS.....	92
3.4 DIAGRAMAS DE FLUJO.....	95
3.4.1 FLUJOGRAMA GENERAL.....	96
3.4.2 PROGRAMA PRINCIPAL.....	97
3.4.3 SUBROUTINA READKEY .....	98
3.4.4 SUBROUTINA MSGON.....	99
3.4.5 SUBROUTINA MSGOFF .....	100
3.4.6 SUBROUTINA MSGSETPOINT.....	100
3.4.7 SUBROUTINA TEMPERATURE.....	101
3.4.8 SUBROUTINA EU .....	102
3.4.9 SUBROUTINA RU.....	102
3.4.10 SUBROUTINA INFERENCE .....	103
3.4.11 SUBROUTINA U_TEST.....	104
3.4.12 SUBROUTINA DEFUZZY .....	105
3.4.13 SUBROUTINA PWM_OUT.....	106
3.5 ESQUEMATICO DEL PROTOTIPO.....	108
CAPITULO 4 .....	113
PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS. ....	120
4.2 PRUEBAS REALIZADAS. ....	120
4.2.1 PRUEBAS Y ANALISIS CON RESPECTO AL DESPERDICIO DE AGUA .....	120
4.2.1.1 CALCULO DEL AHORRO DE AGUA.....	123
4.2.2 PRUEBAS CON RESPECTO AL USO DE ENERGÍA.....	124
4.2.2.1 T=19.81 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.2 T= 20.18 grados C .....	<b>¡Error! Marcador no definido.</b>

4.2.2.3	T= 21.31 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.4	T=22.43 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.5	T=23.75 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.6	T= 24.43 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.7	T=24.93 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.8	T= 25.68 grados C .....	<b>¡Error! Marcador no definido.</b>
4.2.2.9	T=26.45 grados C .....	<b>¡Error! Marcador no definido.</b>
4.3	FACTORES QUE AFECTARIAN LA TEMPERATURA DE SALIDA DEL AGUA: .....	126
4.3.1	LA DISTANCIA ENTRE EL CALENTADOR Y EL PUNTO DE USO DE AGUA CALIENTE: .....	126
4.3.2	EL CAUDAL A USAR. ....	126
4.4	CONSTRUCCIÓN Y COSTOS.....	129
4.4.1	COSTOS DE ELEMENTOS NORMALIZADOS UTILIZADOS.....	129
CAPITULO 5	.....	130
CONCLUSIONES Y RECOMENDACIONES	.....	130

## INDICE DE ANEXOS

<b>ANEXO I: HOJA DE DATOS DEL MICROCONTROADOR ATEMEGA8...</b>	<b>120</b>
<b>ANEXO II: HOJA DE DATOS DEL OPTOTRIAC MOC 3020.....</b>	<b>134</b>
<b>ANEXO III: HOJA DE DATOS DEL LCD.....</b>	<b>141</b>
<b>ANEXO IV: HOJA DE DATOS DEL DS1820.....</b>	<b>143</b>
<b>ANEXO V: HOJA DE DATOS DEL TRIAC BT136.....</b>	<b>165</b>
<b>ANEXO VI: HOJA DE DATOS DE LAS DIMENCIONES DEL TANQUE...</b>	<b>172</b>

## **CAPITULO 1**

# **DESCRIPCIÓN GENERAL Y CARACTERÍSTICAS FUNDAMENTALES DE LA LÓGICA DIFUSA.**

### **1.1 INTRODUCCIÓN**

Debido a las necesidades actuales por parte de los consumidores, que cada día demandan bienes y servicios más inteligentes, más versátiles, prácticos y eficientes, que a su vez sean capaces de tomar decisiones rutinarias por nosotros se prevé que las empresas que deseen ser competitivas, deben empezar a adoptar técnicas modernas de este tipo en el desarrollo de sus reductos, aquellas que no lo hagan, estarán destinadas a perder mercado.

Los países asiáticos como Japón están más cautivados por el desarrollo de aplicaciones comerciales aplicando la lógica difusa que es una de las tecnologías de más rápido crecimiento en la actualidad, mientras que en EE.UU. y otras partes del mundo, la atención se centra en la búsqueda del éxito comercial.

### **1.2 ANTECEDENTES DE LA LÓGICA DIFUSA.**

Cuando se oye hablar por primera vez de la “lógica difusa” (*fuzzy logic*) o, incluso, para quiénes han oído hablar algo sobre este tema, pero carecen de conocimiento en la materia, se produce una inmediata tentación de hacer una pequeña gracia a costa de su propio nombre: debe ser algo “poco claro”, quizás “nada elaborado”; muy “borroso”, en definitiva. Estos son el tipo de comentarios

que suelen hacerse. Pero, por lo contrario, hay un soporte matemático poderoso detrás de la misma.

Sin embargo, la aparente contradicción interna de su propio nombre no ha sido un inconveniente para un amplísimo desarrollo teórico y una, más reciente, aplicación de la lógica difusa al campo de la ciencia, la tecnología e, incluso, al análisis empírico en las ciencias sociales.

Se ha señalado que los antecedentes filosóficos pueden encontrarse en los trabajos de Bertrand Russell, Jan Lukasiewicz y Max Black, que ante las paradojas encontradas en la lógica tradicional, comenzaron a hablar de lógica multivalente y lógica continua, introduciendo, así, en la naturaleza blanco/negro (verdad/falso) de aquella, las sombras de gris inherentes al mundo real. La idea de que cualquier afirmación lógica debe ser cierta o falsa, y ninguna otra cosa podía producir paradojas como la planteada por Russell, y a pesar del intento de obviarlas introduciendo los “axiomas de la teoría de conjuntos” para generar fórmulas “bien definidas”, choca con la evidencia de que en la vida cotidiana son admisibles *grados* de verdad. Paradojas como la mencionada sólo podían ser resueltas en el contexto de la lógica continua. Posteriormente, Zadeh (1965) desarrolló la teoría de conjuntos multivaluados, ampliándola con el concepto de *cuantificador*, e introdujo el término *fuzzy* para denominar este tipo de lógica.

Por ejemplo, supongamos que hacemos la siguiente sentencia: “El Sahara es un desierto”. Evidentemente, la totalidad de la gente estaría de acuerdo en decir que eso es cierto. Supongamos que quitamos un grano de arena. Si volvemos a preguntar, se continuaría diciendo que es cierto. Si continuamos así, quitando grano a grano, y continuamos preguntando, llegaría un momento en que no habría más arena y el Sahara ya no sería un desierto. Si preguntamos ahora, la respuesta debiera ser que es falso.

Pero, ¿en qué momento se produce, exactamente, el cambio del “sí” al “no”?



Evidentemente, un desierto no deja de serlo cuando se le quita un particular grano de arena. Es más real afirmar que conforme el Sahara pierde arena, la afirmación de que es un desierto va haciéndose menos verdad y que al quitar el último grano, la verdad desaparece por completo. La lógica continua reconoce el hecho de que la realidad no tiene por qué ser sólo cierta o falsa, sino que puede haber *grados de verdad*. Dicho de otra forma, los elementos de un conjunto no tienen por qué estar completamente dentro o completamente fuera del mismo; es posible la pertenencia parcial, un *grado de pertenencia*, de un elemento a un conjunto dado.

Como se verá más adelante, la lógica tradicional queda incorporada como un caso particular de la lógica continua o difusa. En este sentido, podría decirse que la lógica difusa es una extensión de un sistema preciso de representación para incluir la vaguedad y la incertidumbre. La vaguedad ya no es el límite de la precisión sino, al contrario, ésta es el límite de aquélla.

Su aplicación se ha visto necesaria, en varios campos (fundamentalmente relacionados con la ingeniería y la informática) entre los cuales pueden contemplarse los sistemas en que la vaguedad es bastante común, como podría ser el caso de la economía

### **1.3 BASES DE LA LÓGICA DIFUSA**

Entre los sistemas difusos se incluyen diversas teorías; Teoría de Conjuntos Borrosos, como extensión de la teoría de conjuntos clásica, Lógica Borrosa, que puede ser considerada una ampliación de las lógicas n-valuadas propuestas por Lukasiewicz en 1930, que son a su vez extensión de la lógica trivaluada (verdadero, falso e indeterminado). También se habla de sistemas de control borrosos, que utilizan las expresiones de la lógica borrosa para formular reglas orientadas al control de sistemas.

Estos sistemas de control borroso pueden considerarse una extensión de los sistemas expertos, pero superando los problemas prácticos que estos sistemas tienen para el razonamiento en tiempo real, causados por la explosión exponencial de las necesidades de cálculo para el análisis lógico completo de amplias bases de reglas.

### 1.3.1 CONJUNTOS DIFUSOS.

#### 1.3.1. 1 INTRODUCCIÓN

Los conjuntos difusos pueden ser considerados como una generalización de los conjuntos clásicos; la teoría clásica de conjuntos sólo contempla la pertenencia o no pertenencia de un elemento a un conjunto, sin embargo la teoría de conjuntos difusos, contempla la pertenencia parcial de un elemento a un conjunto, es decir, cada elemento presenta un grado de pertenencia a un conjunto difuso que puede tomar cualquier valor entre 0 y 1. Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso: para cada valor que pueda tomar un elemento o variable de entrada  $x$  la

Función característica  $m_A(x)$  proporciona el grado de pertenencia de este valor de  $x$  al conjunto difuso  $A$ . Formalmente, un conjunto clásico  $A$ , en un universo de discurso  $U$ , se puede definir de varias formas:

- Enumerando los elementos que pertenecen al conjunto.
- Especificando las propiedades que deben cumplir cada uno de los elementos que pertenecen a ese conjunto o, en términos de la función de pertenencia  $m_A(x)$ :

$$m_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}$$

Podemos además decir que el conjunto  $A$  es matemáticamente equivalente a su función de pertenencia o característica  $m_A(x)$ , ya que conocer  $m_A(x)$  es lo mismo que conocer  $A$ .

La teoría de conjuntos difusos parte de la teoría clásica e introduce una función de pertenencia al conjunto, definida como un número real entre 0 y 1.

Se introduce el concepto de valor lingüístico asociado a cada subconjunto borroso, y definido por una palabra o etiqueta lingüística  $A$ . Para cada valor lingüístico  $A$  se define una función  $\mu_A(t)$ , que indica el grado de probabilidad de que la variable  $t$  esté incluida en el concepto representado por el valor lingüístico  $A$ .

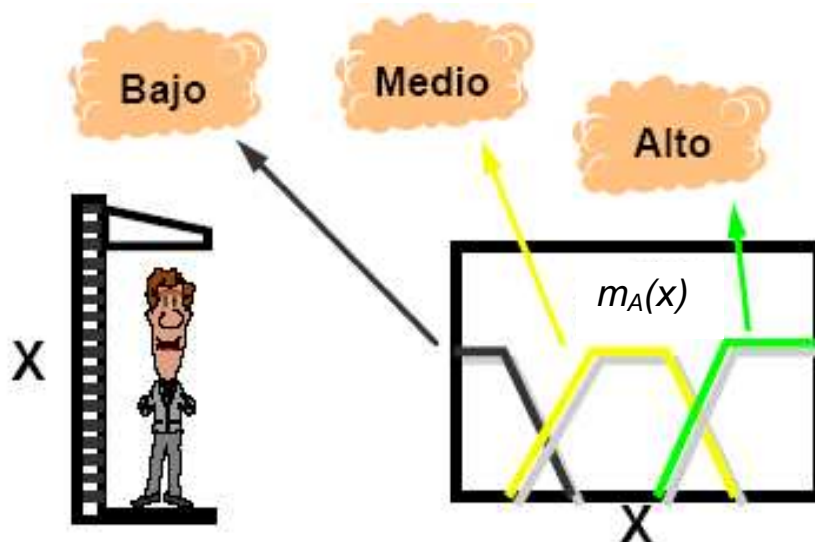


Figura 1.1: CONJUNTOS DIFUSOS Y FUNCIONES DE PERTENENCIA

Para los valores  $x$  que puede tomar una magnitud física como la altura de una persona pueden definirse diversos conjuntos borrosos (Figura 1). Cada conjunto borroso tiene asociado una etiqueta lingüística; como *Medio*, para los valores de estatura de valor entorno al promedio de la población. Además se ha de definir una función de pertenencia  $m_A(x)$  con valores entre 0 y 1.

El valor de esta función indica que también encaja un valor concreto  $x$  en el conjunto difuso  $A$  de las estaturas medias.

De esta forma los conjuntos difusos representan predicados en la lógica proposicional. El objeto de la lógica difusa es proporcionar un soporte formal al razonamiento en el lenguaje natural que se caracteriza por un razonamiento aproximado con proposiciones imprecisas.

### 1.3.1.2 OPERACIONES SIMPLES CON CONJUNTOS DIFUSOS.

Como se ha visto hasta ahora la componente fundamental de un conjunto difuso viene dada por su función de pertenencia. Por esto es que las operaciones con este tipo de conjuntos se definen por medio de dicha función.

Veremos en primer lugar las operaciones básicas paralelas que se pueden realizar con los conjuntos comunes.

- **Subconjunto:** Un conjunto difuso A es un subconjunto difuso de B, esto quiere decir que  $A \subseteq B$ , si :

$$\forall x \in U : m_A(x) \leq m_B(x) \quad (1.1)$$

Y si existe, por lo menos un punto  $x \in U$  tal que  $m_A(x) \neq m_B(x)$ , entonces escribiremos que  $A \subset B$ .

- **Intersección (min-intersección):**

La intersección entre 2 conjuntos difusos vendrá dada de la misma manera que la de dos conjuntos comunes siendo esta:

$$m_{A \cap B}(x) = \min\{m_A(x), m_B(x)\} \quad (1.2)$$

- **Unión (max-intersección):**

La unión entre 2 conjuntos difusos vendrá dada de la misma manera que la de dos conjuntos comunes siendo esta:

$$m_{A \cup B}(x) = \max\{m_A(x), m_B(x)\} \quad (1.3)$$

- **Complementación:**

El complemento de un conjunto difuso,  $A^c$  vendrá dado, punto a punto por:

$$m_{A^c}(x) = 1 - m_A(x) \quad (1.4)$$

### EJEMPLO 1:

Sea  $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  el número de habitaciones en una casa y definamos el conjunto “casa confortable para una familia de 4 miembros”. Tal conjunto difuso puede ser descrito como sigue:

$$A = \{ (1, 0.2), (2, 0.5), (3, 0.8), (4, 1), (5, 0.7), (6, 0.3) \}$$

Como puede observarse todas las casas que tienen entre 1 y 6 habitaciones pertenecen al conjunto, pero no todas lo hacen con el mismo grado. Mientras que las de cuatro habitaciones se contemplan como plenamente satisfactorias, las de una sola habitación tienen un grado de compatibilidad muy bajo, como “casa confortable para una familia de 4 miembros”. Las de más de 6 habitaciones no se consideran confortables. Entre estos dos extremos hay diferentes grados de compatibilidad o pertenencia al conjunto difuso.

A continuación se define otro conjunto difuso con el siguiente título, “tipo de casa grande” :

$$B = \{ (3, 0.2), (4, 0.4), (5, 0.6), (6, 0.8), (7, 1), (8, 1) \}$$

Realizando las operaciones indicadas anteriormente tendremos:

a) Conjunto de “casas grandes y confortables”:

$$A \cap B = \{(3, 0.2), (4, 0.4), (5, 0.6), (6, 0.3)\},$$

b) Conjunto de “casas grandes o confortables”:

$$A \cup B = \{(1, 0.2), (2, 0.5), (3, 0.8), (4, 1), (5, 0.7), (6, 0.8), (7, 1), (8, 1)\},$$

c) Conjunto de “casas no grandes”:

$$B^c = \{(1, 1), (2, 1), (3, 0.8), (4, 0.6), (5, 0.4), (6, 0.2), (9, 1), (10, 1)\}.$$

Puede comprobarse fácilmente que la unión y la intersección anteriormente definidas, son compatibles con el concepto de  $\alpha$ -corte. O sea,

$$\begin{aligned}(A \cap B)_\alpha &= A_\alpha \cap B_\alpha \\ (A \cup B)_\alpha &= A_\alpha \cup B_\alpha\end{aligned}$$

## Ejemplo 2

Considerando los conjuntos difusos  $A =$  “números reales próximos a 10” y  $B =$  “números reales próximos a 15” donde se definen de la siguiente forma:

$$A = \{(x, mA(x)) \mid x \in X\}$$

Donde:

$$\begin{aligned}mA(x) &= 0, \text{ para } x < 5, \\ &= (x-5)/5, \text{ para } 5 \leq x \leq 10, \\ &= (15-x)/5, \text{ para } 10 \leq x \leq 15, \\ &= 0, \text{ para } x > 15.\end{aligned}$$

Y

$$B = \{(x, m_B(x)) \mid x \in X\}$$

Donde:

$$\begin{aligned} m_B(x) &= 0, \text{ para } x < 10, \\ &= (x-10)/5, \text{ para } 10 \leq x \leq 15, \\ &= (20-x)/5, \text{ para } 15 \leq x \leq 20, \\ &= 0, \text{ para } x > 20. \end{aligned}$$

Si realizamos la intersección entre estos dos conjuntos el resultado será un nuevo conjunto llamado C = "Números cercanos a 10 y a 15" el mismo que tendría la siguiente forma:

$$\begin{aligned} m_C(x) &= m_{A \cap B}(x) = \min[m_A(x), m_B(x)] = \min[(15-x)/5, (x-10)/5], \text{ para } 10 \leq x \leq 15, \\ &= 0, \text{ en el resto.} \end{aligned}$$

Realizando la unión de estos dos conjuntos se obtendrá el conjunto D = "números reales próximos a 10 o a 15", el mismo que vendrá dado por la siguiente expresión:

$$\begin{aligned} m_D(x) &= m_{A \cup B}(x) = \max[(x-5)/5, (15-x)/5, (x-10)/5, (20-x)/5], \text{ para } 5 \leq x \leq 20, \\ &= 0, \text{ en el resto.} \end{aligned}$$

Definiendo ambas operaciones gráficamente. La intersección será el conjunto difuso señalado con línea gruesa en la Figura 2, mientras que la unión será el conjunto reflejado en la Figura 3.

□

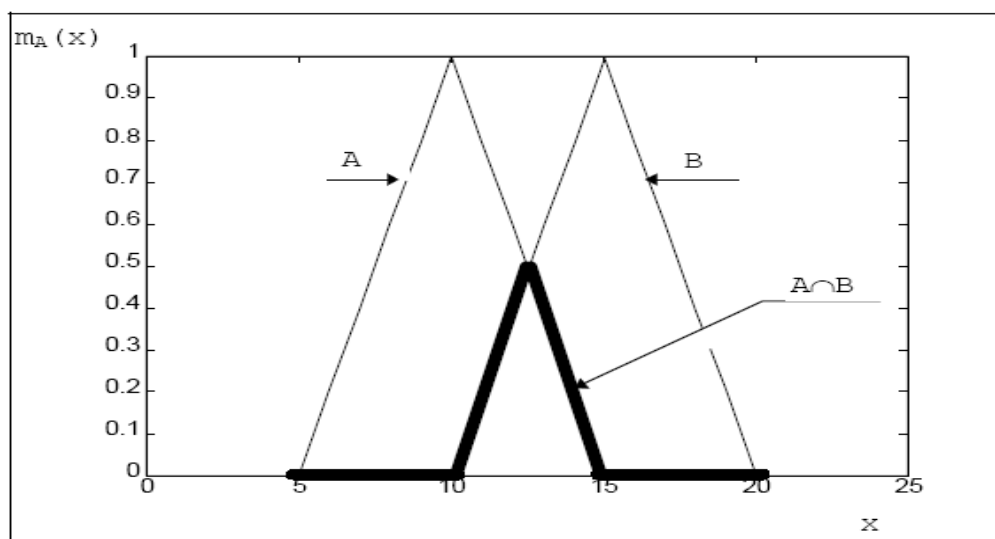


Figura 1.2 INTERSECCIÓN ENTRE A Y B

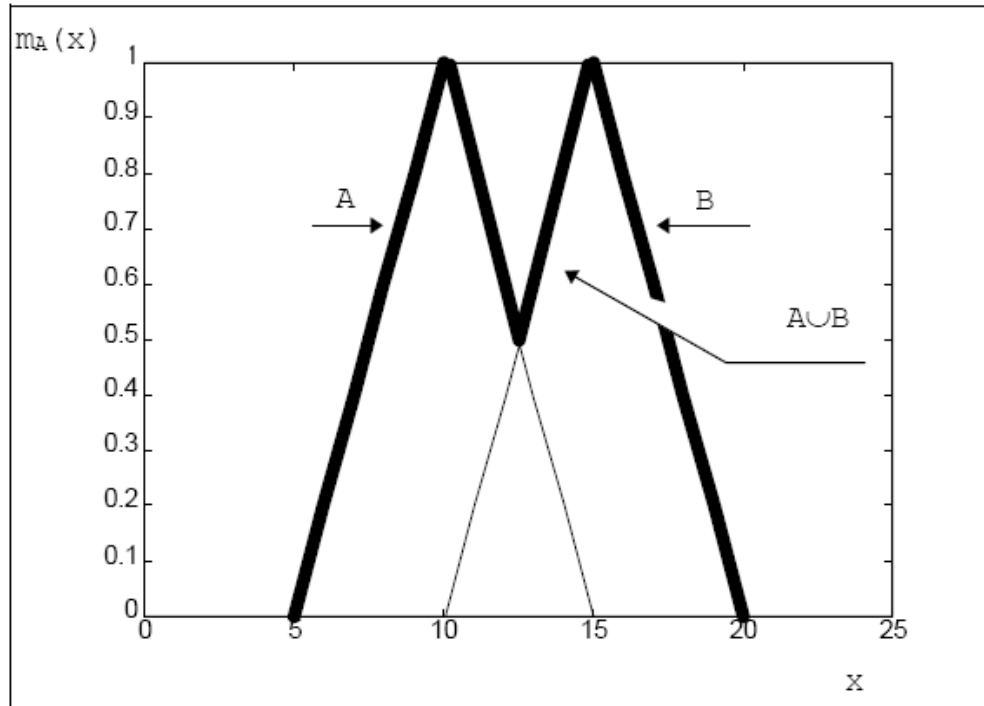


Figura 1.3 UNIÓN ENTRE A Y B.

Se sabe que estas operaciones constituyen un álgebra de Boole en el caso de conjuntos comunes. Mientras que en el caso de los conjuntos difusos esto no es así, ya que no cumplen la ley del medio excluido, es decir,

$$A \cap A^c \neq 0 \text{ y } A \cup A^c \neq U$$

Por estas las operaciones que se acabaron de ver sólo tienen las propiedades de un álgebra distributiva.

### 1.3.1.3 OTRAS OPERACIONES CON CONJUNTOS DIFUSOS



Las operaciones indicadas anteriormente como son la intersección, la unión y el complemento con conjuntos difusos no son las únicas posibilidades para una extensión de las operaciones clásicas realizadas con conjuntos.

Se han propuesto una gran variedad de operadores, ordenados ya sea por su generalidad o por su adaptabilidad, justificándolos ya sea por razones empíricas o axiomáticas.

A continuación se indicara algunas de estas operaciones las mismas que nos pueden ser muy útiles al tratar con conjuntos difusos.

#### **1.3.1.3.a POTENCIA DE ORDEN $m$**

La potencia de orden  $m$  de un conjunto difuso  $A$  será un conjunto difuso cuya función de pertenencia es:

$$m_{A^m}(x) = [m_A(x)]^m, \quad x \in X \quad (1.5)$$

#### **1.3.1.3.b PRODUCTO CARTESIANO**

Sean  $A_1, A_2, \dots, A_n$  conjuntos difusos en  $X_1, X_2, \dots, X_n$ . El producto cartesiano es un conjunto difuso en el espacio producto  $X_1 \times X_2 \times \dots \times X_n$ , cuya función de pertenencia es:

$$m_{A_1 \times A_2 \times \dots \times A_n}(x) = \min_i \{m_{A_i}(x_i)\} \quad x = (x_1, x_2, \dots, x_n), x_i \in X_i \quad (1.6)$$

#### **1.3.1.3.c DIFERENCIA LIMITADA**

La diferencia limitada,  $C = A \ominus B$ , se define como:

$$C = \{(x, m_{A \oplus B}(x)) | x \in X\}$$

Donde:

$$m_{A \oplus B}(x) = \max\{0, m_A(x) + m_B(x) - 1\} \quad (1.7)$$

#### ***1.3.1.3.d SUMA LIMITADA***

La suma limitada,  $C = A \oplus B$ , se define como:

$$C = \{(x, m_{A \oplus B}(x)) | x \in X\}$$

Donde:

$$m_{A \oplus B}(x) = \min\{1, m_A(x) + m_B(x)\} \quad (1.8)$$

#### ***1.3.1.3.e SUMA ALGEBRAICA***

La suma algebraica o suma probabilística,  $C = A + B$ , se define como:

$$C = \{(x, m_{A+B}(x)) | x \in X\}$$

Donde :

$$m_{A+B}(x) = m_A(x) + m_B(x) - m_A(x) \cdot m_B(x) \quad (1.9)$$

#### ***1.3.1.3.f PRODUCTO ALGEBRAICO***

El producto algebraico,  $C = A \cdot B$ , se define como:

$$C = \{(x, m_{A \cdot B}(x)) | x \in X\}$$

Donde:

$$m_{A \cdot B}(x) = m_A(x) \cdot m_B(x) \quad (1.10)$$

### Ejemplo 3

Suponiendo que  $A = \{(3, 0.5), (5, 1), (7, 0.6)\}$  y  $B = \{(3, 1), (5, 0.6)\}$ .

Calcularemos el producto algebraico entre estos dos conjuntos difusos.

$$A \times B = \{(3;3), 0.5\}, \{(5;3), 1\}, \{(7;3), 0.6\}, \{(3;5), 0.5\}, \{(5;5), 0.6\}, \{(7;5), 0.6\}$$

Se representara el resultado en forma de una matriz de doble entrada, la misma que define una relación difusa entre A y B, en la que las casillas recogen el valor mínimo de las funciones de pertenencia del correspondiente par.

<b>3</b>	<b>5</b>	<b>7</b>	
0.5	0.1	0.6	<b>3</b>
0.5	0.5	0.6	<b>5</b>

Los operadores que se acabo de ver entre ellos, por ejemplo la min-intersección, están dentro o pertenecen a una clase generalizada de operadores llamados **normas triangulares** (t-normas), mientras que otros como la suma

limitada pertenecen a otro grupo generalizado con el nombre de **conormas triangulares** (t-conormas).

Operadores utilizados en la intersección y la unión de conjuntos difusos respectivamente.

El concepto de norma triangular es utilizado en la estadística para datos los mismos que deben pertenecer al intervalo  $[0,1]$ . Es por esto que en lo que corresponde a la matemática difusa nuestro interés será enfocado especialmente en el campo de la lógica difusa y de los sistemas expertos.

Existe otra clase de operadores, denominados como **operadores promedio o compensadores**, los mismos que al agregar conjuntos difusos como por ejemplo en el contexto de la toma de decisiones son de mucha importancia.

Estos operadores utilizan un tipo de criterios que frecuentemente se usan en la teoría de la utilidad o en los métodos de ayuda en la decisión multicriterio.

Tratan de representar algún tipo de situación que se encuentre en el intermedio de un criterio mínimo (intersección) y un criterio máximo (unión) de las funciones de pertenencia.

Los operadores también pueden clasificarse como paramétricos y no paramétricos en cualquiera de los 2 casos.

A continuación se revisara los operadores más importantes:

#### **1.3.1.4 OPERADORES DIFUSOS BASICOS**

##### ***1.3.1.4.a T-CONORMAS***

Están definidas como funciones bivalentes de la siguiente forma:

$$[0,1] \times [0,1] \longrightarrow [0,1]$$

Y las propiedades que cumplen son las siguientes:

1.  $s(1, 1) = 1$
2.  $s(m_A(x), 0) = s(0, m_A(x)) = m_A(x), x \in X$
3. Es una función monótona, o sea, que si  $m_A(x) \leq m_C(x)$  y  $m_B(x) \leq m_D(x)$ , entonces:

$$s(m_A(x), m_B(x)) \leq s(m_C(x), m_D(x))$$

4. Cumple con la propiedad conmutativa, o sea:

$$s(m_A(x), m_B(x)) = s(m_B(x), m_A(x))$$

5. Cumple con la propiedad asociativa, o sea:

$$s(m_A(x), s(m_B(x), m_C(x))) = s(s(m_A(x), m_B(x)), m_C(x))$$

#### **1.3.1.4.b T-NORMAS**

Se han definido en correspondencia con los operadores vistos anteriormente, una clase general de operadores para la agregación, que son las ya mencionadas normas triangulares, también denominadas como t-normas.

En si dichas t-normas tienen mucha relación con las ya vistas t-normas ya que se trata de funciones muy similares a las anteriores es decir funciones bivalentes, de la siguiente manera:

$$[0,1] \times [0,1] \rightarrow [0,1]$$

A demás deben satisfacer las siguientes propiedades:

1.  $t(0, 0) = 0$
2.  $t(m_A(x), 1) = t(1, m_A(x)) = m_A(x), x \in X$
3. Es una función monótona, o sea, que si  $m_A(x) \leq m_C(x)$  y  $m_B(x) \leq m_D(x)$ , entonces:

$$t(m_A(x), m_B(x)) \leq t(m_C(x), m_D(x))$$

4. Cumple con la propiedad conmutativa; o sea:

$$t(m_A(x), m_B(x)) = t(m_B(x), m_A(x))$$

5. Cumple con la propiedad asociativa; o sea:

$$t(m_A(x), t(m_B(x), m_C(x))) = t(t(m_A(x), m_B(x)), m_C(x))$$

La función T define una clase general de operadores intersección para los conjuntos difusos.

Es posible calcular valores de la función de pertenencia para la intersección de más de dos conjuntos difusos, debido a que los operadores que pertenecen a esta clase permiten utilizar la propiedad asociativa.

Tanto las T-normas como la t-conorma representan conceptos duales. Ya que cualquier T-norma puede ser generada a partir de una T-conorma, utilizando la siguiente relación:

$$t(m_A(x), m_B(x)) = 1 - s(1-m_A(x), 1-m_B(x))$$

Existen muchos tipos de operadores funcionales que por sus propiedades pueden satisfacer las condiciones para ser ya sea una T-norma o una T-conorma.

Por lo tanto no existe una forma única de combinación en cuanto a números difusos (intersección “y” y unión “o” tradicionales, si no que existen múltiples formas para realizarlo.

Existen también otros procedimientos par ponerlas en relación (operadores, compensadores, algunos de los cuales serán revisados más adelante), los mismos que nos serán especialmente útiles en diversas aplicaciones, como el análisis de decisión.

En cuanto a las parejas más usuales de T-normas y T-conormas son las siguientes:

• **PRODUCTO DRÁSTICO:**

$$t_W(m_A(x), m_B(x)) = \min\{m_A(x), m_B(x)\} \text{ si } \max\{m_A(x), m_B(x)\} = 1$$

$$= 0, \text{ en otro caso.}$$

• **SUMA DRÁSTICA:**

$$s_W(m_A(x), m_B(x)) = \max\{m_A(x), m_B(x)\} \text{ si } \min\{m_A(x), m_B(x)\} = 0$$

$$= 1, \text{ en otro caso.}$$

- **DIFERENCIA LIMITADA:**

$$t_1(m_A(x), m_B(x)) = \max\{0, m_A(x) + m_B(x) - 1\}$$

- **SUMA LIMITADA:**

$$s_1(m_A(x), m_B(x)) = \min\{1, m_A(x) + m_B(x)\}$$

- **PRODUCTO DE EINSTEIN:**

$$t_{1.5}(m_A(x), m_B(x)) = \frac{(m_A(x) \cdot m_B(x))}{(1 + m_A(x) \cdot m_B(x))}$$

- **SUMA DE EINSTEIN:**

$$s_{1.5}(m_A(x), m_B(x)) = \frac{(m_A(x) + m_B(x))}{(1 + m_A(x) \cdot m_B(x))}$$

- **PRODUCTO ALGEBRAICO:**

$$t_2(m_A(x), m_B(x)) = m_A(x) \cdot m_B(x)$$



- **SUMA ALGEBRAICA:**

$$s_2(m_A(x), m_B(x)) = m_A(x) + m_B(x) - m_A(x) \cdot m_B(x)$$

- **PRODUCTO DE HAMACHER:**

$$t_{2.5}(m_A(x), m_B(x)) = \frac{(m_A(x) \cdot m_B(x))}{(m_A(x) + m_B(x) - m_A(x) \cdot m_B(x))}$$

- **SUMA DE HAMACHER:**

$$s_{2.5}(m_A(x), m_B(x)) = \frac{(m_A(x) + m_B(x) - 2m_A(x) \cdot m_B(x))}{1 - m_A(x) \cdot m_B(x)}$$

- **MINIMO**

$$t_3((m_A(x), m_B(x))) = \min\{m_A(x), m_B(x)\}$$

- **MÁXIMO**

$$s_3(m_A(x), m_B(x)) = \max\{m_A(x), m_B(x)\}$$

Los operadores que hemos visto anteriormente están ordenados de la siguiente forma:

$$t_w \leq t_1 \leq t_{1.5} \leq t_2 \leq t_{2.5} \leq t_3$$

$$S_3 \leq S_{2.5} \leq S_2 \leq S_{1.5} \leq S_1 \leq S_w$$

Lo que se quiere resaltar es que para cada par de conjuntos difusos, esto quiere decir que cualquier operador intersección, o una t-norma, estará limitado por el operador min y el operador  $t_w$ .

Lo mismo se tendrá con una t-conorma ya que estará limitada por el operador max y el operador  $s_w$ .

Diferentes familias parametrizadas de t-normas y t-conormas han sido propuestas por diversos autores para extender el rango de estos operadores de tal forma que puedan adaptarse al contexto en el que serán utilizados.

Por ejemplo Hamacher propone un modelo para la intersección en forma parametrizada como se puede ver a continuación:

$$m_{A \cap B} = \frac{(m_A(x) \cdot m_B(x))}{(\gamma + (1 - \gamma)(m_A(x) + m_B(x) - m_A(x) \cdot m_B(x)))}$$

Por lo tanto tendremos que:

Si  $\gamma = 0$  será el producto de Hamacher y,

Si  $\gamma = 1$  será el producto algebraico.

### 1.3.1.5 OPERADORES COMPENSATORIOS

Como se dijo antes en muchas ocasiones los procedimientos de agregación se enfrentaran o necesitaran encontrar un compromiso entre los objetivos en conflicto como se indica por ejemplo en la teoría del bienestar y el análisis multicriterio.

Es posible que en ciertos casos se admita la compensación y se permita un resultado entre el peor y el mejor de todos los resultados posibles, tomando en cuenta el mínimo y el máximo grado de pertenencia de los conjuntos agregados.

A continuación algunas de los operadores más importantes que son utilizados con los conjuntos difusos:

➤ **Operador “y”**

$$m_y(m_A(x), m_B(x)) = \gamma \cdot \min\{m_A(x), m_B(x)\} + (1/2)(1 - \gamma)(m_A(x) + m_B(x)), \gamma \in [0, 1]$$

➤ **Operador “o”**

$$m_o(m_A(x), m_B(x)) = \gamma \cdot \max\{m_A(x), m_B(x)\} + (1/2)(1 - \gamma)(m_A(x) + m_B(x)), \gamma \in [0, 1]$$

El factor  $\gamma$  indica el grado de proximidad al significado lógico de “y” y “o”.

Así, entonces:

Para  $\gamma = 1$ , se tendrá el operador min y max, respectivamente, mientras que para  $\gamma = 0$ , ambos son la media aritmética.

A parte de estos operadores se han propuesto otros más generales, como por ejemplo, el  $\gamma$ -operador o “y compensatorio” de Zimmermann-Zisno el mismo que se basa en una combinación entre el producto y la suma algebraicos.

En vista de lo que se ha dicho hasta aquí, se puede afirmar que existen varias propuestas para la agregación de conjuntos difusos, por lo que se contribuye a que la teoría se haga bastante confusa.

Los criterios que han sido dados para la selección por Zimmermann pueden ser de gran ayuda, entre los más importantes están los siguientes:

- **Fuerza Axiomática:** mientras menos limitativos son los axiomas en los que se basa un operador este es mejor.
- **Ajuste Empírico:** No solo es importante que el operador cumpla con ciertos axiomas teóricos, también debe dar buenos resultados en la práctica.
- **Adaptabilidad:** El operador debe ser bueno para modelar diversas situaciones y ser adaptable a cualquier contexto específico. Esta cualidad es alcanzable gracias a la parametrización.
- **Eficiencia Numérica:** Existen operadores complejos que requieren mas potencia de cálculo que otros. Esto es muy importante cuando el problema que se esta tratando es de gran escala.
- **Compensación:** Debe permitirse una cierta compensación entre un bajo grado de pertenencia de un elemento a un conjunto y un elevado grado a otro. Es obvio que el operador min no lo hace posible.
- **Rango de Compensación:** mientras mayor sea el rango de compensación mejor será el operador compensatorio.
- **Buen comportamiento en la agregación:** el grado de pertenencia de un conjunto agregado dependerá frecuentemente del número de conjuntos combinados. Es decir mientras mas conjuntos adicionales el grado de pertenencia del agregado resultante disminuirá. Por razones formales existen opiniones que indican el grado de pertenencia resultante nunca debería incrementarse.
- **Elección en función del nivel de escala permisible:** Por ejemplo trabajando con una escala ordinal nunca se debería utilizar el operador producto. En caso de igualdad en los demás criterios, es preferible usar el

operador más bajo desde el punto de vista de la recolección de información.

### 1.3.2. NÚMEROS DIFUSOS

Un número difuso puede ser definido como un conjunto difuso  $A$  en  $R^1$  si  $A$  es convexo y existe exactamente en un punto, así:

$$M \in R^1, \text{ con } m_A(M) = 1 (A_{\alpha=1} = M)$$

Una expresión con la que se puede definir la expresión anterior sería, “Aproximadamente  $M$ ”. Se suele definir un número difuso tipo L-R (Left-Right) para una mejor manipulación como se indica a continuación:

$$\begin{aligned} m_A(x) &= L\left(\frac{(M-x)}{l}\right), \text{ si } x \leq M; l > 0 \\ &= R\left(\frac{(x-M)}{r}\right), \text{ si } x \geq M; r > 0 \end{aligned}$$

Donde  $R$  y  $L$  son fuertemente decrecientes en  $R^+$ , con  $L(0)=R(0)=1$ .  $M$  se denomina valor central del número difuso.  $L$  y  $R$  son las funciones izquierda y derecha respectivamente, mientras que  $l$  y  $r$  son la extensión, amplitud, o dispersión a izquierda y derecha respectivamente.

Generalizando un intervalo difuso de tipo L-R se vera que se parece mucho a las expresiones que acabo de ver teniendo así:

$$\begin{aligned}
 m_A(x) &= L\left(\frac{(M_1 - x)}{l}\right), \text{ si } x \leq M_1; l > 0 \\
 &= 1 \text{ si } x \in [M_1, M_2] \\
 &= R\left(\frac{(x - M_2)}{r}\right), \text{ si } x \geq M_2; r > 0
 \end{aligned}$$

A continuación se puede observar un ejemplo de lo que se ha dicho de número difuso del tipo L-R con los siguientes datos:

$$\begin{aligned}
 L(x) = R(x) &= \frac{1}{1+x^2}, \text{ con } l = 2, r = 4 \text{ y } m = 5 \text{ y con un intervalo difuso,} \\
 \text{para } L(x) = R(x) &= \frac{1}{1+x^2}, \text{ con } l = 1, r = 3, M_1 = 10 \text{ y } M_2 = 15.
 \end{aligned}$$

Entonces los conjuntos quedan definidos como se ve a continuación:

- Número difuso: 
$$\begin{aligned}
 m(x) &= L\left(\frac{(5-x)}{2}\right) = \frac{1}{\left(1+\left(\frac{(5-x)}{2}\right)^2\right)}, \text{ para } x \leq 5 \\
 &= R\left(\frac{(x-5)}{4}\right) = \frac{1}{\left(1+\left(\frac{(x-5)}{4}\right)^2\right)}, \text{ para } x \geq 5
 \end{aligned}$$

- Intervalo difuso:

$$\begin{aligned}
 m(x) &= L(10-x) = \frac{1}{(1+(10-x)^2)}, \text{ para } x \leq 10 \\
 &= 1, \text{ para } 10 \leq x \leq 15 \\
 &= L\left(\frac{(x-15)}{3}\right) = \frac{1}{\left(1+\left(\frac{(x-15)}{3}\right)^2\right)}, \text{ para } x \geq 15
 \end{aligned}$$

Lo visto se muestra en la figura 4 a continuación:

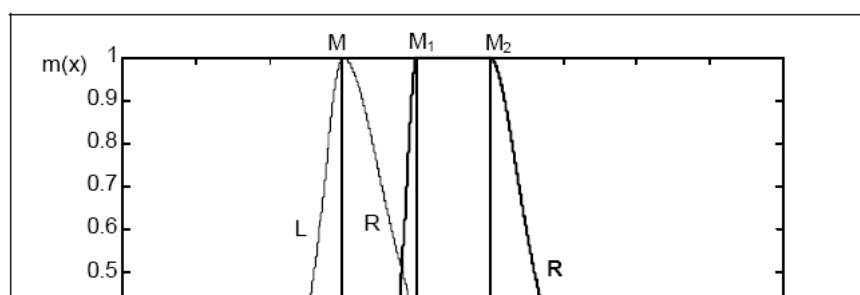


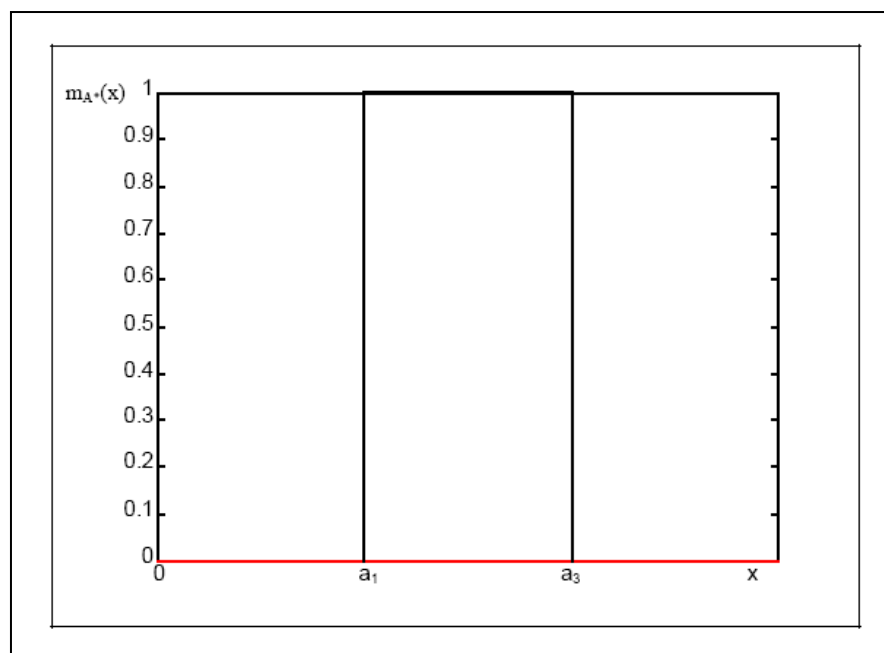
Figura 1.4. NUMERO E INTERVALO DIFUSO DEL TIPO L-R

### 1.3.2.1 INTERVALOS DE CONFIANZA PARA NUMEROS DIFUSOS

Cualquier número que se tenga por ejemplo,  $a$ , y que pertenezca a un campo real se puede interpretar utilizando el concepto de función de pertenencia como se muestra a continuación:

$$m_a(x) = 1, \text{ para } x = a \\ = 0, \text{ para } x \neq a$$

Si en cambio se trata de definir un intervalo de confianza en  $R$  para un número ordinario como se muestra en la figura 5 a continuación.



**Figura 1.5 NÚMERO A DADO POR UN INTERVALO DE CONFIANZA.**

Este intervalo contendrá un grupo binario clásico, A, el mismo que representa un cierto valor de incertidumbre acerca del valor autentico de dicho número.

Entonces si se supone que  $A = [a_1, a_3]$  esto quiere decir que el número debe encontrarse entre  $a_1$  y  $a_3$  es decir el número no puede ser mayor que  $a_3$  y no puede ser menor que  $a_1$  por lo que dicho número se puede expresar por la siguiente función de pertenencia:

$$\begin{aligned} m_A(x) &= 0, & x < a_1 \\ &= 1, & a_1 \leq x \leq a_2 \\ &= 0, & x > 3 \end{aligned}$$

### 1.3.2.2. NÚMEROS DIFUSOS NORMALES Y CONVEXOS CON $\alpha$ CORTES

Si se toma un nuevo número pero en este caso un número difuso como un subconjunto difuso normal y convexo en R.

Para que este sea convexo implicara que la  $\alpha$  cortes  $A^\alpha = [a_1^{(\alpha)}, a_3^{(\alpha)}]$  estarán anidados, es decir:

$$(\alpha' < \alpha) \Rightarrow (a_1^{(\alpha')} \leq a_1^{(\alpha)}, a_3^{(\alpha')} \geq a_3^{(\alpha)})$$

Mientras que para que sea normal implicara que:

$$\exists x \in R \therefore m_A(x) = 1$$

Ahora si se representa la  $\alpha$  corte como  $A_\alpha = [a_1^{(\alpha)}, a_3^{(\alpha)}]$ , la condición de convexidad implica, como puede verse en la figura 6 que:

$$(\alpha' < \alpha) \Rightarrow (A^\alpha \subset A^{\alpha'})$$



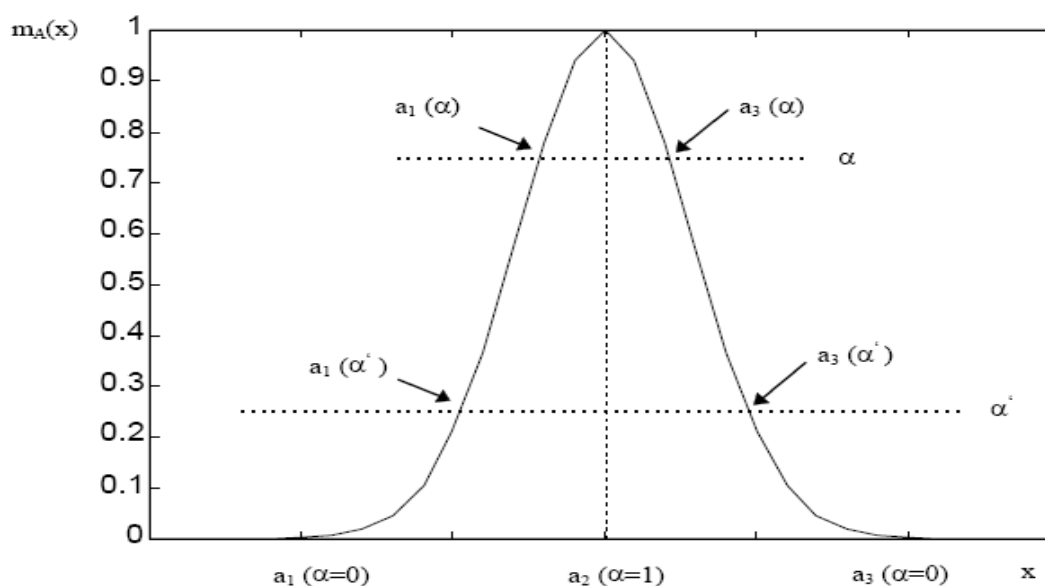


Figura 1.6 NÚMERO NORMAL Y CONVEXO CON A-CORTES

Cuando se realizan operaciones matemáticas con números difusos estas se pueden establecer en base a intervalos de confianza.

A continuación vamos a indicar la forma de realizar las operaciones más importantes con números difusos basándonos en los intervalos de confianza, tendremos entonces:

- **Suma:**

$$[a_1, a_3](+)[b_1, b_3] = [a_1 + b_1, a_3 + b_3]$$

- **Resta:**

$$[a_1, a_3](-)[b_1, b_3] = [a_1 - b_3, a_3 - b_1]$$

- **Multiplicación:**

$$[a_1, a_3](\cdot)[b_1, b_3] = [a_1 \cdot b_1 \cap a_1 \cdot b_3 \cap a_3 \cdot b_1 \cap a_3 \cdot b_3, a_1 \cdot b_1 \cup a_1 \cdot b_3 \cup a_3 \cdot b_1 \cup a_3 \cdot b_3]$$

- **Inverso:**

$$[a_1, a_3]^{-1} = [a_1, a_3]^{-1} = \left[ \frac{1}{a_3}, \frac{1}{a_1} \right]$$

- **División:**

$$[a_1, a_3](\div)[b_1, b_3] = \left[ \frac{a_1}{b_1} \cap \frac{a_1}{b_3} \cap \frac{a_3}{b_1} \cap \frac{a_3}{b_3}, \frac{a_1}{b_1} \cup \frac{a_1}{b_3} \cup \frac{a_3}{b_1} \cup \frac{a_3}{b_3} \right] \text{ excepto para } a_1 \leq 0 \leq a_3$$

Si el conjunto difuso está definido sobre los  $\mathbb{R}_+$ , sería:

$$[a_1, a_3](\div)[b_1, b_3] = \left[ \frac{a_1}{b_3}, \frac{a_3}{b_1} \right]$$

- **Mínimo:**

$$[a_1, a_3](\cap)[b_1, b_3] = [a_1 \cap b_1, a_3 \cap b_3]$$

- **Máximo:**

$$[a_1, a_3](\cup)[b_1, b_3] = [a_1 \cup b_1, a_3 \cup b_3]$$

- **Multiplicación por un número real:**

Podemos representar un número real como:  $a = [a, a]$ ,  $a \in \mathbb{R}$ , por tanto:

$$a[b_1, b_3] = [a \cdot b_1 \cap a \cdot b_3, a \cdot b_1 \cup a \cdot b_3]$$

Todos las operaciones que se ha indicado en la parte anterior son validas tanto para números difusos naturales (N) como para enteros (Z).

### 1.3.2.3. NUMEROS DIFUSOS TRIANGULARES

En la parte práctica los números difusos más usados son los triangulares ya que se dispone de una manipulación relativamente cómoda, aunque muchos autores han cuestionado la utilización de este tipo de números difusos como una solución general.

Obviamente este tipo de números difusos son la versión mas simple del tipo L-R, los mismos que ya fueron revisados anteriormente.

A continuación en la figura 7 se puede ver la forma que toman los números difusos triangulares (NDT), los mismos que serán definidos mediante  $(a_1, a_2, a_3)$ , y su función de pertenencia sería:

$$\begin{aligned} m_A(x) &= 0, & x < a_1 \\ &= \frac{(x - a_1)}{(a_2 - a_1)}, & a_1 \leq x \leq a_2 \\ &= \frac{(a_3 - x)}{(a_3 - a_2)}, & a_2 \leq x \leq a_3 \\ &= 0, & x > a_3 \end{aligned}$$

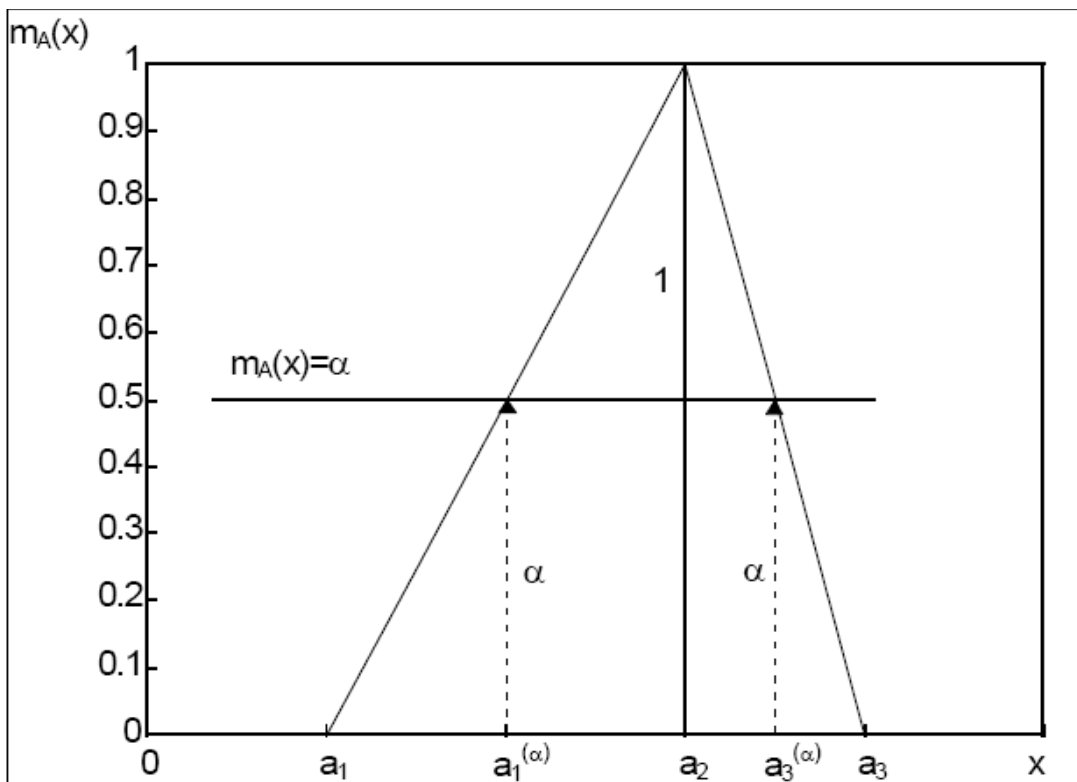


Figura 1.7 REPRESENTACIÓN GRÁFICA DE UN NÚMERO DIFUSO TRIANGULAR A ( $a_1, a_2, a_3$ )

#### 1.3.2.4. NUMEROS DIFUSOS TRAPEZOIDALES

Como se vio anteriormente los números difusos triangulares son los más utilizados debido a la facilidad que estos implican en su uso, pero existen otro tipo de números difusos que aunque no son tan usados como los triangulares son muy importantes. Los números trapezoidales son la versión mas sencilla de los conceptos de intervalos difusos L-R de los cuales ya se ha hablado anteriormente. En este tipo de números no se obtiene un cruce en  $\alpha=1$ , en vez de esto se tendrá una línea recta entre el intervalo ( $a_2, a_3$ ) y la función de dicho número se la puede expresar de la siguiente manera:

$$\begin{aligned}
 m_A(x) &= 0, \quad x < a_1 \\
 &= \frac{(x - a_1)}{(a_2 - a_1)}, \quad a_1 \leq x \leq a_2 \\
 &= 1, \quad a_2 \leq x \leq a_3 \\
 &= \frac{(a_4 - x)}{(a_4 - a_3)}, \quad a_3 \leq x \leq a_4 \\
 &= 0, \quad x > a_4
 \end{aligned}$$

A continuación en la figura 8 se ha realizado el grafico de un número difuso trapezoidal  $A(a_1, a_2, a_3, a_4)$

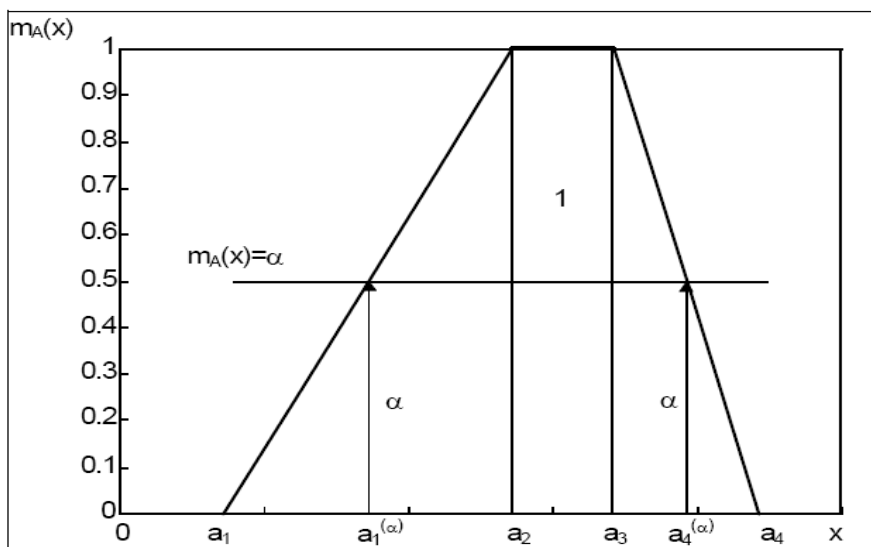


Figura 1.8 REPRESENTACIÓN DE UN NÚMERO DIFUSO TRAPEZOIDAL  $A(a_1, a_2, a_3, a_4)$

### 1.3.2. SISTEMAS DE INFERENCIA DIFUSOS

El conjunto de proposiciones IF-THEN con las que se modela el problema planteado son denominadas como reglas difusas y tienen la siguiente forma:

“Si  $a$  es  $A$  entonces  $b$  es  $B$ ”

En donde los conjuntos difusos  $A$  y  $B$  se encuentran definidos en los rangos dados por “ $a$ ” y “ $b$ ” respectivamente.

Generalmente la expresión “ $a$  es  $A$ ” es conocida como premisa o antecedente, mientras que “ $b$  es  $B$ ” se conoce como consecuente o conclusión.

#### 1.3.3.1 IMPLICACIÓN DIFUSA

La función característica de la expresión “Si a es A ENTONCES b es B” será,

$m_{A \rightarrow B}(x,y)$ , conociéndola como implicación lógica.

A continuación se presenta la tabla de verdad de la implicación lógica tradicional.

<b>P</b>	<b>Q</b>	<b>P <math>\rightarrow</math> Q</b>
V	V	V
V	F	F
F	V	F
F	F	V

Cada una de la proposiciones IF-THEN es a la vez un conjunto difuso, cuya función característica mide el grado de verdad de la relación de implicación entre x e y.

Funciones características asociadas, extraídas de aplicar analogías entre operadores son:

$$m_{A \rightarrow B}(x,y) = 1 - \min[m_A(x), 1 - m_B(y)]$$

$$m_{A \rightarrow B}(x,y) = \max[1 - m_A(x), m_B(y)]$$

En el siguiente ejemplo se explica con mas detalle la funcion de las proposiciones IF-THEN.

Supongamos que i a=5, b=3 y c=10, entonces tendremos:

“Si a = b ENTONCES c=c”. Lo que esto implica o trata de demostrar es que un falsedad (a=b) puede implicar una verdad (c=c).

La utilidad práctica de dichas proposiciones en la ingeniería solo tendrán valor si se trata de dos proposiciones verdaderas.

## 1.4 SISTEMAS COMERCIALES QUE UTILIZAN LA TÉCNICA DIFUSA.

El Sistema más espectacular funcionando actualmente, es el Metro de la Ciudad Japonesa de Sendai. Desde 1987 es usado un Sistema de Control Difuso para mantener los trenes rodando rápidamente a lo largo de la ruta, frenando y acelerando suavemente, deslizándose entre las estaciones, parando con precisión sin perder un solo segundo o sacudir fuertemente a los pasajeros.

A continuación se nombrara algunas compañías con sus productos en donde tienen como base la utilización de la lógica Difusa:

- Los japoneses manipulan productos gigantes como el *Matsushita*, una máquina aspiradora automática que limpia y lava. La máquina de lavado, evalúa la carga y ajusta por sí misma, el detergente necesario, la temperatura del agua y el ciclo de lavado.
- El televisor SONY, que automáticamente ajusta contraste, brillo y tonalidades de color.
- En la tecnología automotriz, los coches NISSAN, poseen transmisión automática y frenos antibloqueo.
- Las industrias Mitsubishi, tienen designado un sistema de control difuso para elevadores, mejorando la eficiencia en el procedimiento manual que siempre se presenta cuando grandes grupos esperan para tomar el ascensor al mismo tiempo.
- En EE.UU. la compañía de elevadores OTIS, está desarrollando sus propios productos para horarios en ascensores según las demandas de variaciones de tiempos.

### **1.4.1. ÁREAS DE APLICACIÓN QUE TIENE LA LÓGICA DIFUSA.**

Se puede asegurar y casi garantizar un amplio campo de aplicaciones con un alto grado de interés. Entre otras podemos mencionar las siguientes:

- **En el área médica:**

Se ha empleado para el diagnóstico, desde campos como la acupuntura, hasta el análisis de los ritmos cardíacos.

- **Productos al consumidor:**

Lavadoras

Hornos de microondas

Procesadores de arroz

Limpiadores al vacío

Cámaras de video

Televisores

Sistemas térmicos

Traductores.

- **Sistemas:**

Elevadores

Trenes

Automóviles (máquinas, transmisiones, frenos)

controles de tráfico.

- **Software:**

Diagnóstico Médico

Seguridad

Compresión de datos

### **1.5 VENTAJAS DEL SISTEMA DE CONTROL DIFUSO.**

Se puede nombrar algunas ventajas que tiene al utilizar un sistema de control difuso:



- Pueden ser evaluadas mayor cantidad de variables.
- Variables lingüísticas, no numéricas, son usadas simulando la vía del conocimiento humano.
- Relaciona entradas y salidas, sin tener que entender todas las variables, permitiendo que el sistema pueda ser más fiable y estable que uno con un Sistema de Control convencional.
- Simplifica asignación de soluciones previas a problemas sin resolver.
- Es posible obtener prototipos rápidamente, ya que no requiere conocer todas las variables acerca del Sistema antes de empezar a trabajar.
- El desarrollo de éstos es más económico que el de sistemas convencionales, porque son más fáciles de designar.
- Simplifican la adquisición y representación del conocimiento.
- Unas pocas reglas abarcan gran cantidad de complejidades.
- Por medio de ellos es más cómodo designar rápidamente un prototipo que en los sistemas convencionales. Los Sistemas Difusos requieren mayor simulación y una excelente depuración y prueba antes de que sean operacionales.

### **EL NUEVO RETO DE LA LÓGICA DIFUSA: INTERNET.**

Según algunos prestigiosos investigadores en Internet, parece que el futuro para abordar la ingente cantidad de datos, recuperar información, “controlar” y gestionar la red, pasa por el uso de tecnologías borrosas. Prueba de ello fué la celebración del primer encuentro sobre lógica difusa e Internet (FLINT 2001, se puede acceder a las ponencias a través de la página Web de BISC) celebrado en la universidad de Berkeley en el verano de 2001 y organizado por el propio profesor Zadeh.

La idea principal es la tendencia hacia el *Computing with words* (computación con palabras), usando básicamente técnicas de *Soft Computing* (que engloba básicamente a la lógica difusa, las redes neuronales y la computación evolutiva – algoritmos genéticos). Estos términos, acuñados por el

profesor Zadeh, ponen de manifiesto el nuevo reto que ha de acometer *el soft computing* y que ya se plasman en diferentes líneas de investigación, como son:

- Una nueva generación de motores de búsqueda en Internet, que usan técnicas de *soft computing* y que tratan de mejorar la búsqueda actual (lexicográfica) usando una búsqueda conceptual.
- Técnicas avanzadas para describir “perfiles de usuario” que permitan un uso de internet más inteligente “a la carta”.
- Comercio electrónico basado en técnicas de *soft computing* (Por ejemplo lo que el profesor Mamdani denomina *Soft Knowledge*).
- *Semantic Web*.

Esta lista se podría extender a otros muchos aspectos, pero conviene destacar algunas técnicas que deben ser mejoradas y adaptadas a los nuevos tiempos, volúmenes y estructura de la información que es necesario manipular en Internet:

- Nuevos modelos de representación del conocimiento, como por ejemplo los conceptual fuzzy sets del profesor Takagi.
- Mejoras en los métodos de agregación de información y en los algoritmos de clasificación y clustering.
- Técnicas para la generación de ontologías (taxonomías de conceptos de un determinado dominio) y para la utilización de las mismas con limitaciones en el tiempo de respuesta.
- Agentes inteligentes autónomos para realizar servicios en Internet (en búsqueda, chat, correo electrónico, comercio electrónico, envío de datos, multimedia...).

- Técnicas de gestión y extracción de conocimiento en bases de datos (Data Mining y *Data Warehouse*).

## **CAPITULO 2**

### **ANÁLISIS DEL PROYECTO.**

#### **2.1 SISTEMAS DE CONTROL NORMALES.**

Para iniciar el análisis del proyecto es necesaria una breve descripción de los sistemas de control existentes para que así el desarrollo del proyecto se haga más sencillo de entender.

Un sistema de control es un sistema físico que se controla así mismo o a otro sistema o sistemas de control.

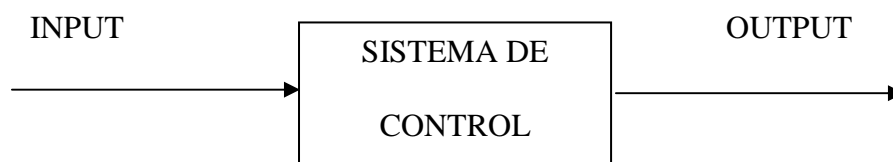
Existen muchos ejemplos de sistemas de control y no necesariamente tienen que ver con la electrónica. Algunos ejemplos de dichos sistemas son:

- Un interruptor eléctrico controlando el flujo de corriente.
- Una persona que controla la velocidad y la dirección de un automóvil es decir una persona conduciendo un automóvil.
- Un sistema que mediante un termostato controle la temperatura de una oficina o una habitación.

Un sistema de control típico tiene una o más entradas, una o más salidas y un bloque en el cual se realiza el procesamiento del sistema. El tipo de procesamiento que ocurre dentro del sistema de control está clasificado como un sistema de control ya sea de lazo abierto (open loop) o de lazo cerrado (closed loop).

### 2.1.1 SISTEMA DE CONTROL DE LAZO ABIERTO.

A continuación en la Figura 9 se presenta un diagrama de bloques en el cual se puede observar un sistema de control típico de lazo abierto (open loop).



**Figura 2.1 SISTEMA DE CONTROL DE LAZO ABIERTO**

Un sistema de control de lazo abierto es aquel en el que la acción de control es independiente de la salida. En otras palabras la salida que se tiene del sistema de control es directamente emparentada con su entrada.

Un sistema de control de lazo abierto toma sus entradas, realiza el procesamiento sobre estas entradas y produce una salida.

Una calculadora es un ejemplo de un sistema de lazo abierto. La entrada esta dada por las teclas que se oprimen y la operación que se desee realizar, la calculadora toma dicha entrada, opera la misma según el código que se le ha indicado, y produce una salida dada como el resultado de la operación realizada.

### 2.1.2 SISTEMA DE CONTROL DE LAZO CERRADO.

Un sistema de lazo cerrado como el que se muestra a continuación en la Figura 10 difiere del sistema de lazo abierto en que su salida depende tanto de su entrada como de su salida.

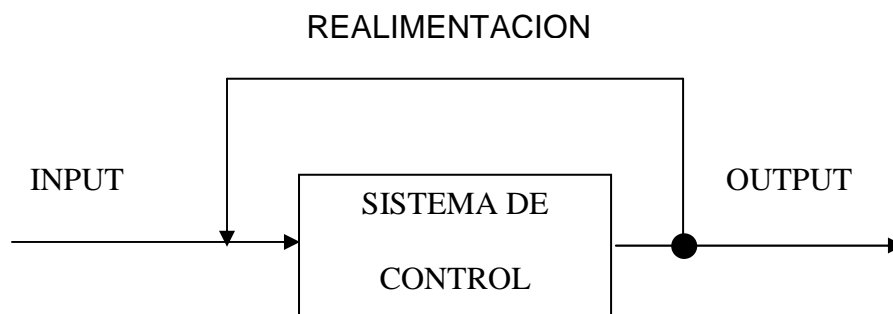


Figura 2.2 SISTEMA DE CONTROL DE LAZO CERRADO

La salida se produce en función de la entrada y en función de la salida debido al lazo de realimentación. En la mayoría de sistemas, el lazo de realimentación forma una entrada extra al sistema.

La conducción de un automóvil es un ejemplo muy claro de un sistema de control de lazo cerrado. La Persona ajusta la velocidad del auto mediante la presión del pedal de aceleración, la realimentación en este sistema esta dada por la persona, quien regula el flujo de combustible al motor basándose en el velocímetro. Así si la persona desea incrementar la velocidad a 55 KMH esta presionara el pedal de aceleración, y el automóvil acelerara.

El automóvil continúa su aceleración hasta que la persona observa que el velocímetro muestra una velocidad de 55KMH, y disminuye la presión del pedal de aceleración.

### **2.1.3 SISTEMAS DE CONTROL PROPORCIONALES INTEGRALES DERIVATIVOS (PID).**

Los sistemas de control mas usados comúnmente en nuestros días están basados en el método Proporcional Integral Derivativo (PID).

James Watt es acreditado frecuentemente por la creación de la teoría clásica de control en la cual se basa el método PID.

Los sistemas basados en el método PID tratan de mantener su salida en un nivel predeterminado.

Para mantener una salida constante, el controlado mide la salida y controla la entrada (en general un sensor) y aplica modelos matemáticos a la diferencia de las dos la misma que es denominada como error.

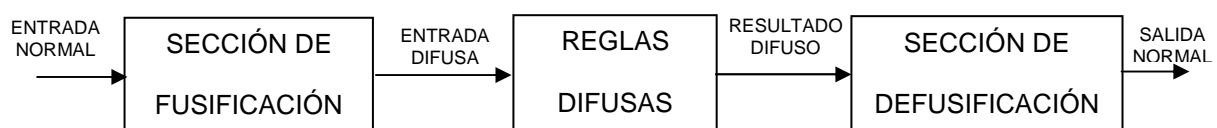
- **Proporcional:** El error es multiplicado por una constante proporcional  $P$ , y enviada a la salida.  $P$  representa el lugar donde la salida del controlador es proporcional al error del sistema.
- **Integral:** El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un periodo de tiempo. Luego es multiplicado por una constante  $I$  y se agrega a la salida proporcional.  $I$  representa el error de estado estacionario del sistema.
- **Derivativo:** El error se deriva con respecto al tiempo y se multiplica por una constante  $D$  y luego se suma a las señales anteriores. La función de la derivativa es gobernar la respuesta de control a los cambios en el sistema ya que una mayor derivativa corresponde a un cambio más rápido y el controlador puede responder acordeamente.

Por tener una exactitud mayor a los controladores proporcional, proporcional derivativo y proporcional integral se utiliza en aplicaciones más cruciales tales como control de presión, flujo, química, fuerza, velocidad y otras variables, además es utilizado en reguladores de velocidad de automóviles.

## 2.2 SISTEMAS DE CONTROL DIFUSOS.

En general un sistema de control que funciona con lógica difusa esta formado por tres componentes. El primero esta conformado por la sección de fusificación, el segundo formado por las reglas difusas y el tercero que lo forma la sección de la defusificación.

A continuación en la Figura 11 se muestra un diagrama de bloques para un sistema de control difuso.



**Figura 2.3 SISTEMA DE CONTROL DIFUSO**

### **2.2.1 SECCIÓN DE FUSIFICACIÓN.**

También se denomina proceso de Borrosidad, es la primera sección de un control difuso, y su tarea es convertir una señal de entrada, en un conjunto de variables difusas es por esto que dicha sección lleva el nombre de sección de clasificación difusa o sección de fusificación.

Dicho trabajo de fusificación se realiza asignando diversos valores a partir de un conjunto de funciones de pertenencia o membresía, los mismos que serán nuestros valores difusos con los que trabajaran las siguientes secciones del sistema de control.

Los valores de cada una de las funciones de pertenencia se etiquetan mediante una función sea esta  $m(x)$ , y son determinados por el valor de entrada de la señal sea esta  $x$ , y la figura de cada función de membresía.

### **2.2.2 SECCIÓN DE REGLAS DIFUSAS.**

Esta sección toma los datos que han sido fusificados en la sección anterior, los mismos que son sometidos a las reglas difusas que se tengan.

Es decir, el bloque de inferencia es aquel en el cual se realiza la “traducción matemática” de las reglas difusas: estas reglas modelan el sistema pero para poder trabajar con ellas y extraer un resultado se debe de evaluar matemáticamente la información que reflejan. Como ya se ha mencionado



anteriormente, las reglas más utilizadas para diseñar un sistema basado en lógica difusa toman la forma:

“**SI** Antecedente 1 **Y** Antecedente 2... **ENTONCES** Consecuente 1 **Y**...”

En el caso lógico proporcional, evaluar la regla **SI – ENONCES** es sencillo. Si la premisa es verdadera, la conclusión es verdadera, por lo contrario, se genera un problema cuando el antecedente es difuso, pues no se sabe como se vera afectado el consecuente.

Así, las reglas difusas permiten expresar el conocimiento que se tiene acerca de la relación entre los antecedentes y los consecuentes en un cierto grado de verdad en donde si el antecedente es verdadero en cierto grado, entonces el consecuente es verdadero en cierto grado.

Pero una regla puede no ser siempre aplicable, de la misma forma, hay ocasiones que es preciso modificarla para adaptarse a una cierta situación que es cambiante del sistema.

En ocasiones, una solución a este problema consiste en añadir una nueva variable de entrada al controlador y añadir nuevas condiciones en el antecedente de algunas reglas, basadas en la nueva variable.

Existen muchas formas de modificar una regla. Por ejemplo, modificar la etiqueta lingüística de alguna variable (del antecedente o del consecuente de la regla).

### **Ejemplo:**

- SI Humedad es GRANDE, ENTONCES Apertura \_ riego es POCO.

Podría pasar a la siguiente regla (si, por ejemplo, es verano):

- SI Humedad es GRANDE, ENTONCES Apertura \_ riego es MODERADA.

Los sistemas de control difusos son generalmente del tipo de sistemas realimentados. Consecuentemente la realimentación en un sistema de control difuso ocurre dentro de la sección de reglas difusas.

El resultado que se obtiene de la sección de reglas difusas esta dado por una salida de datos difusos, luego estos datos difusos pasan a la siguiente sección que se la denomina de defusificación.

### **2.2.3 SECCIÓN DE DEFUSIFICACIÓN.**

El último paso en la construcción de un sistema de control lógica difusa es convertir de nuevo las variables difusas generadas por la base de reglas difusas en valores con interpretación real. El proceso se denomina 'defuzzificación' por que combina las variable difusas para generar el valor real de la señal (también comúnmente denominado CRISP o no-difuso) el cual puede ser usado posteriormente para generar

Acciones de control. Por ejemplo, en el caso de un sistema de control para tanques acoplados, la señal 'CRISP' seria un voltaje que puede ser usado para operar el amplificador en el manejador de la bomba.

Difuso después de aplicar las reglas difusas y evaluar las variables difusas, necesitaremos trasladar estos valores nuevamente hacia el mundo real.

Requeriremos entonces de una función de membresía (membership functions) para cada una de las variables de salida.

Esta sección del sistema toma los datos fusificados de la sección anterior y los transforma en datos reales no difusos para que puedan ser analizados de manera normal.

Se han propuesto diversos mecanismos para poder encontrar un valor concreto de un valor difuso, el mismo que represente fielmente a la totalidad del concepto difuso.

Aunque no existe un estándar universalmente aceptado, los siguientes son algunos métodos usados generalmente para la defusificación.

- Método del Centro del área (CoA) que también es llamado Centro de gravedad (CoG).
- Método Centro del Máximo (CoM).
- Media Ponderada.
- Singleton
- Método de Larsen

### **2.2.2.1 MÉTODO DEL CENTRO DE GRAVEDAD (COG) O DEL CENTRO DEL ÁREA (COA).**

Utiliza como salida del sistema el centro de gravedad de la función característica de salida.

Matemáticamente dicho método se calcula con la siguiente manera:

$$\bar{y} = \frac{\int y m_B(y) dy}{\int m_B(y) dy}$$

Es uno de los métodos mas utilizados en las aplicaciones de lógica difusa referidas a la ingeniería ya que se obtiene una solución única, aunque su punto débil radica en la dificultad de los cálculos.

### **2.2.2.2 MÉTODO DE EL CENTRO MÁXIMO (COM)**

Se elige como valor para la variable de salida aquel para el cual la función característica del conjunto difuso de salida es máxima.

Matemáticamente se define de la siguiente manera:

$$\text{salida} = \frac{V_{\text{centro}}(\text{función de pertenencia variable de salida 1}) \times V(\text{regla1}) + V_{\text{centro}}(\text{función de pertenencia variable de salida 2})}{V(\text{regla1}) + V(\text{regla2})}$$

En general no es un método óptimo, ya que este valor máximo puede ser alcanzado por varias salidas.

### 2.2.3 METODO DE DISEÑO

Para el diseño existen algunos métodos pero el más fácil de seguir es el siguiente:

1. Se identifican todas las entradas y salidas, usando variables lingüísticas.
2. Se asignan funciones de membresía a cada una (fusificación), según el criterio que proporciona el conocimiento del sistema a controlar.
3. Se construyen las reglas difusas que son de la siguiente forma:

Si A1 conexión A2 entonces C. Es decir, si ocurren los antecedentes A1 y/o A2, el resultado es el consecuente C, o en términos de control, si la etiqueta de la primera variable medida es A1, y la de la segunda es A2 (en caso de ser 2), formándose el conjunto difuso solución mediante la llamada agregación de los resultados así obtenidos.

4. Se defusifica la acción de control, obteniéndose un valor numérico en vez de un difuso.

A continuación se presenta un ejemplo con todos los pasos, tanto de fusificación, reglas difusas y su defusificación.

#### Ejemplo 4

Se ha hecho un estudio que indica que la edad del crecimiento de los 10 a los 14 años provoca un incremento promedio de peso de 3.7Kg/año, y la estatura promedio en cm. es incrementada en 6.2 cm. /año. Se sabe que para un desarrollo armónico se necesita:

- Carbohidratos, para fuente de energía, obtenidos en los vegetales como verduras y frutas cítricas, y que proporcionan al organismo vitaminas A, B1, B2, C, minerales como Fe, Ca.
- Proteínas, para el crecimiento y mantenimiento del organismo, que provienen de alimentos como la carne, peces, huevos, y que proporcionan vitaminas A, B1, B2, minerales como Ca, Fe, Co, I.
- Grasas, que sirven como reserva en el organismo, para épocas de necesidad, existentes en la leche, el queso, el tocino, margarina, mantequilla, y que proporcionan vitaminas A, B1, B2 y Ca.

Se conoce además que la necesidad diaria de proteínas es de 85 gramos, no pudiendo ser menos de 50 ni más de 100. Debe indicarse que el problema completo exigirá conocer los requerimientos diarios de los demás grupos de alimentos mencionados.

Se debe proponer un sistema difuso que vaya indicando en cada período, la cantidad de proteína diaria que requeriría un adolescente en tratamiento o que lo está iniciando, para tener un desarrollo normal.

Siguiendo el procedimiento de diseño, se identifican las variables lingüísticas PESO, referente al incremento de peso anual, y ESTATURA, para el incremento anual de estatura, como variables medidas, y la variable PROTEINA, como la variable de control, referente a la cantidad de proteínas diarias a ser suministradas.

La segunda etapa corresponde a la fusificación de las variables:

- Para PESO y en vista de que no se conocen otros datos estadísticos, se escoge un universo de discurso de 0 a 7.7 Kg. /año con 7 funciones de membresía, prefiriendo siempre escoger un número impar de estas funciones, para dar una cierta simetría al conjunto obtenido, y tener la normalidad en el valor promedio establecido de las estadísticas como se muestra en la figura 12. Es también útil mencionar que se pueden normalizar los universos de discurso, lo que por un lado permite estandarizar el diseño, y por otro lado, mejorar la precisión cuando este universo es muy estrecho.

Las etiquetas escogidas para cada función son las siguientes:

PMB: incremento de peso muy bajo.

PB: incremento de peso bajo.

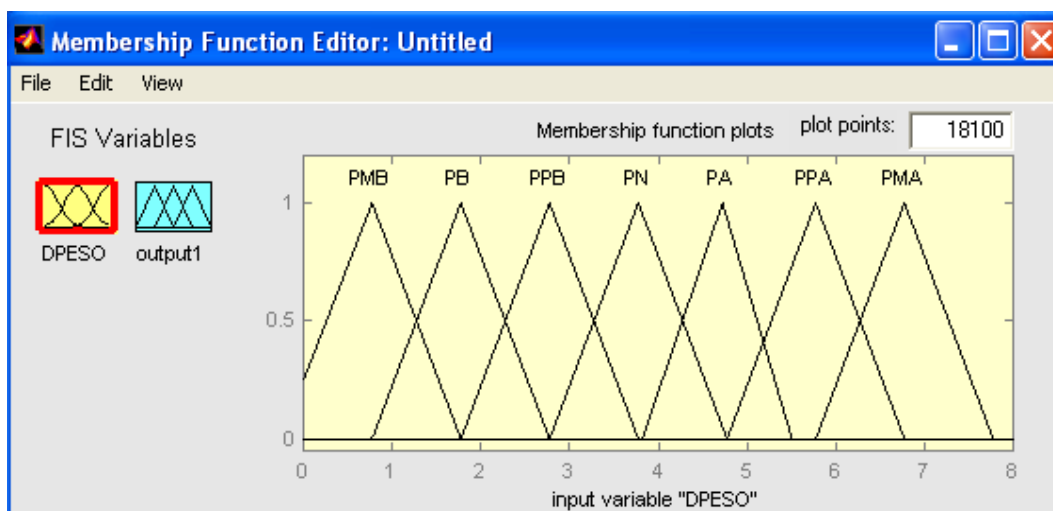
PPB: incremento de peso un poco bajo.

PN: incremento de peso normal.

PPA: incremento de peso un poco alto.

PA: incremento de peso alto.

PMA: incremento de peso muy alto.



**Figura 2.4 FUSIFICACIÓN DE LA VARIABLE DPESO**

- Para ESTATURA, por razones parecidas, se escoge un universo aproximado de 0 a 13 y un soporte de 0.2 a 12.2, con 5 funciones de membresía. Cae indicarse que en un principio se podría trabajar solamente con 3 hasta incorporar una cantidad suficiente de conocimiento. El resultado se muestra en la Figura 13. Teniendo las siguientes etiquetas:

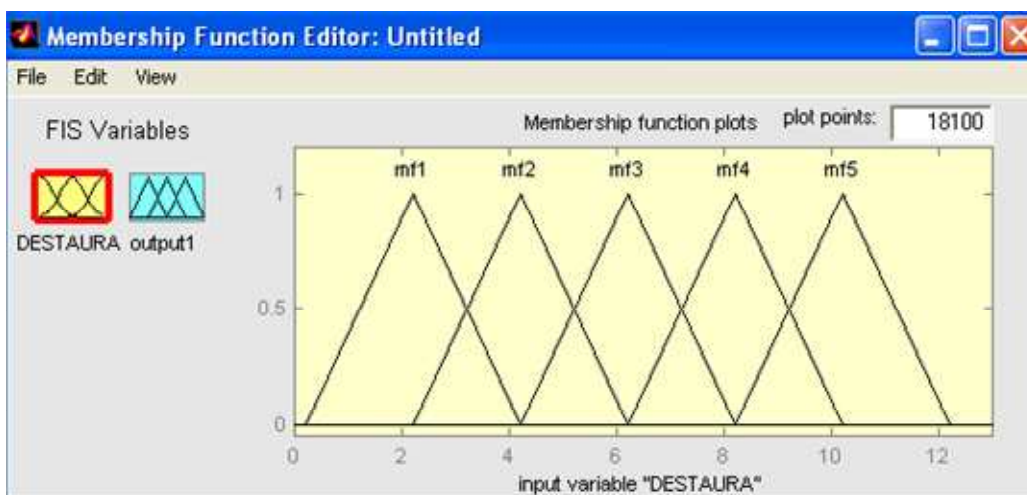
EB: Incremento de estatura bajo

EPB: incremento de estatura un poco bajo.

EN: incremento de estatura normal.

EPA: incremento de estatura un poco alto.

EA: incremento de estatura alto.



**Figura 2.5 FUSIFICACIÓN DE LA VARIABLE DESTAURA**

- Para PROTEINA, el universo va de 60 a 110, con 5 funciones de membresía, siendo las de los extremos trapezoidales y cortadas en 60 y 110; podría dejarse que continúen en 1 para otros casos. Esta limitación corresponde al mínimo y máximo de proteínas que deben suministrarse en una dieta, siendo entonces una restricción física. El gráfico respectivo se muestra en la figura 14. Las etiquetas escogidas son:

PRMB: proteínas muy bajas.

PRB: proteínas bajas.

PRN: proteínas altas.  
 PRA: proteínas altas  
 PRMA: proteínas muy altas.

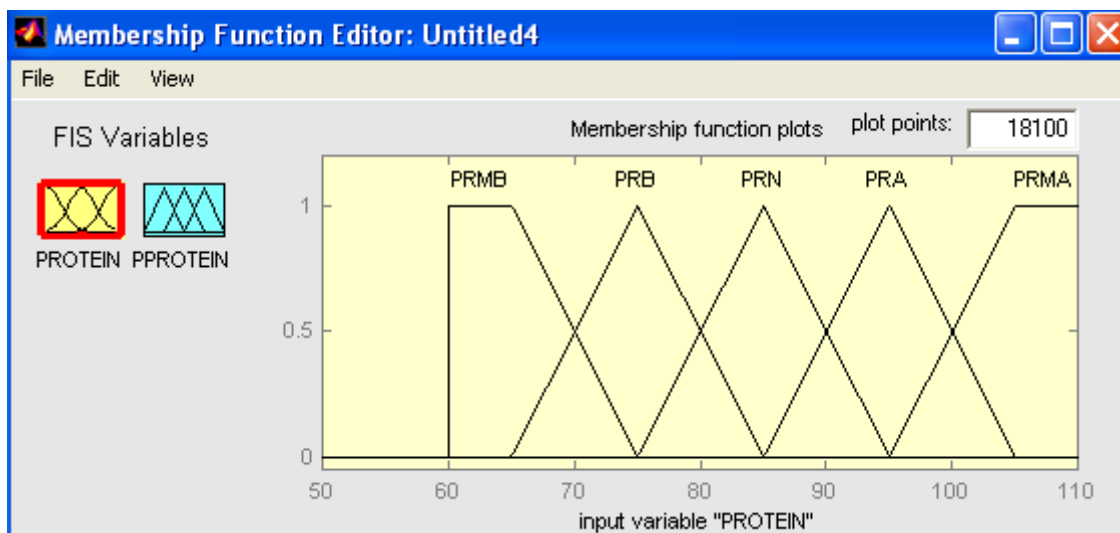


Figura 2.6 FUSIFICACIÓN DE LA VARIABLE DE CONTROL PROTEIN

La tercera fase corresponde a la elaboración de lo que se llama Base de Reglas, que definen la relación de las variables medidas o las de control.

Los resultados para las variables utilizadas se muestran a continuación:

PESO ESTATURA	PMB	PB	PPB	PN	PPA	PA	PMA
	EB	11. PRMA	12. PRMA	13. PRA	14. PRN	15. PRN	16. PRN
EPB	21. PRMA	22. PRA	23. PRA	24. PRN	25. PRN	26. PRN	27. PRB
EN	31. PRA	32. PRA	33. PRN	34. PRN	35. PRN	36. PRB	37. PRB
EPA	41. PRA	42. PRN	43. PRN	44. PRN	45. PRB	46. PRB	47. PRMB



<b>EA</b>	51. <i>PRA</i>	52. <i>PRN</i>	53. <i>PRN</i>	54. <i>PRN</i>	55. <i>PRB</i>	56. <i>PRMB</i>	57. <i>PRMB</i>
-----------	-------------------	-------------------	-------------------	-------------------	-------------------	--------------------	--------------------

**Tabla 2.1 BASE DE REGLAS DIFUSAS**

La información para elaborar la tabla anterior se presenta a continuación:

En la primera fila y en la primera columna, se colocan los antecedentes, es decir, las variables medidas, asignando indistintamente a una la columna y a la otra la fila, de manera ordenada, empezando en la etiqueta de mayor valor y terminando en la etiqueta de menor valor o viceversa.

Dentro de cada cuadro de la tabla, va un consecuente, es decir una etiqueta de la variable de control, asignándose una función de membresía a cada casilla, que se ha enumerado para simplificar su referencia.

Para facilitar el llenado de la tabla, se parte de la casilla central, existiendo una sola si el número de etiquetas es impar para cada variable medida. Comúnmente, a esta casilla (en este caso la N° 34) debería corresponder el resultado normal o su equivalente.

Luego se procede con los extremos, asignando los valores máximos o mínimos, según la aplicación; por ejemplo, en este caso, un adolescente con menos incremento de estatura y de peso requerirá un máximo de proteínas, según se ha asignado a la casilla N°11. En la casilla N° 57, de los adolescentes con mayores índices de crecimiento en cambio se ha colocado la etiqueta de la menor cantidad de proteínas diarias, es decir PRMB.

Se procede luego a organizar el valor de las restantes casillas, de preferencia de manera simétrica, incrementando o decrementando los valores de manera paulatina. Debe mencionarse que no siempre se llenan todas las casillas.

Estas reglas sirven para que el sistema infiera de manera correcta el parámetro de control. Así por ejemplo, la regla N° 45 se lee:

Si EPA (Incremento de estatura un poco alta)  $\cap$  (“y”, pero puede ir “o” dependiendo de la aplicación) PPA (incremento de peso un poco alto)  $\rightarrow$  (implica que) PRB (La cantidad de proteínas diaria debe ser baja). En resumen:

$$\text{Si EPA } \cap \text{ PPA } \rightarrow \text{ PRB}$$

Por ultimo, para la defusificación como se ha visto anteriormente se pueden utilizar varios métodos, unos más precisos que otros, dependiendo de si el modelo empleado es el de Mandami o Takagi-Sugeno. Para este ejemplo se emplea el modelo de Mandami, y los métodos mas usados.

A fin de que se comprenda, se usara el caso de un adolescente cuyo incremento de peso a sido de 2Kg. /año, y su incremento de estatura de 3cm. /año.

De las figuras 12 y 13 y la tabla 2.1, resulta primero que un incremento de peso de 2 corresponde a una etiqueta PB con una función de membresía de 0.7 y una etiqueta PPB con una función de membresía de 0.3.

Respecto al incremento de estatura de 3, pertenece a la etiqueta EB con un valor de 0.6 y la EPB con una función de membresía de 0.4.

Esto provoca que las reglas de la tabla que se activan sean las número 12, 13,22 y 23 según lo que se resume a seguir:

- 12)  $EB \cap PB$  ;  $0.6 \cap 0.7 = 0.6$  (Etiqueta de la variable de control: PRMA).
- 13)  $EB \cap PPB$  ;  $0.6 \cap 0.3 = 0.3$  (PRA).
- 22)  $EPB \cap PB$  ;  $0.4 \cap 0.7 = 0.4$  (PRA).
- 23)  $EPB \cap PPB$  ;  $0.4 \cap 0.3 = 0.3$  (PRA).

Siguiendo con el proceso de diseño en este ejemplo que se a tomado se nombrara los métodos de defusificación.

**a. MAX – MIN**

Con los resultados obtenidos, se forma el conjunto difuso de salida, para la defusificación se empleara la regla del máximo:

$$0.6V \quad 0.3V \quad 0.4V \quad 0.3 = 0.6$$

Resultando ser el máximo de los valores mínimos de las funciones de membresía obtenidos.

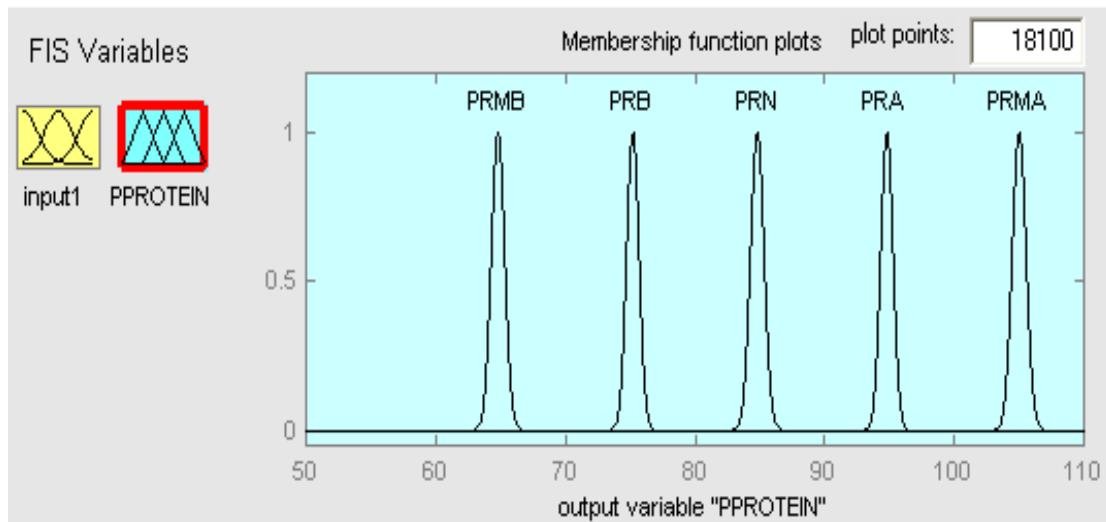
Por ultimo se asigna a la variable de control el valor correspondiente a 0.6.

$$\text{PROTEIN} = 60 + (110 - 60) * 0.6 = 90 \text{ gm/día}$$

Que es la cantidad diaria que debe suministrarse al adolescente.

**b. EMPLEANDO CONJUNTOS SINGLENTON.**

En este caso, la variable PROTEN tiene la forma de la Figura.... , donde se han asignado funciones singlenton en los valores medios de los rangos, que en las etiquetas intermedias corresponde a los vértices de los triángulos definidos antes, y para los trapecios de los extremos, se asignan estas funciones donde el trapecio termina o inicia la base superior (65 y 105 respectivamente), quedando equidistantes las funciones.



**Figura 2.7 FUSIFICACIÓN DE LA VARIABLE PROTEIN CON CONJUNTOS SINGLETON.**

En este caso la defusificación se obtiene con la fórmula:

$$PROTEIN = \frac{\sum mi * acción}{\sum mi}$$

Donde  $i$  es el número de etiquetas activadas, en el caso 4,  $mi$  es cada uno de los valores de las funciones de membresía obtenidas y acción es el valor de PROTEIN para la etiqueta resultante, llegándose entonces a lo siguiente:

$$PROTEIN = \frac{0.6 * 105 + 0.3 * 95 + 0.4 * 95 + 0.3 * 95}{0.6 + 0.3 + 0.4 + 0.3} = \frac{158}{1.6} = 98.75 \text{ gm / día}$$

Que es la cantidad de proteínas diarias que debe suministrarse al adolescente.

#### **2.2.4 FORMANDO EL CONJUNTO SOLUCIÓN DE SALIDA.**

Para formar este conjunto, se procesa de la siguiente manera:  
Se forma un nuevo conjunto llamado conjunto solución en donde en el eje horizontal va el universo de la variable de control, en el eje vertical naturalmente la función de membresía, pero inicialmente es un conjunto vacío.

En él se pegan los conjuntos correspondientes a las etiquetas activadas, pero cortadas a la altura de la función de membresía resultante; así por ejemplo, con la regla N.-12, se activa PRMA con un corte a una altura de 0.6. Una vez realizado todo esto, se forma la unión de las figuras obtenidas, con el máximo, o se puede aplicar también la suma de las funciones de membresía; en este caso se ha optado por la unión, resultando la parte sombreada de la Figura.... Que sería la agregación

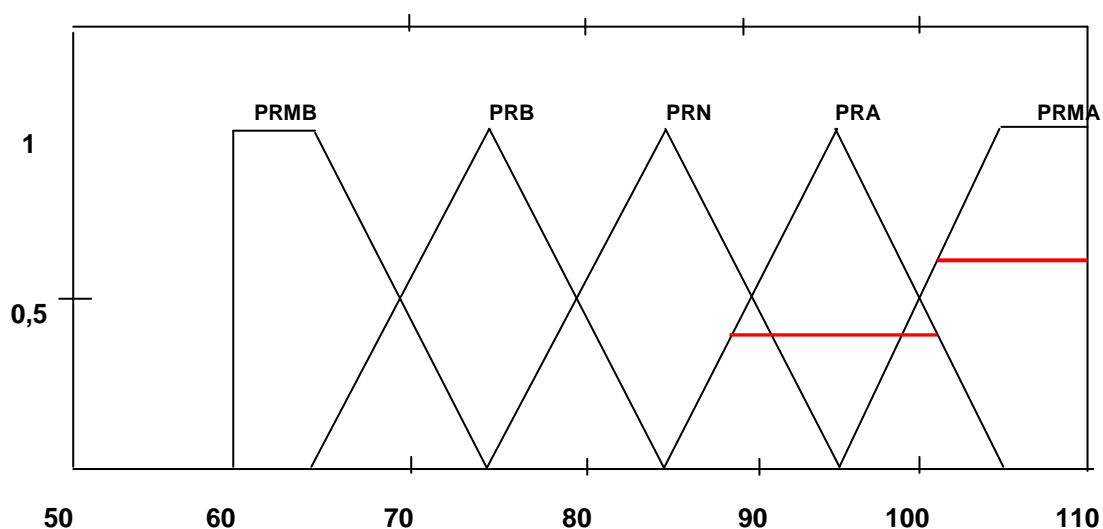


Figura 2.8 CONJUNTO DIFUSO SOLUCIÓN

Se calcula el centro de gravedad de la parte sombreada, al que se le llama Centroide, fórmula debida fundamentalmente a Mamdani. En este caso, el valor obtenido, es la cantidad de proteínas diarias que deben darse al adolescente, es decir:

$$99.6 \text{ gm./día}$$

Con esto queda completo este primer ejemplo, para lo cual se ha empleado la herramienta de diseño y análisis.

### 2.3 IDENTIFICACIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA PARA EL PROTOTIPO.

En esta sección debido a las necesidades del prototipo a construir se han identificado algunas variables que podemos mencionar como son la TEMPERATURA seteada por un usuario, como la temperatura de entrada al equipo, LA POTENCIA, y el FLUJO del agua.

### 2.3.1. TEMPERATURA

En vista que la temperatura del agua es la variable principal a controlar se tomara las siguientes consideraciones.

Se sabe que la temperatura natural del cuerpo humano es de 37 °C por lo que una temperatura adecuada del agua para un usuario se encontraría en un rango entre xx °C y yy °C. Este rango de temperatura se ha tomado después de haber realizado un conjunto de pruebas las mismas que consistieron en tomar la temperatura del agua mediante el uso de un termómetro mientras que se variaba la misma.

A continuación se muestra una tabla con los resultados obtenidos de las pruebas mencionadas anteriormente.

<b>Temperatura</b>	<b>Estado</b>
17	Ambiente
20	Ambiente
22	Ambiente
24	Ambiente
26	Tibio
28	Tibio
30	Tibio
32	Tibio
34	Tibio
36	Tibio
38	Tibio
40	Tibio
42	Tibio
44	Caliente
46	Caliente
48	Caliente
50	Caliente

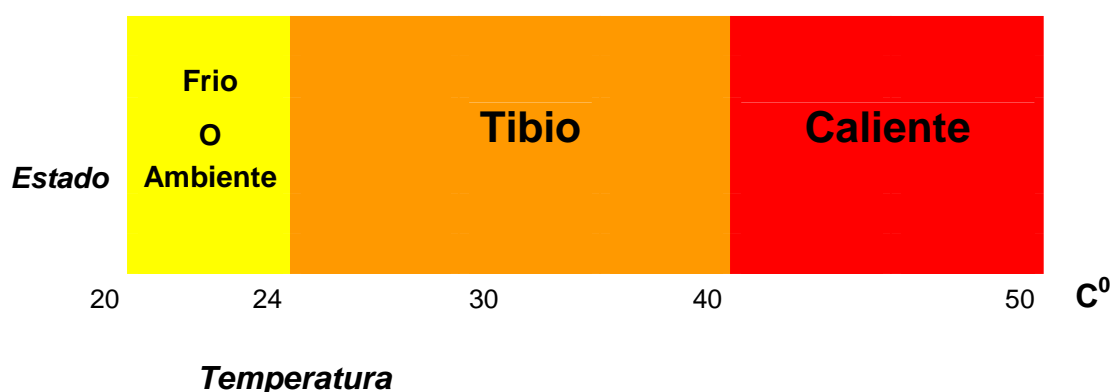
52	Caliente
54	Caliente
56	Caliente
58	Caliente
60	Caliente
62	Caliente
64	Caliente
66	Caliente
68	Caliente y tiende a hervir
70	Caliente y tiende a hervir

**Tabla 2.2 RANGO DE TEMPERATURAS Y SU ESTADO.**

Con los datos de la tabla anterior, se ha logrado establecer un rango de temperaturas que en términos comunes se han clasificado como:

- Agua fría o ambiente.
- Agua tibia.
- Agua caliente.

Dichos rangos de temperatura del agua se aprecian mejor en la Fig. 2.9 que se encuentra a continuación.



**Figura 2.9 RANGOS DE TEMPERATURA**

Como también hay que mencionar el material del elemento resistivo que se vaya a utilizar para suministrarle la potencia suficiente para alcanzar la

temperatura deseada y así obtener el mejor rendimiento y llegar al objetivo planteado de ahorro en la energía eléctrica y agua.

Los efectos de la temperatura sobre el valor de resistencia varían cuando la temperatura cambia.

Este nuevo valor de resistencia a una nueva temperatura, conociendo el valor de la resistencia a una temperatura dada se obtiene utilizando la siguiente fórmula:

$$R_{tf} = R_{to} \times [1 + \alpha (t_f - t_o)]$$

Donde:

- $R_{tf}$  = resistencia final a la temperatura  $t_f$ , en ohmios
- $R_{to}$  = resistencia inicial a la temperatura  $t_o$ , en ohmios
- $\alpha$  = coeficiente de temperatura (ver la tabla siguiente)
- $t_f$  = temperatura final en  $^{\circ}\text{C}$
- $t_o$  = temperatura inicial en  $^{\circ}\text{C}$

Tabla de coeficientes de variación de resistencia por grado de temperatura.

Material	$\alpha$	Material	$\alpha$
Aluminio	0.0039	Plata	0.0038
Manganita	nulo	Estaño	0.0042
Bronce fosforoso	0.002	Plomo	0.0037
Nicromio	0.00013	Kruppina	0.0007
Carbón	0.0005	Tungsteno	0.0041
Níquel	0.0047	Latón	0.002
Niquelina	0.0002	Wolframio	0.0045
Cobre	0.00382	Oro	0.0034

**Tabla 2.2 COEFICIENTES DE VARIACION DE RESISTENCIA.**

**Ejemplo:**



Se tiene un conductor de cobre con resistencia = 20 ohmios a 10°C.. Cuál será el nuevo valor de la resistencia, si la temperatura sube a 70°C. ?

Aplicando la fórmula  $R_{tf} = R_{to} \times [1 + \alpha (tf - to)]$  con los siguientes valores:

- $R_{to} = 20$  ohmios
- $\alpha = 0.00382$  (cobre)
- $tf = 70^\circ\text{C}$
- $to = 10^\circ\text{C}$

Se obtiene:  $R_{tf} = 20 [1 + 0.00382 (70 - 10)] = 24.584$  ohmios.

### **2.3.2 FLUJO DEL AGUA.**

Esta variable en el diseño se la tomara como constante siempre a la entrada y a la salida.

En conclusión el análisis de las variables a usarse con la lógica difusa para el diseño del prototipo de control de temperatura se lo detallara en la sección correspondiente al diseño.

## **CAPITULO 3**

## **DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DE CONTROL DE TEMPERATURA.**

En este capítulo se detallara el diseño de el prototipo que se implementará utilizando un microcontrolador de la familia MSC-51 y el termómetro DS1820, describiendo los elementos a utilizarse y especificando los parámetros de operación de esta aplicación así como también se realizara el estudio del costo del mismo tanto como los precios de los elementos, lo cual permitirá calcular el valor para su producción unitaria.

### **3.1 DISEÑO DEL PROTOTIPO.**

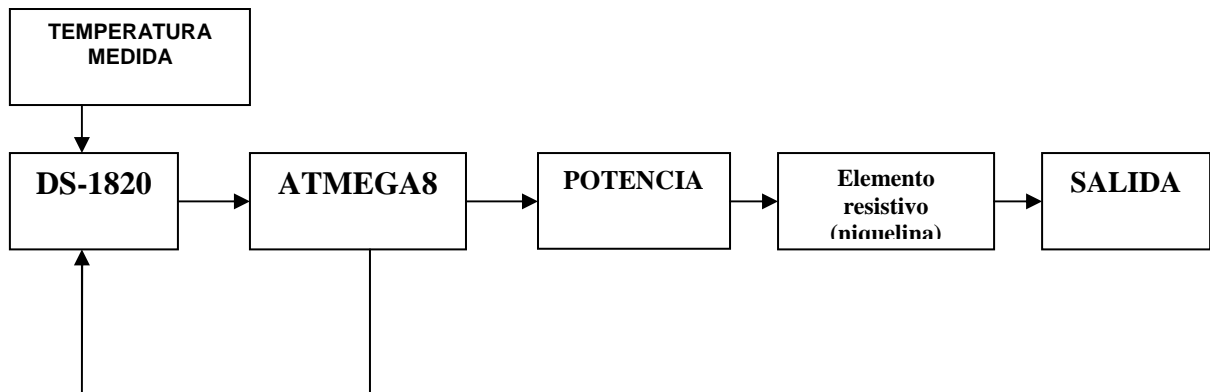
En el diseño de este prototipo de controlador electrónico de temperatura se ha tomado varias consideraciones para poder alcanzar este objetivo, en el capítulo dos se realizo un análisis de las variables para el control difuso del mismo, y se ira mencionando a continuación.

### **3.2 PARÁMETROS DE DISEÑO**

#### **3.2.1 CONTROL DE LA TEMPERATURA.**

Básicamente para llegar al objetivo de controlar la temperatura del agua mediante el diseño del calentador electrónico, el controlador difuso debe entregar la potencia necesaria al elemento resistivo, es importante definir el como se requiere que funcione el prototipo, como se va a realizar, que tipo de lenguaje se va a utilizar, el interfaz a utilizarse, como alcanzar la temperatura seteada por el usuario en el menor tiempo posible para el aprovechamiento óptimo de la energía eléctrica al calentar la niquelina del calentador electrónico y en el tiempo que no se encuentre funcionando, como se van almacenar los datos para analizarlos de la mejor manera posible, así también que la presentación de los datos a setear por el usuario sea la mas adecuada y agradable para el mismo,

A continuación en la figura 3.1 se presenta un diagrama de bloques indicando de manera general y sencilla el funcionamiento del prototipo.



**Figura 3.1 DIAGRAMA DE BLOQUES.**

Para la implementación del control de temperatura de este calentador electrónico por lógica difusa, el procedimiento viene desde que un usuario ingresa a su gusto una temperatura, en donde el agua que entre se va a medir su temperatura en utilizando el sensor inteligente DS1820, el que envía datos al microcontrolador ATMEGA8, además que maneja el tiempo entre las muestras de temperatura, a la salida se calculara el un ángulo de disparo que modificara la potencia para el elemento resistivo (niquelina) para así obtener la temperatura seteada por un usuario y así también alcanzar lo que es el ahorro de la energía eléctrica y el agua potable.

En el anterior capítulo se mostró un una tabla con rangos de temperatura en términos comunes, ahora se mencionan que rangos de temperatura del agua para el cuerpo humano, en donde el mismo reacciona a un cambio de temperatura después de 2 grados centígrados.

Temperatura	Estado
38	Tibio

39	<b>Tibio</b>
40	<b>Tibio</b>
41	<b>Caliente</b>
42	<b>Caliente</b>
43	<b>Caliente</b>
44	<b>Caliente</b>
45	<b>Caliente</b>
46	<b>Muy Caliente</b>
47	<b>Muy Caliente</b>

**Tabla 3.1 DIAGRAMA DE BLOQUES.**

### **3.2.2 SELECCIÓN DE PARTES Y PIEZAS.**

En esta sección se explicara el porque se escogieron los dispositivos utilizados para la construcción del hardware.

#### **3.2.2.1 EL ATMEGA8**

Este microcontrolador es de baja potencia CMOS de 8-bit, basado en la arquitectura AVR RISC, ejecuta 130 instrucciones de gran alcance en un solo ciclo de reloj.

El ATmega8 alcanza rendimientos de procesamiento que se acerca a 1 MIPS por el megaciclo, permitiendo al sistema optimizar el consumo de energía contra el proceso de velocidad.

La base del AVR combina un sistema de instrucción rico con 32 registros de fines generales del funcionamiento. Todos los 32 registros están conectados directamente con la unidad de la lógica aritmética (ALU), permitiendo que dos registros independientes sean alcanzados en una sola instrucción ejecutada en un ciclo de reloj. La arquitectura es más eficiente por que alcanza rendimientos de procesamiento hasta diez microcontroladores que convencionales.

El ATmega8 proporciona las características siguientes:

Tiene las capacidades de 8 kbytes de un sistema interno flash programable con la capacidad leer mientras escribe, 512 octetos de EEPROM, 1Kbyte de SRAM, 23 líneas de entrada - salida para propósitos generales, tres flexibles timer y contadores con modos comparativos, interrupciones internas y externas, un USART programable serial, un byte orientad a una interfaz en serie de dos hilos, un ADC de 6 canales(8 canales en paquetes de TQFP y de MLF) donde 4 de 6 canales tienen una exactitud de 10 bits y 2 canales tienen una exactitud de 8 bit, un contador de tiempo programable con el oscilador interno, un puerto SPI y cinco modos de software seleccionables de ahorro de energía.

El modo de baja potencia guarda el contenido de los registro pero detiene el oscilador, desactivando todas las otras funciones del chip hasta la próxima interrupción o reinicio del hardware.

En el modo de ahorro de potencia, el contador de tiempo asincrónico continúa funcionando, permitiendo que el usuario mantenga una base del contador de tiempo mientras que está durmiendo el resto del dispositivo. La reducción de ruidos del ADC detiene al CPU y a todos los módulos de entrada-salida excepto al contador de tiempos asincrónicos y el ADC, para reducir al mínimo el ruido de la conmutación durante conversiones del ADC. En el modo espera, el cristal/resonador del oscilador están funcionando mientras que el resto de dispositivos están descansando.

Esto permite un inicio muy rápido combinado con la consumo de baja potencia. El dispositivo es manufacturado con la tecnología de memoria permanente de alta densidad de Atmel.

. El Atmel ATmega8 es un microcontrolador de gran alcance eso proporciona una solución alto-flexible y rentable a muchos usos encajados del control.

Mayor información se encuentra en el anexo 1.

### 3.2.2.2 TIPO DE ENSAMBLADOR

En el propósito de realizar la programación correspondiente al software de nuestro proyecto se ha decidido utilizar como herramienta el compilador BASCOM AVR debido a que al trabajar con instrucciones en lenguaje C la programación resulta mas sencilla y amigable que en los otros compiladores.

### 3.2.2.3 TRIAC

El Triac es un dispositivo semiconductor que pertenece a la familia de los dispositivos de control por tiristores. El triac es en esencia la conexión de dos tiristores en paralelo pero conectados en sentido opuesto y compartiendo la misma compuerta que control. (ver imagen).



Figura 3.2 SÍMBOLO DEL TRIAC.

El triac sólo se utiliza en corriente alterna y al igual que el tiristor, se dispara por la compuerta. Como el triac funciona en corriente alterna, habrá una parte de la onda que será positiva y otra negativa.

La parte positiva de la onda (semiciclo positivo) pasará por el triac siempre y cuando haya habido una señal de disparo en la compuerta, de esta manera la corriente circulará de arriba hacia abajo (pasará por el tiristor que apunta hacia abajo), de igual manera:

La parte negativa de la onda (semiciclo negativo) pasará por el triac siempre y cuando haya habido una señal de disparo en la compuerta, de esta

manera la corriente circulará de abajo hacia arriba (pasará por el tiristor que apunta hacia arriba)

Para ambos semiciclos la señal de disparo se obtiene de la misma patilla (la puerta o compuerta).

Lo interesante es, que se puede controlar el momento de disparo de esta patilla y así, controlar el tiempo que cada tiristor estará en conducción. (recordar que un tiristor solo conduce cuando ha sido disparada (activada) la compuerta y entre sus terminales hay un voltaje positivo de un valor mínimo para cada tiristor)

Entonces, si se controla el tiempo que cada tiristor está en conducción, se puede controlar la corriente que se entrega a una carga y por consiguiente la potencia que consume.

Ejemplo: Una aplicación muy común es el atenuador luminoso de lámparas incandescentes (circuito de control de fase).

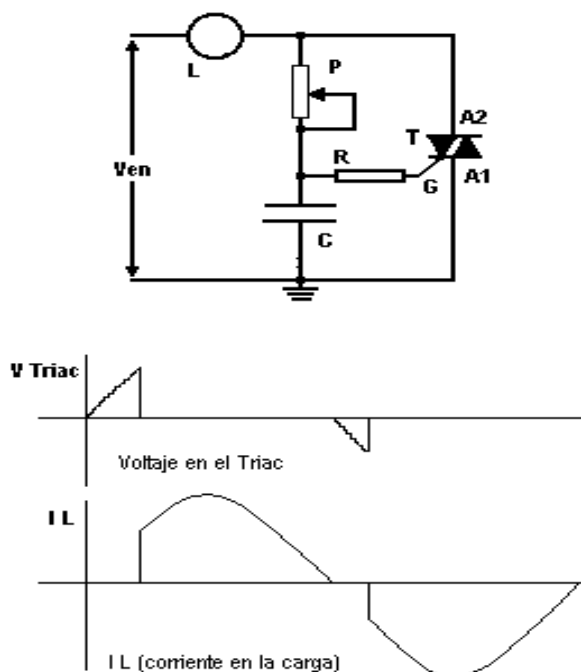


Figura 3.3 CIRCUITO DE UN ATENUADOR LUMINOSO.

Donde:

- Ven: Voltaje aplicado al circuito (A.C.)
- L: lámpara
- P: potenciómetro
- C: condensador (capacitor)
- R: Resistencia
- T: Triac
- A2: Ánodo 2 del Triac
- A3: Ánodo 3 del Triac
- G: Puerta o compuerta del Triac

El triac controla el paso de la corriente alterna a la lámpara (carga), pasando continuamente entre los estados de conducción (cuando la corriente circula por el triac) y el de corte (cuando la corriente no circula)

Si se varía el potenciómetro, se varía el tiempo de carga del condensador causando que se incremente o reduzca la diferencia de fase de la tensión de alimentación y la que se aplica a la compuerta

Nota: la diferencia de fase o la fase entre dos señales u ondas se define como el ángulo (diferencia de tiempo) que existe entre los dos orígenes de las mismas.

Mayores detalles del triac que se empleo se presentan en los Anexo 5.

#### **3.2.2.4 DISPLAY (LCD)**

Pensando en una configuración sencilla y agradable para el usuario se ha elegido una pantalla LCD como implemento para visualizar el menú donde se setearan los parámetros que regirán el control de temperatura de nuestro prototipo, de echo el LCD trabaja con  $2 \times 16 = 32$  caracteres a pesar de que el display sólo puede visualizar 16 caracteres por línea, puede almacenar en total 40 por línea.



Es el usuario el que especifica qué 16 caracteres son los que se van a visualizar.



Figura 3.4 REPRESENTACION DEL LCD.

El LCD dispone de dos tipos de memorias independientes:

- **DD RAM**
- **CG RAM**

DD RAM (Display Data Ram) en esta memoria se almacenan los caracteres que están siendo visualizados o que se encuentran en posiciones no visibles. El display almacena en esta memoria dos líneas de 40 caracteres pero sólo se visualizan 2 líneas de 16 caracteres. Por ello la **DD RAM** tiene un tamaño de  $2 \times 40 = 80$  bytes.

LA CG RAM (Character Generator RAM) es la memoria que contiene los caracteres definibles por el usuario, la memoria está dividida en 8 bloques, correspondiendo cada bloque a un carácter definible por el usuario. Por ello el usuario puede definir como máximo 8 caracteres, cuyos códigos van del 0 al 7. Todas las direcciones están en hexadecimal.

Mayores detalles se presentan en los Anexo 3.

### 3.2.2.5 El DS-1820

La elección de este sensor de temperatura se basó en que no requiere de ajustes (compensación de junta fría y ganancia del amplificador) ni de componentes externos para su utilización, este interfaz 1-wire es el único en su género por que solo requiere un pin para la comunicación.

El DS1820 tiene un único código serial de 64 bits, el cual permite funcionar a múltiples DS1820 con el mismo bus, el cual soporta un rango de temperaturas que van desde  $-55^{\circ}\text{C}$  hasta una temperatura máxima de  $125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  hasta  $+257^{\circ}\text{F}$ ) y tiene una exactitud de  $\pm 0.5^{\circ}\text{C}$  en el rango de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ .

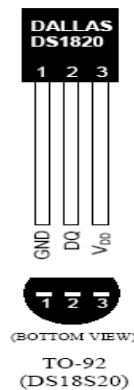


Figura 3.5 REPRESENTACION DE UN DS1820 DE FORMA DE TRANSISTOR.

Otra característica de este termómetro digital es la de que convierte la temperatura en 750 ms como máximo, provee una medida de temperatura en centígrados de 9 bits.

Entre sus aplicaciones se puede nombrar control termostático, sistemas industriales, productos de consumo, termómetros o en cualquier sistema sensible térmico.

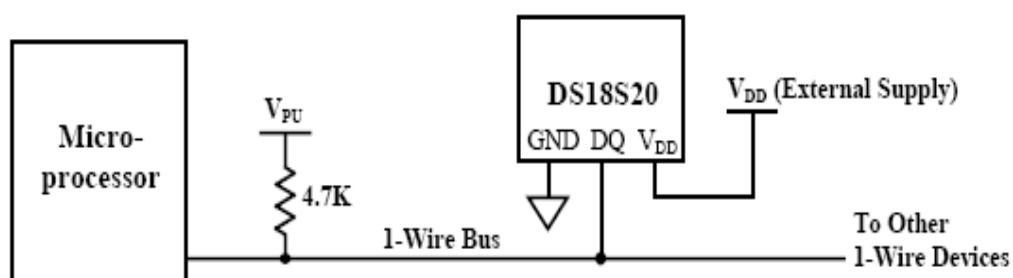


Figura 3.6 DIAGRAMA DE CONEXIÓN DE UN DS1820.

Mayores detalles se presentan en los Anexo 4.

### 3.2.2.6 EL OPTOTRIAC. (MOC 3020)

Este dispositivo es un conductor que funciona como un triac, esta diseñado para ser la interfase entre los controles electrónicos y triacs de poder y así controlar las cargas de resistividad e inductancia para operaciones de voltaje alterno de 115 a 240 voltios.

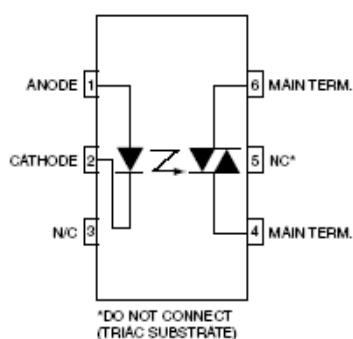


Figura 3.7 DIAGRAMA DE PINES DEL OPTOTRIAC.

## APLICACIONES

- En controles Industriales
- Válvulas de control solenoides
- Luces de tráfico
- switch estáticos de poder de AC
- Máquinas expendedoras.
- Lámparas incandescentes control de intensidad
- Relay de estado sólido
- Control para motores.

Mayores detalles se presenta en los Anexo 2.

### 3.2.2.7 EL TANQUE.

Las dimensiones del tanque utilizado se encuentran relacionadas directamente a la potencia el elemento resistivo a utilizar; a continuación se muestra un diseño realizado en Autocad que se muestra en el Anexo 6.

### 3.2.2.8 EL ELEMENTO RESISTIVO.

A este elemento es cual se le entregara la potencia que sea necesaria para alcanzar la temperatura seteada por el usuario para esto se tomara en cuenta una potencia de 2000 watios y a la entrada 110 voltios, donde:

$$V = I * R \quad (3.1)$$

$$P = V * I \quad (3.2)$$

Donde despejando y reemplazando;

$$R = \frac{V^2}{P} = \frac{(110v)^2}{2000w} = 6.05\Omega \quad (3.3)$$

Entonces como se puede ver el elemento resistivo debe de ser de por lo menos 6.05  $\Omega$  para alcanzar el objetivo que se desea.

### 3.2.2.9 FUENTE REGULADA.

Para el correcto funcionamiento de los diferentes dispositivos electrónicos que se están empleando en este diseño como son por ejemplo el microcontrolador ATEMEGA8 y el LCD que requieren un voltaje regulado de 5 V. para lo cual se procedió a diseñar una fuente regulada, que se muestra a continuación.

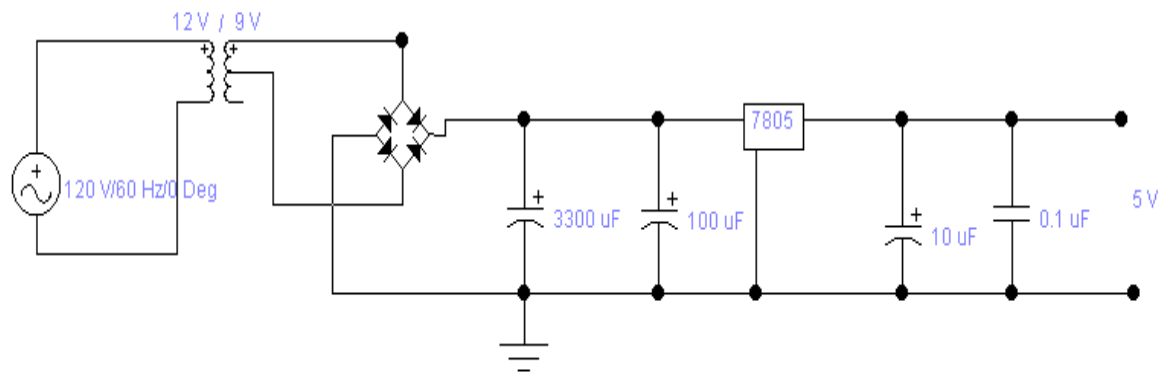


Figura 3.8 ESQUEMATICO DE LA FUNTE REGULADA.

### 3.3 IDENTIFICACIÓN DE LAS VARIABLES DE ENTRADA Y SALIDA DEL CONTROLADOR ELECTRONICO DE TEMPERATURA DIFUSO.

En el capítulo anterior se menciona que para el diseño del prototipo se debe pasar por algunos pasos, la identificación de las variables lingüísticas, asignar funciones de membresía a cada una (fusificación) y se realiza la base de reglas difusas del controlador, a continuación se detallara lo antes mencionado.

En este proceso de identificación de las variables se tendrá:

La temperatura seteada (referencia) ingresada por el usuario que esta en un rango o un universo de 20 a 60 grados centígrados que va a estar conformada por 5 funciones de membresía, así como el error, también conformado por 5 funciones de membresía.

Donde el error esta dado por:  $Error = T_{referencia} - T_{DS1820}$

Las etiquetas escogidas son las siguientes:

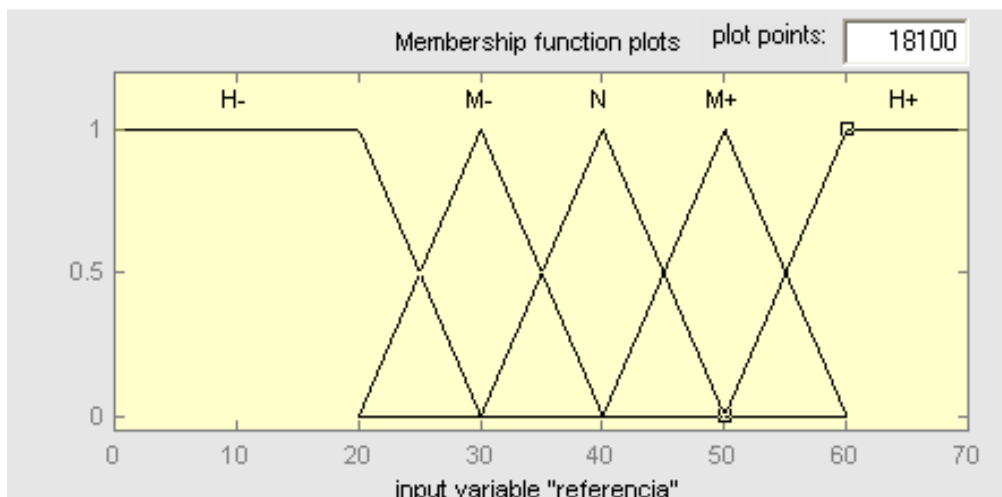
**H -**: high menos

**M -**: médium menos

**N**: normal

**M +**: médium más

**H +**: high más

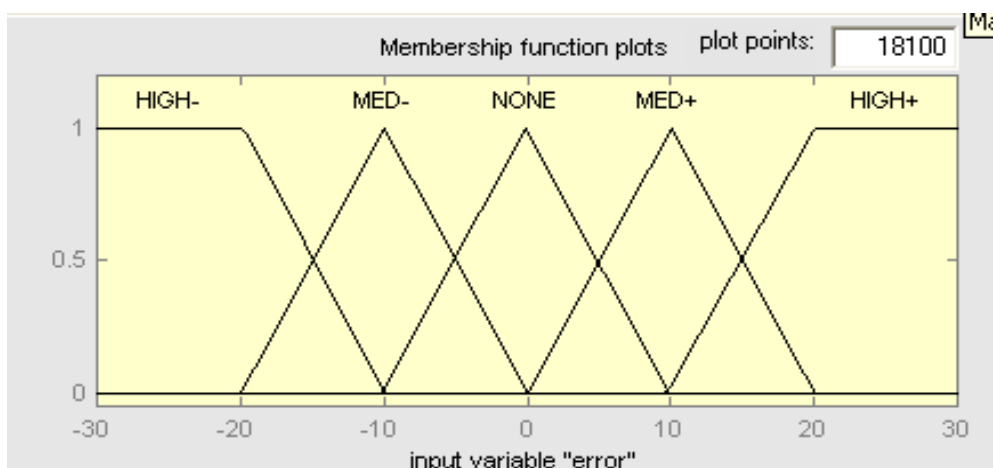


**Figura 3.9 FUSIFICACIÓN DE LA VARIABLE TEMPERATURA DE REFERENCIA**

Cabe mencionar que se ha realizado un artificio para facilitar el diseño, solo en el rango de grado de pertenecía que iría desde 0 a 100 (eje Y), el mismo artificio se lo realizó en la variable lingüística Error, pero aquí el incremento se lo realizó al grado de pertenecía y al universo de la variable Error (ejes X,Y).

Entonces esta variable tendrá un universo de la función desde -20 hasta 20 grados centígrados y el grado de pertenecía va de 0 a 100 como se puede apreciar en la figura siguiente

Por medio de este artificio matemático las pendientes de las rectas de las dos variables tendrán el mismo valor  $m= 10$ , para sacar las ecuaciones de las rectas ( $Y=mX + b$ ) de las respectivas variables lingüísticas.



**Figura 3.10 FUSIFICACIÓN DE LA VARIABLE ERROR**

### 3.3.1 CALCULO DEL GRADO DE PERTENECÍA DE LAS VARIABLES

Como ya se vio anteriormente las variables que se escogieron para la realización del control temperatura, fueron el la temperatura de referencia y el error, ahora se explicara como se realizo el calculo de los valores de cada una de las funciones de pertenencia que se etiquetan mediante una función. Empezando por la variable Error (  $E_{fuzzy}(i)$  ), donde para ambos casos se tomara:

**H -**:  $E_{fuzzy}(1)$

**M -**:  $E_{fuzzy}(2)$

**N**:  $E_{fuzzy}(3)$

**M +**:  $E_{fuzzy}(4)$

**H +**:  $E_{fuzzy}(5)$

Las pendientes para cada una de las rectas serán:  $Em_{neg} = -10$ ,  
 $Em_{pos} = 10$ .

Se comenzara con el intervalo de valores menores a -20:

$E_{fuzzy}(1) = 100$

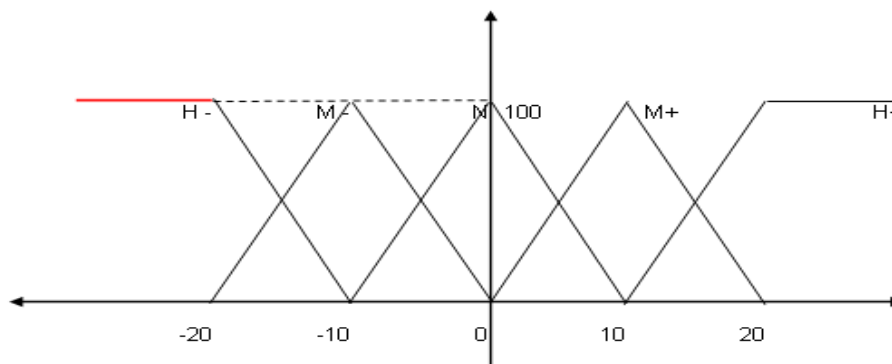


Gráfico 3.1 RECTAS EN EL INTERVALO MENOR A -20

En el intervalo de -20 a -10

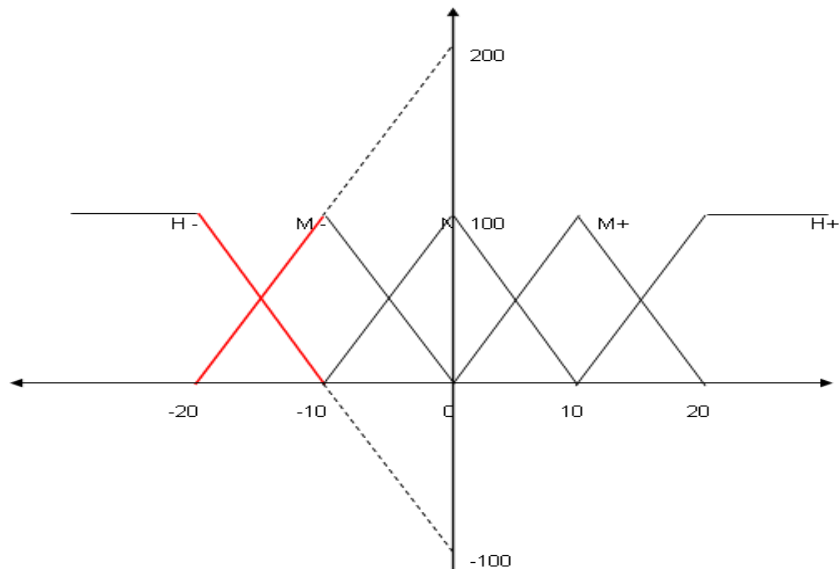
$$y = mX + b$$

$$E_{fuzzy}(1) = Em_{neg} + b$$

$$E_{fuzzy}(1) = -10X - 100$$

$$E_{fuzzy}(2) = Em_{pos} + b$$

$$E_{fuzzy}(2) = 10X + 200$$



**Gráfico 3.2 RECTAS EN EL INTERVALO -20 A -10**

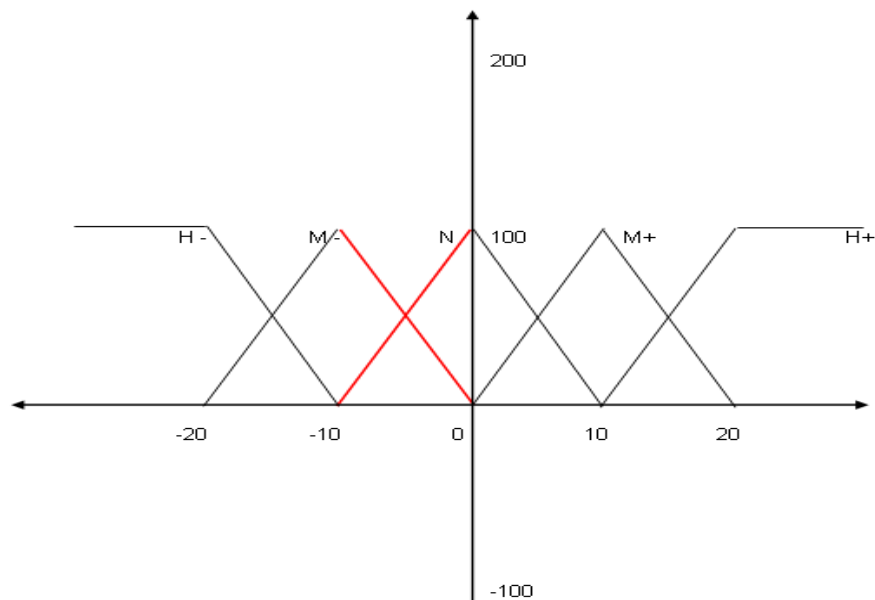
En el intervalo de -10 a 0

$$E_{\text{fuzzy}}(2) = E_{m\_neg}$$

$$E_{\text{fuzzy}}(2) = -10X$$

$$E_{\text{fuzzy}}(3) = E_{m\_pos} + b$$

$$E_{\text{fuzzy}}(3) = 10X + 100$$



**Gráfico 3.3 RECTAS EN EL INTERVALO -10 A 0**

En el intervalo de 0 a 10

$$E_{\text{fuzzy}}(3) = E_{m\_neg} + b.$$

$$E_{\text{fuzzy}}(3) = -10X + 100$$



$$E_{\text{fuzzy}}(4) = E_{m\_pos}$$

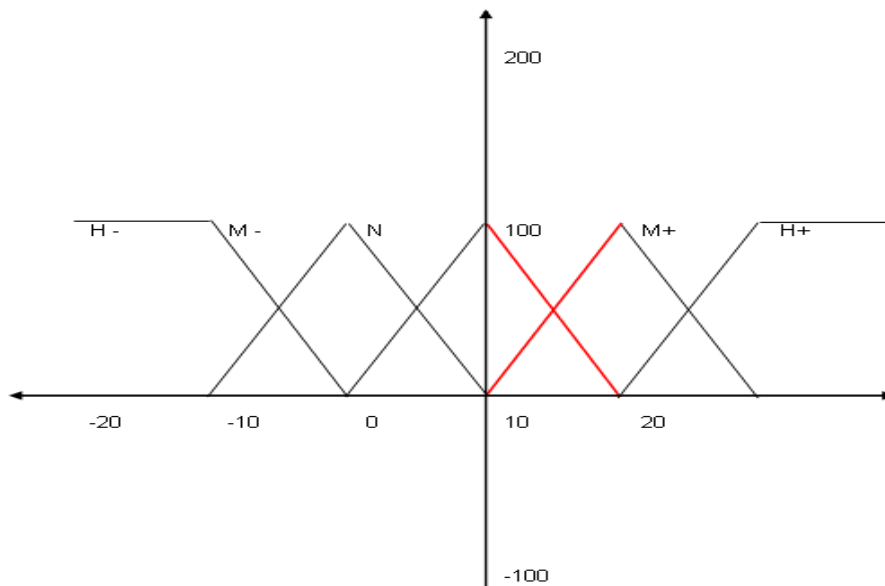


Gráfico 3.4 RECTAS EN EL INTERVALO 0 A 10

En el intervalo de 10 a 20

$$E_{\text{fuzzy}}(4) = E_{m\_neg} + b.$$

$$E_{\text{fuzzy}}(4) = -10X + 200$$

$$E_{\text{fuzzy}}(5) = E_{m\_pos} + b$$

$$E_{\text{fuzzy}}(5) = 10X - 100$$

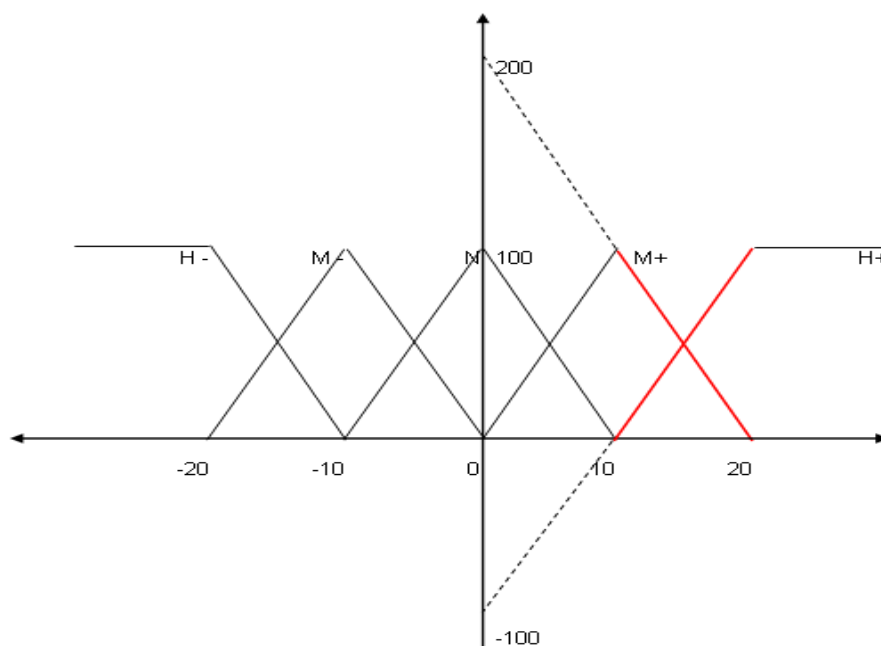


Gráfico 3.5 RECTAS EN EL INTERVALO 10 A 20

En el intervalo de mayores a 20

$$E\_fuzzy(5) = 100$$

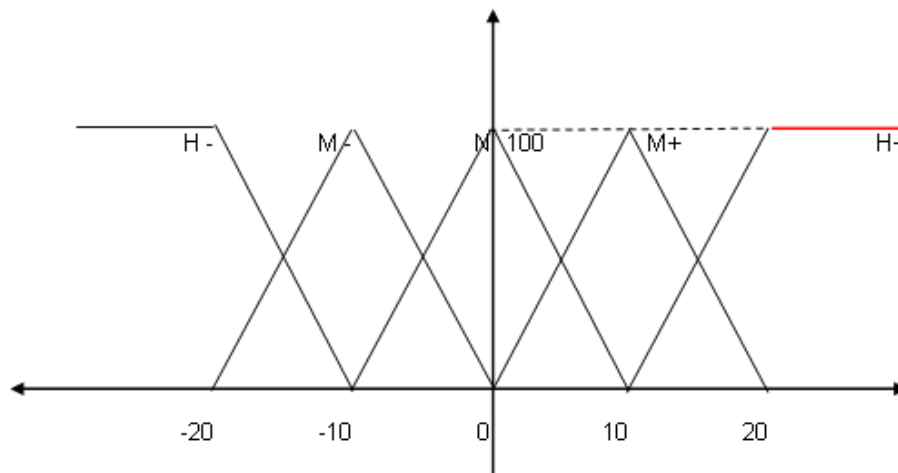


Gráfico 3.6 RECTAS EN EL INTERVALO MAYOR A 20

EL mismo procedimiento se realizara para la otra variable temperatura de referencia ( R\_fuzzy (i) ) motivo por el cual no se explicara nuevamente el proceso de construcción de cada una de las ecuaciones para esta nueva variable..

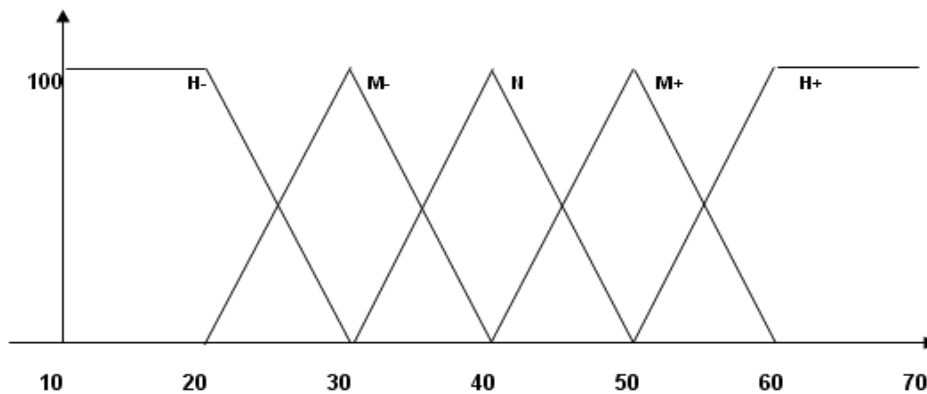


Gráfico 3.2 VARIABLE TEMPERATURA DE REFERENCIA

Donde las pendientes son:  $Rm\_neg = -10$  y  $Rm\_pos = 10$

Se comenzara con el intervalo de valores menores a 20:

$$R\_fuzzy(1) = 100$$

En el intervalo de 20 a 30

$$y = mX + b$$

$$R\_fuzzy (1) = Rm\_neg + b$$

$$R\_fuzzy (1) = -10X + 300$$

$$R\_fuzzy (2) = Rm\_pos + b$$

$$R\_fuzzy (2) = 10X - 200$$

En el intervalo de 30 a 40

$$R\_fuzzy (2) = Rm\_neg + b$$

$$R\_fuzzy (2) = -10X + 400$$

$$R\_fuzzy (3) = Rm\_pos + b$$

$$R\_fuzzy (3) = 10X - 300$$

En el intervalo de 40 a 50

$$R\_fuzzy (3) = Rm\_neg + b.$$

$$R\_fuzzy (3) = -10X + 500$$

$$R\_fuzzy (4) = Rm\_pos + b$$

$$R\_fuzzy (4) = 10X - 400$$

En el intervalo de 50 a 60

$$R\_fuzzy (4) = Rm\_neg + b.$$

$$R\_fuzzy (4) = -10X + 600$$

$$R\_fuzzy (5) = Rm\_pos + b.$$

$$R\_fuzzy (5) = 10X - 500$$

En el intervalo de mayores a 60

$$R\_fuzzy (5) = 100$$

Todo este proceso ayudara como ya mencionamos a calcular los valores de grado de pertenencia en las respectivas variables para el control, donde el valor escogido será el valor más pequeño.

Continuando con este proceso de diseño para el sistema de control difuso corresponde la elaboración de la base de reglas difusas, en otros autores las llaman también “Memoria Asociativa Difusa (FAM)”, donde en el capítulo anterior

ya se menciona y se explica su elaboración paso a paso donde define la relación de las variables medidas o las de control.

En este caso serán 25 Las reglas difusas resultantes activan 7 “singleton” denominados:

**MMB:** MUY MUY BAJO

**MB:** MUY BAJO

**B:** BAJO

**M:** MEDIO

**A:** ALTO

**MA:** MUY ALTO

**MMA:** MUY MUY ALTO

### 3.3.2 CREACION DE LAS REGLAS DIFUSAS.

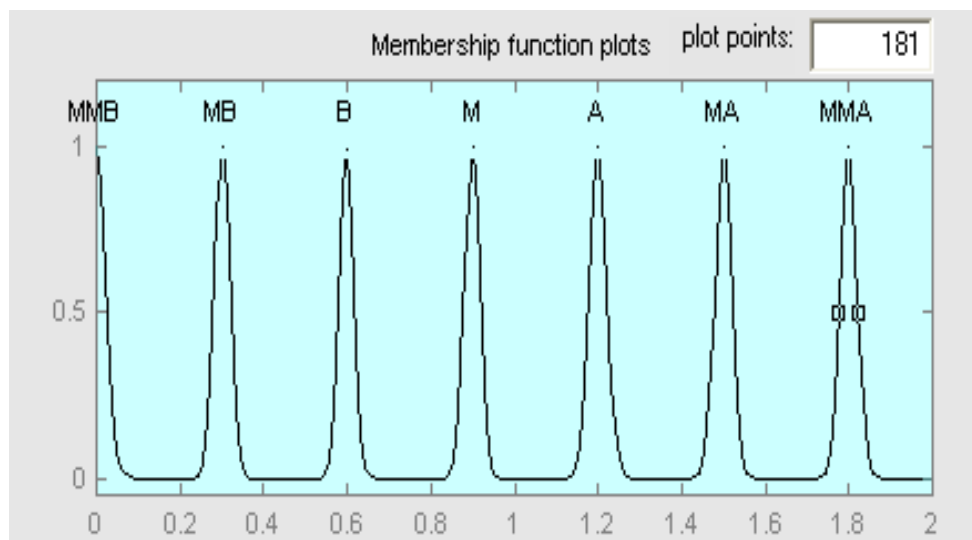
A continuación se ha definido la Base de Reglas del controlador difuso (Inferencia), que indican la relación de las variables medidas o las de control.

Los resultados para las variables de control utilizadas se muestran a continuación:

<b>ERROR</b> <b>REFERENCIA</b>	<b>H-</b>	<b>M-</b>	<b>N</b>	<b>M+</b>	<b>H+</b>
<b>HIG-</b>	1. MMB	6. MB	11. MB	16. B	21 MMA
<b>MEDIUM-</b>	2. MMB	7. MB	12. B	17. M	22. MMA
<b>N</b>	3. MMB	8. B	13. M	18. A	23. MMA
<b>MEDIUM+</b>	4. MMB	9. PRN	14. A	19. MA	45. MMA
<b>H+</b>	5. MMB	10. A	15. PRN	20. MMA	25. MMA

Tabla 3.2- Reglas Difusas del Controlador Difuso

Empleando conjuntos Singleton la variable de salida será de siguiente forma:



**Figura 3.11 FUNCION DE MEMBRANCIA DE LA SALIDA**

Para controlar la cantidad de potencia que se enviara al elemento resistivo (Niquelina) se probó con dos alternativas que se describirán a continuación:

- a) Control por modulación de ancho de pulso PWM, el mismo que será controlado mediante software generado en el Microcontrolador ATMEGA8, y que por los motivos que a continuación se detallarán no se escogió como una solución para el diseño del prototipo y a su vez para el cumplimiento de los objetivos planteados por el mismo.
  - El retardo que se presenta entre pulsos de onda modulada genera un corte notorio de voltaje entregado al elemento resistivo, mismo corte que puede llegar a producir una ruptura o a su vez el elemento resistivo podría llegar a quemarse.
  - Al utilizar este método se estaría realizando un control del voltaje entregado al elemento resistivo, cuando uno de los objetivos planteados en este proyecto es realizarlo controlando la intensidad de corriente.

- Al producir un apagado total y un encendido seguido de este, estaría sacrificandose La vida util del elemento resistivo y por consiguiente se vería comprometida la eficiencia del prototipo, llegando de esta manera a no cumplir con el objetivo planteado.

b) Control mediante cruce por cero con onda rectificada, que en nuestro caso será la solución elegida por que cumple con los requerimientos del proyecto.

La funcionalidad radica en la forma de entregar la cantidad de corriente necesaria al elemento resistivo, tomando en cuenta la lectura de temperatura y la temperatura seteada por el usuario.

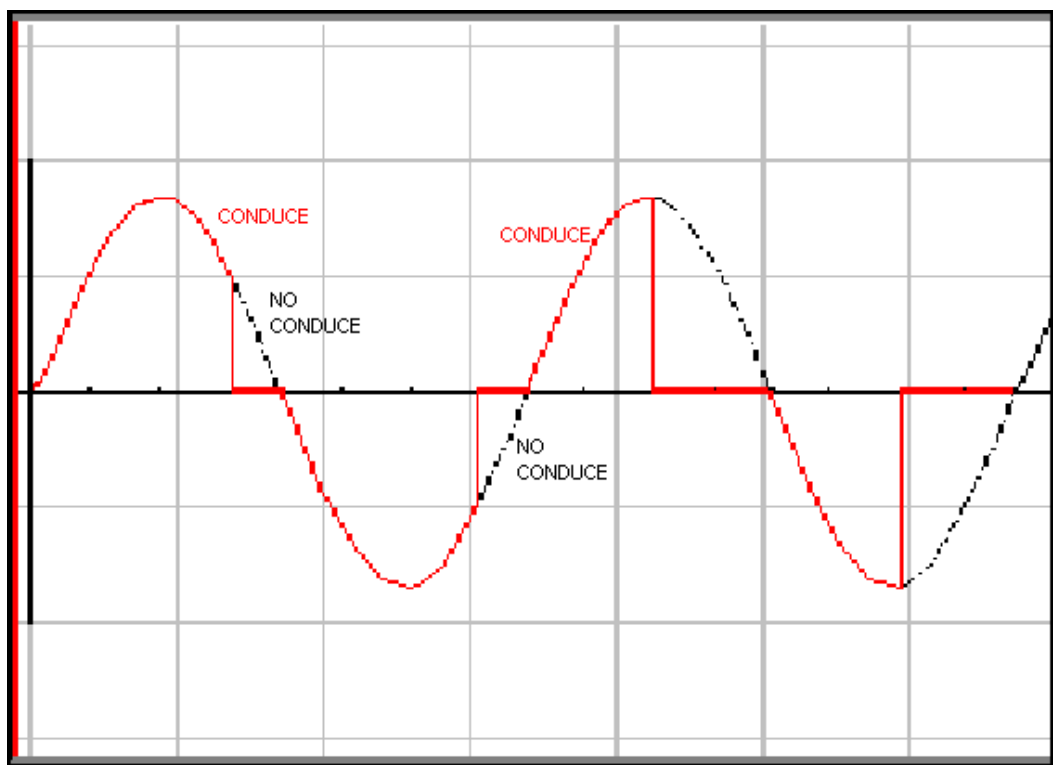


Figura 3.12 INTENSIDAD DE CORRIENTE ENTREGADA AL ELEMENTO RESISTIVO

En la figura anterior se puede apreciar una explicación grafica generalizada de la corriente entregada al elemento resistivo (Niquelina).

Cabe mencionar que para realizar el cruce por cero primero deberá generar los pulsos, que entrara al microcontrolador lo cual se logro de siguiente manera en donde se escogí el flanco negativo:

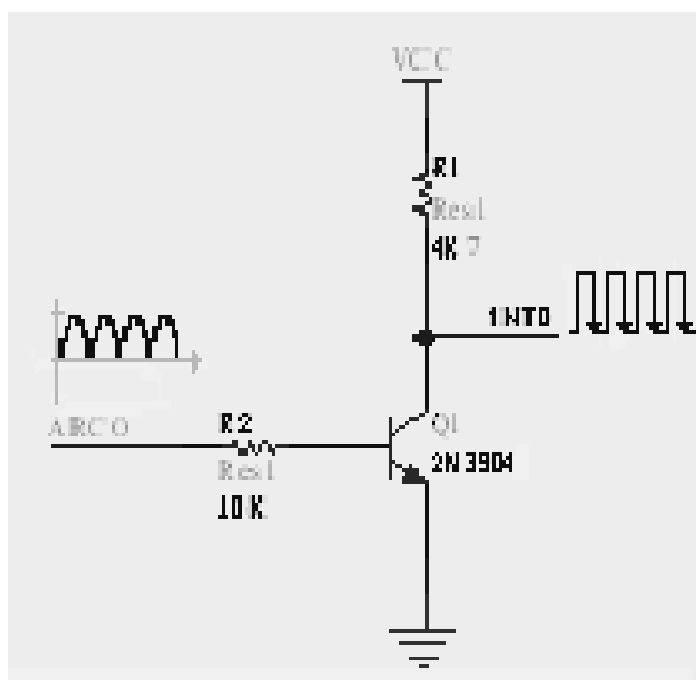


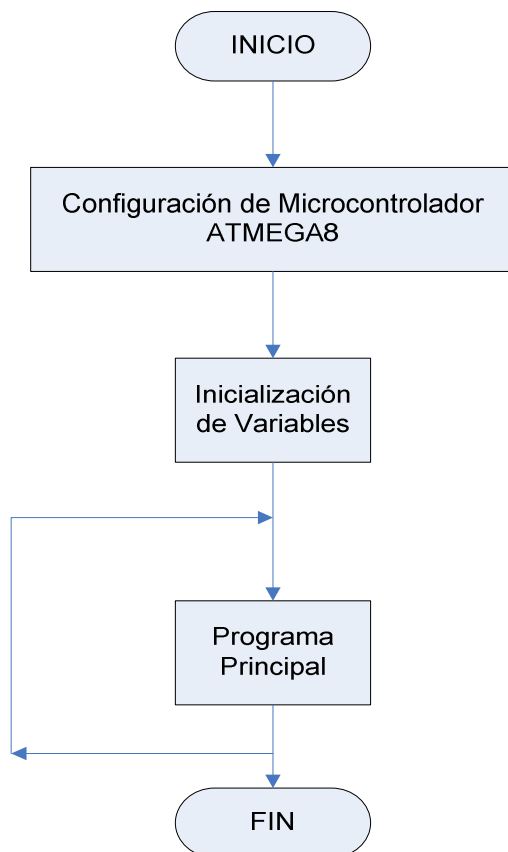
Figura 3.13

En microcontrolador se realizaran los cálculos correspondientes como ya se explico anteriormente.

Cabe decir que no se utilizo lo que SCR por la razón en que la onda enviada a la niquelina por el triac solo dispara utilizando el hemiciclo positivo mas n en el hemiciclo negativo no funciona y no dispara.

### 3.4 DIAGRAMAS DE FLUJO.

### 3.4.1 FLUJOGRAMA GENERAL

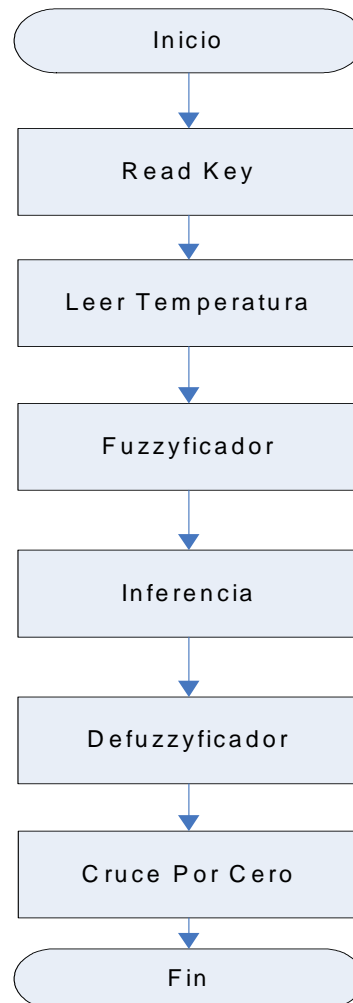


**Diagrama 3.1 Programa General**

Como se puede ver la estructura general del proyecto comienza por configurar el Microcontrolador ATMEGA8, luego de determinar y setear la configuración se procede a inicializar las variables a utilizar en el programa, entonces se da inicio al programa principal el mismo que toma los datos y los evalúa cada cierto tiempo, luego de esto finaliza si así lo requiere el usuario.



### 3.4.2 PROGRAMA PRINCIPAL.



**Diagrama 3.2 Programa Principal**

Como se puede apreciar en el flujograma anterior el programa comienza al leer las teclas de configuración del prototipo, luego de setear la temperatura referencial y mostrarla en la pantalla al igual que la temperatura del ambiente que en este caso corresponde a la temperatura del agua, se prosigue a iniciar la fusificación de la variable de entrada, se aplica después las reglas de inferencia y se defusifica la variable de salida, al final se actualiza el dato para realizar una nueva lectura de temperatura y repetir el proceso.

### 3.4.3 SUBROUTINA READKEY

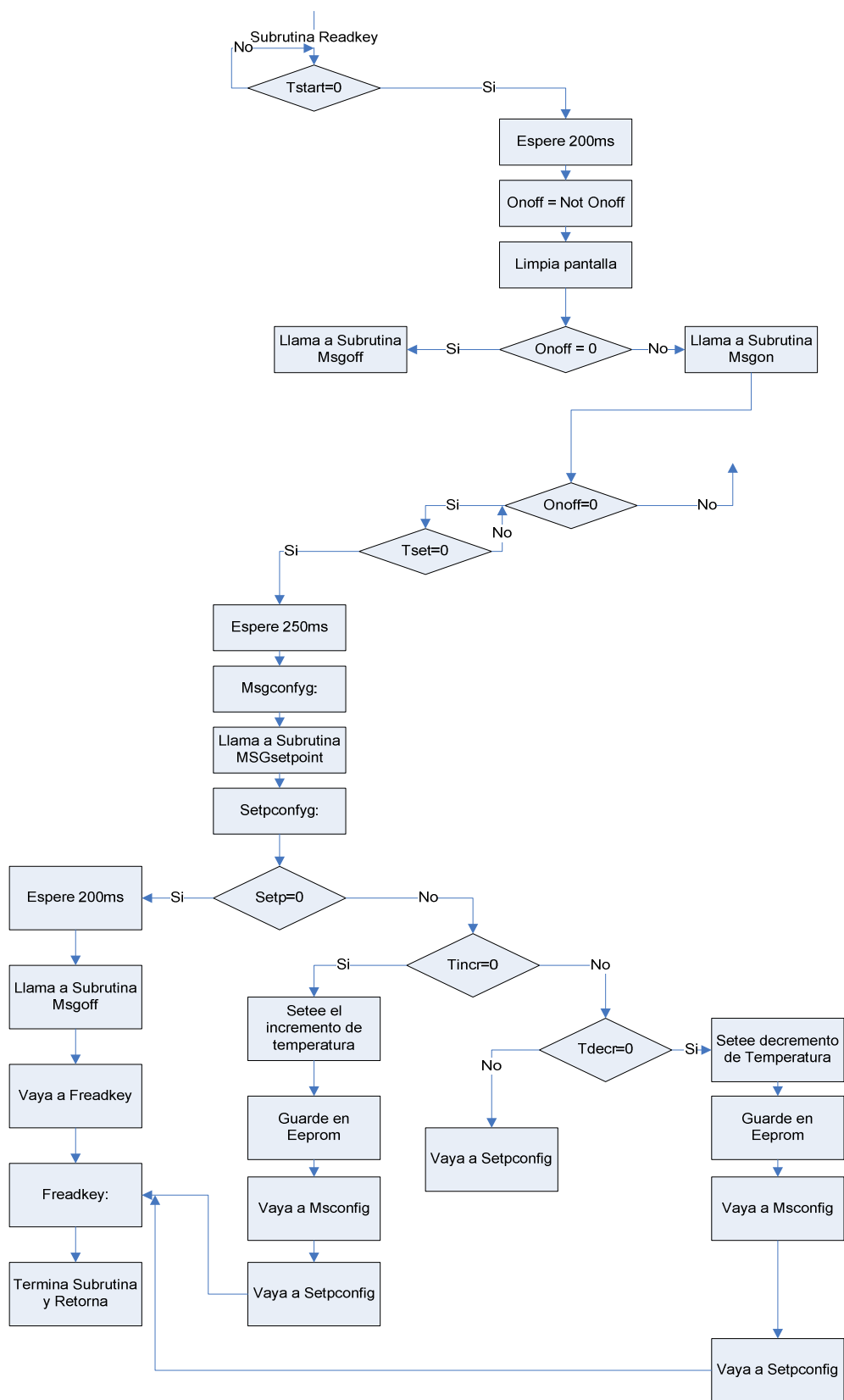


Diagrama 3.3 Subrutina Readkey

La subrutina readkey espera hasta que el usuario presione la tecla Start, cuando ya ha sido presionada espera 200ms y realiza una not a la variable Onoff, limpia la pantalla del lcd y evalúa si la variable Onoff es 0, si es así entonces llama a la subrutina Msgoff si en cambio es diferente de 0 llama a la subrutina Msgon y nuevamente evalúa si la variable Onoff es 0, si esto no se cumple espera a que sea así y evalúa si la variable Tset es igual a 0 si no es así espera, cuando la variable sea igual a cero espera 250ms y va hacia Msconfig en donde es llamada la subrutina MSGsetpoint, después de esto evalúa nuevamente si la variable Setp es igual a cero si se cumple esta condición espera 200ms y llama a la subrutina Msgoff, luego va a Freadkey, finaliza la subrutina y retorna; en caso de que la variable Setp no cumpla con la condición de ser 0 evalúa la variable Tincr, si esta es cero deja que el usuario setee la temperatura y guarda la misma en la Eeprom, seguido a esto indica que vaya a Msconfig, luego a Setpconfig y por ultimo va hacia Freadkey donde la subrutina termina y retorna, en caso de que Tincr no sea igual a cero evalúa la variable Tdecr, si no es igual a cero se mueve hacia Setpconfig, caso contrario si es igual a cero deja que el usuario setee el decremento de temperatura y lo guarda en la Eeprom luego se mueve hacia Msconfig y desde aquí hacia Setpconfig en donde se moverá hacia Freadkey terminando la subrutina y retornando a continuar con la siguiente instrucción del programa.

#### 3.4.4 SUBROUTINA MSGON

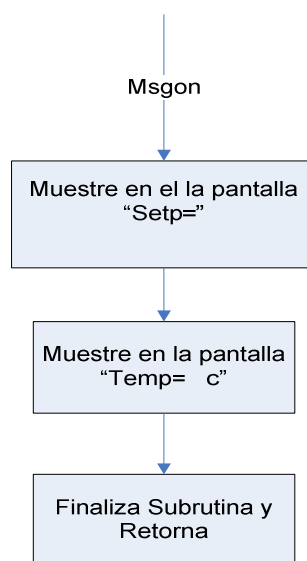


Diagrama 3.4 Subrutina Msgon

Esta subrutina primero muestra en la pantalla del LCD la palabra “Setp=”, y en la parte inferior de la pantalla muestra “Temp= c”, terminado esto finaliza la subrutina y retorna.

### 3.4.5 SUBRUTINA MSGOFF

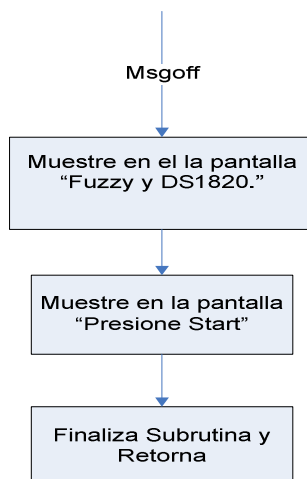


Diagrama 3.5 Subrutina Msgoff

La subrutina Msgoff, muestra en la primera línea del LCD “Fuzzy y DS1820” y en la línea inferior se muestra “Presione Start”, es decir se muestra la presentación del programa al usuario, luego termina la subrutina y retorna.

### 3.4.6 SUBRUTINA MSGSETPOINT

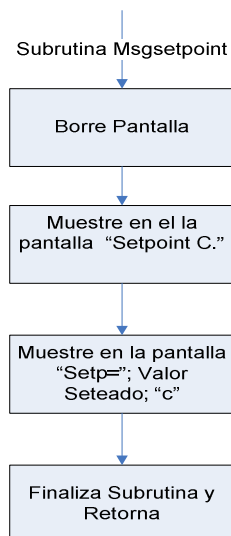


Diagrama 3.6 Subrutina Msgsetpoint.

Esta subrutina empieza por borrar la pantalla del LCD luego muestra en la misma pantalla las palabras Setpoint C, luego mostrara el valor seteado por el usuario, finaliza la subrutina y retorna.

### 3.4.7 SUBRUTINA TEMPERATURE.

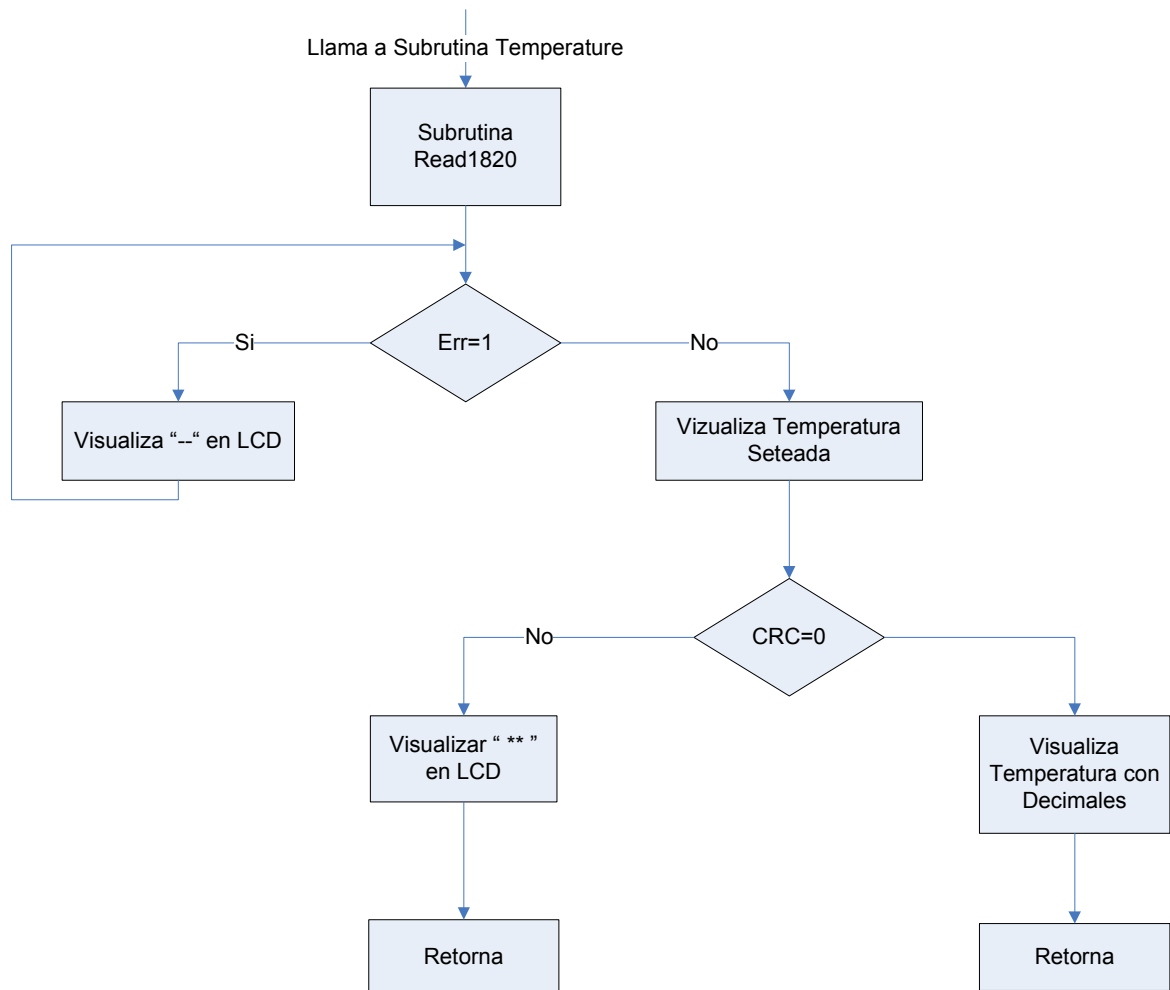


Diagrama 3.7 Subrutina Temperatura.

La subrutina Temperature comienza llamando a la subrutina Read1820 la misma que nos entrega la temperatura medida por el termómetro DS1820, con el valor de temperatura evalúa si el error es igual a 1 si es axial solamente visualiza “—” en la pantalla del LCD caso contrario visualiza la temperatura seteada, luego

evalúa si CRC=0 si es así visualiza la temperatura entregada por el DS1820 con sus decimales y retorna, caso contrario visualiza "\*\*\*" en el lcd y retorna.

### 3.4.8 SUBROUTINA EU

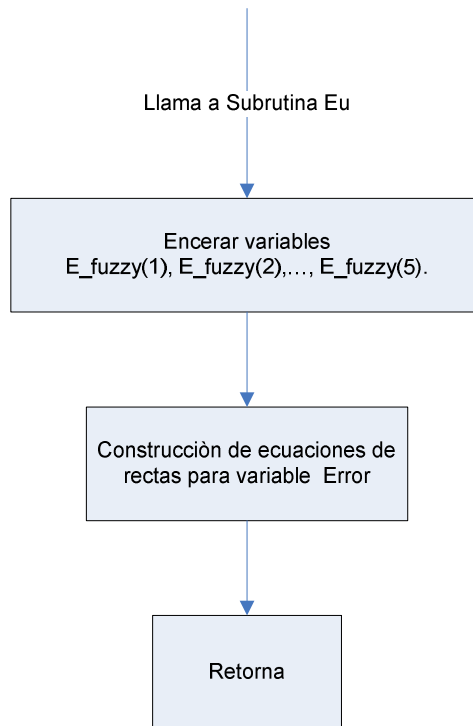


Diagrama 3.8 Subrutina Eu. (Error)

Eu es la subrutina que encera la variable E\_fuzzy y nos ayuda realizando las operaciones aritméticas para construir la ecuación de cada una de las rectas fuzzy para la variable Error.

### 3.4.9 SUBROUTINA RU

La subrutina Ru es muy similar a la subrutina Eu ya que primero encera las variables R\_fuzzy(1), R\_fuzzy(2), .....R\_fuzzy(5), ya enceradas las variables construye las ecuaciones fuzzy para cada una de estas y las guarda para después de esto finaliza y retorna.

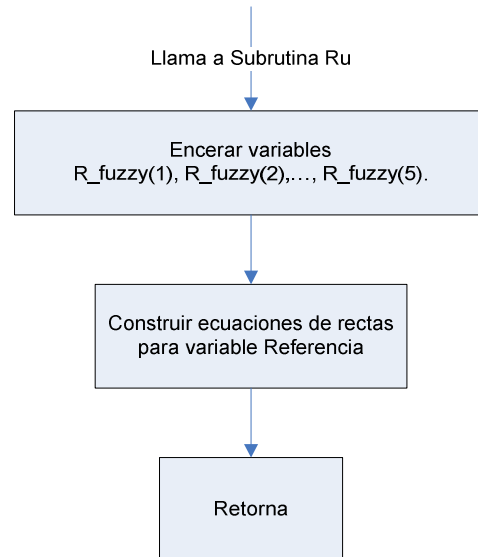


Diagrama 3.9 Subrutina Ru. (Referencia)

### 3.4.10 SUBRUTINA INFERENCE

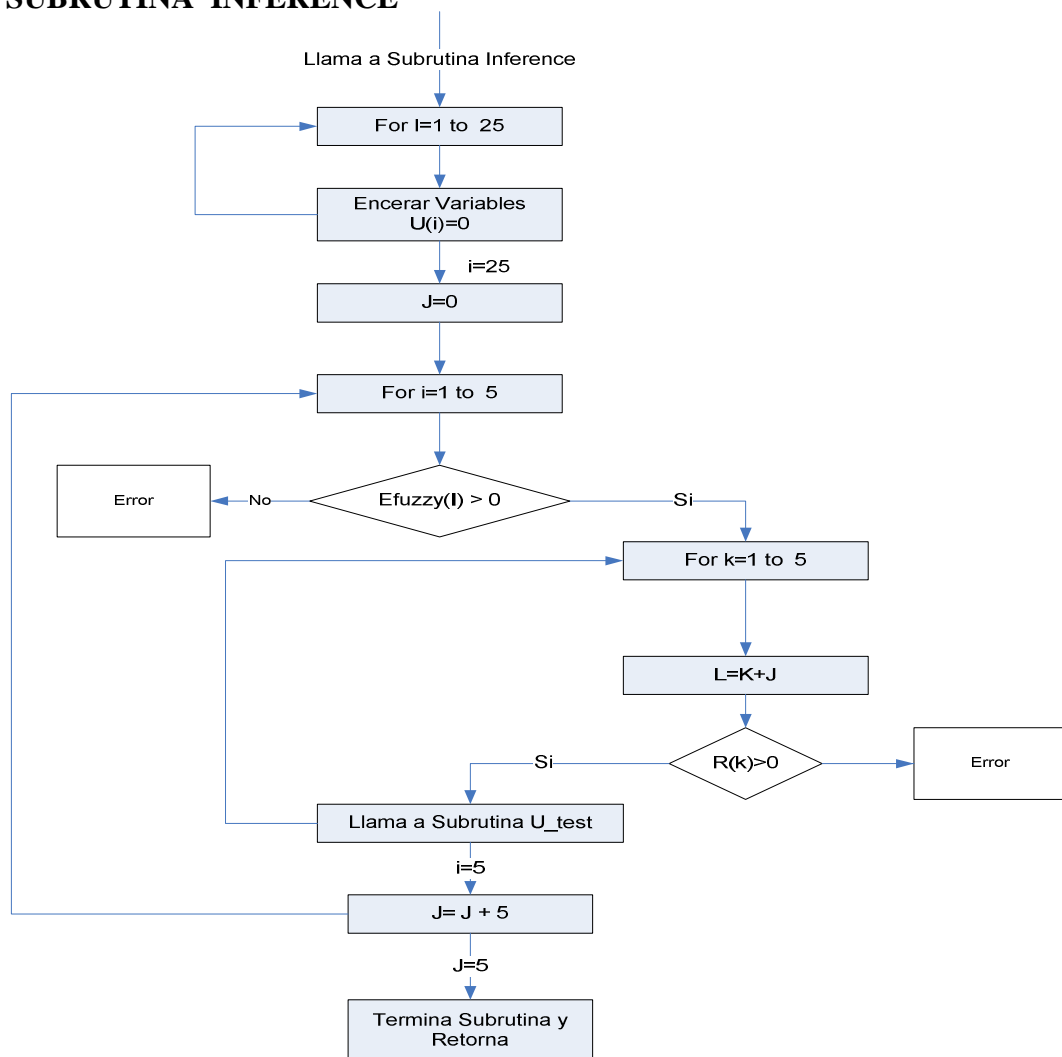


Diagrama 3.10 Subrutina Inference

La subrutina Inference comienza encerrando la variables  $U(1)$ ,  $U(2), \dots, U(25)$ , luego encera la variable  $J$ , luego para cada una de las funciones  $E_{fuzzy}(1)$ ,  $E_{fuzzy}(2), \dots, E_{fuzzy}(5)$  evalúa si es mayor que cero si no es axial incurriríamos en un error en el programa ya que estos valores siempre serán mayores que 0, en cambio si es mayor que 0 entra en un lazo donde realiza la operación  $L=K+J$  y evalúa si las funciones  $R(1)$ ,  $R(2), \dots, R(5)$  son mayores que cero, si es axial llama a la subrutina  $U\_Test$  la misma que se encargara de construir la matriz de inferencia, luego regresa y realiza la operación  $J=J+5$  y repite según el lazo, después de haber evaluado todas las funciones la subrutina termina y retorna.

### 3.4.11 SUBRUTINA $U\_TEST$

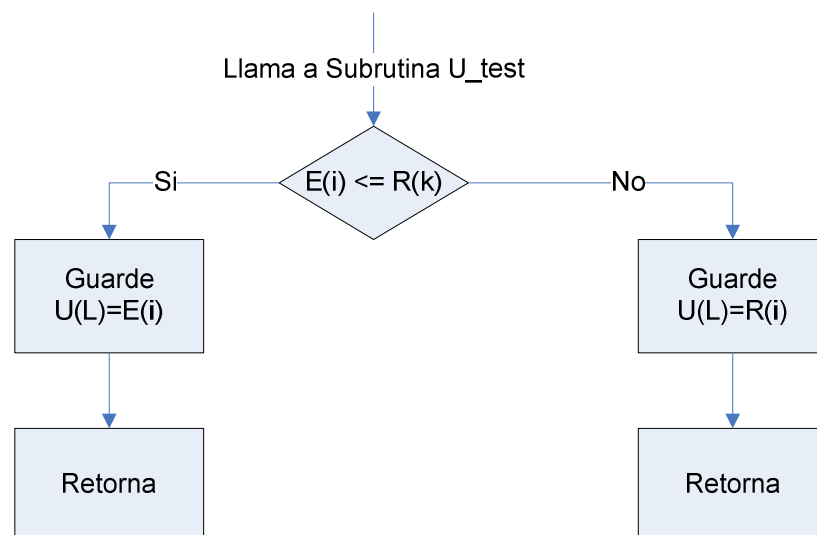


Diagrama 3.11 Subrutina  $U\_TEST$

La subrutina  $U\_test$  realiza la conjunción  $E(i) \cap R(k)$  entre la variable temperatura de referencia y la variable Error guardando en  $U(L)$  siempre el numero de menor valor. Luego de guardado el valor la subrutina termina y retorna.



### 3.4.12 SUBRUTINA DEFUZZY

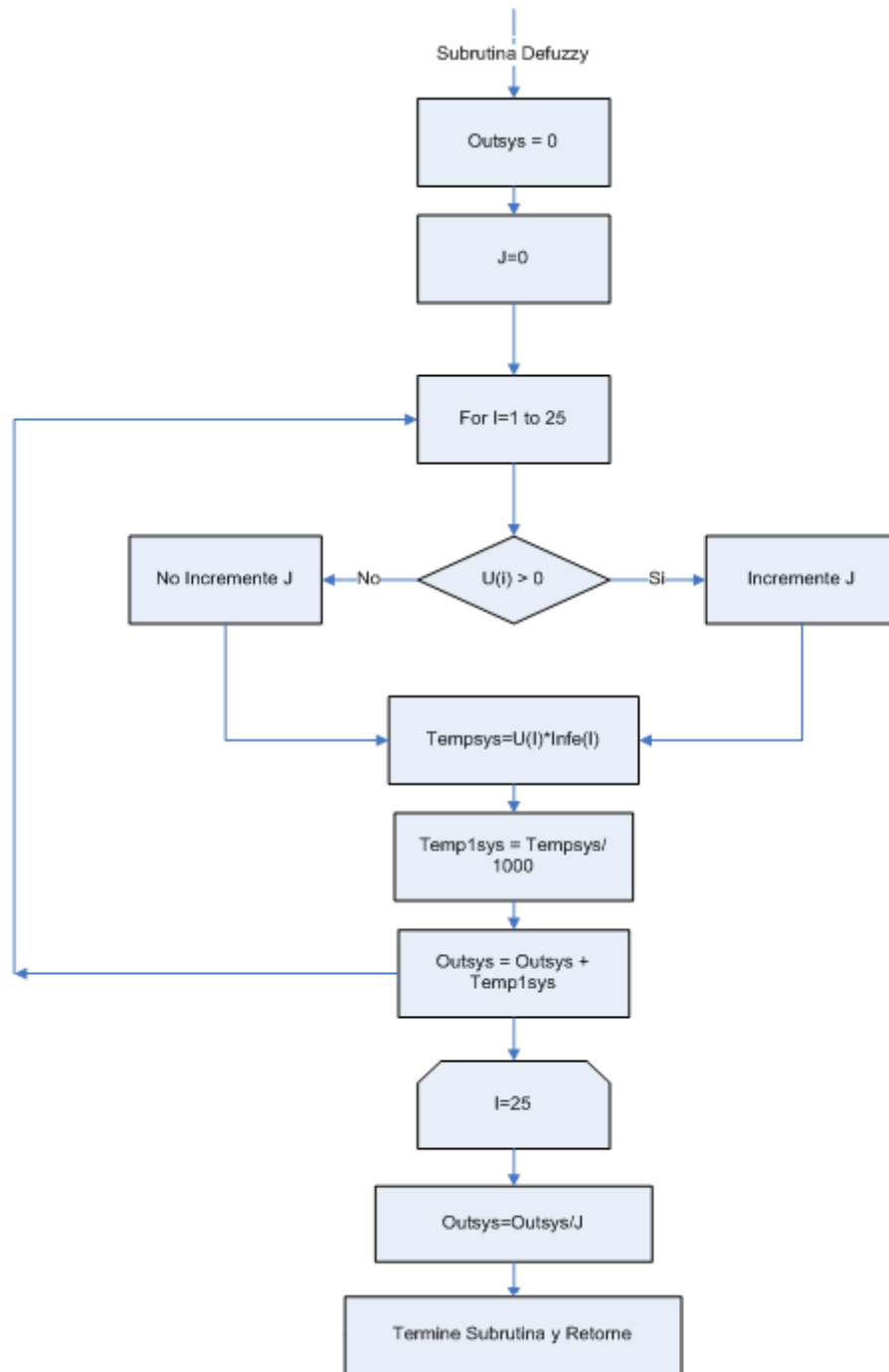


Diagrama 3.12 Subrutina Defuzzy

Esta subrutina se encarga de la defusificación de las variables de salida, comienza encerrando las variables  $Outsys$  y  $J$ , luego realiza un lazo desde 1 hasta 25 dentro de este lazo evalúa individualmente según el lazo si  $U(1)$ ,

$U(2), \dots, U(25)$  es mayor que 0, si es así incrementa  $J$  caso contrario no lo incrementa y continua guardando en la variable Tempsys la multiplicación entre  $U(i) * Infe(i)$ , luego divide la variable Tempsys sobre 1000 y guarda este valor en la variable Temp1sys, finalmente multiplica este nuevo valor por Outsys y lo guarda en Outsys. Al finalizar el lazo for divide la variable Outsys para  $J$ , este valor lo guarda en Outsys, luego termina la subrutina y retorna.

### 3.4.13 SUBROUTINA CRUCE POR CERO

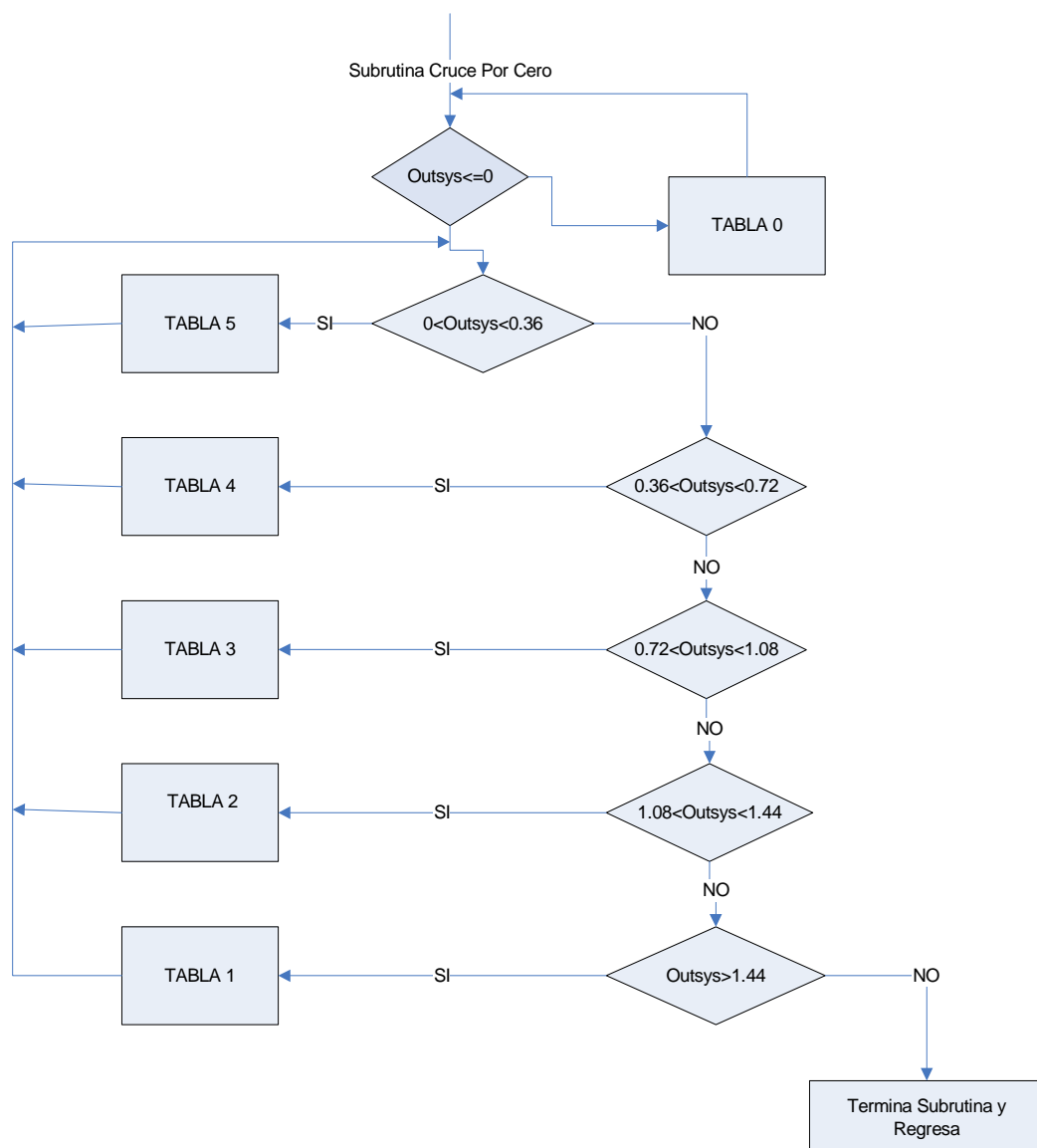


Diagrama 3.13 Subrutina Cruce Por Cero.

A la salida de la subrutina Defuzzy el programa nos entregará valores de *Outsys*, los cuales serán relacionados con la tabla correspondiente al intervalo al cual correspondan, dichos intervalos definirán el ángulo de disparo para el triac y por consiguiente la intensidad de corriente que será entregada al elemento resistivo.

De esta manera si el valor fuera mayor de 1.44 le corresponderá la tabla 1 en lo cual el triac disparará toda la forma de onda ya que la temperatura mediada por el DS1820 está muy por debajo de la temperatura seteada por el usuario y a medida que vaya disminuyendo el valor de *Outsys* entra a pertenecer al intervalo ( $1.08 \leq Outsys \leq 0.144$ ), el mismo que tendrá un periodo de conducción que será controlado por la tabla número 2; si el valor de *Outsys* se encuentra entre ( $0.72 \leq Outsys \leq 1.08$ ) el periodo de conducción de corriente estará dado por la tabla 3, mientras el valor de temperatura sigue acercándose al seteado por el usuario y el valor de *Outsys* se encuentre en un intervalo entre 0.72 y 0.36 ( $0.72 \leq Outsys \leq 0.36$ ) el periodo de conducción estará dado por la tabla 4 cuando el valor de *Outsys* llegue al intervalo entre 0 y 0.36 ( $0 \leq Outsys \leq 0.36$ ) el periodo de conducción estará dado por la tabla 5 en donde el triac dispararía una parte muy pequeña de onda ya que la temperatura medida está muy cercana a la temperatura seteada.

Si el valor de *Outsys* es menor o igual a 0 ( $Outsys \leq 0$ ), quiere decir que la temperatura medida está por encima de la temperatura seteada por el usuario y el dispositivo se apagará.

Para el cálculo de la energía que se enviara a la níquelina se divide a la forma de onda de la intensidad de corriente en ocho partes iguales donde cada una es 2 ms, teniendo así un total de 8 partes correspondientes al periodo de la onda (16ms).

$$T = \frac{1}{60\text{Hz}} = 16\text{ms}$$

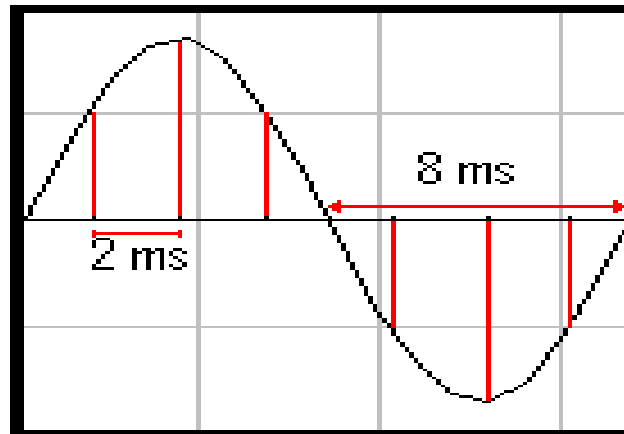


Figura 3.14

A continuación se muestra dichos intervalos:

- 1.- Outsys > 1.44 que corresponde a la Tabla 1 donde ya se dijo se envía toda la forma de onda.

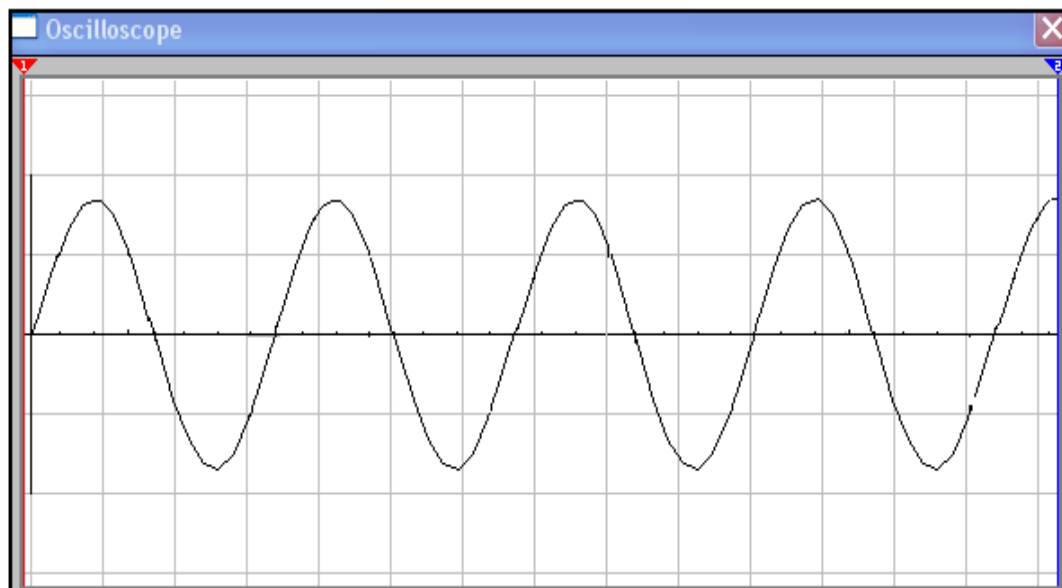


Figura 3.15 Onda Completa

2.-  $1.08 < \text{Outsys} < 0.1.44$  que corresponde a la Tabla 2, la onda en el elemento resistivo solo conducirá 3 partes por semiperiodo, es decir un total de 6 partes por periodo (12ms):

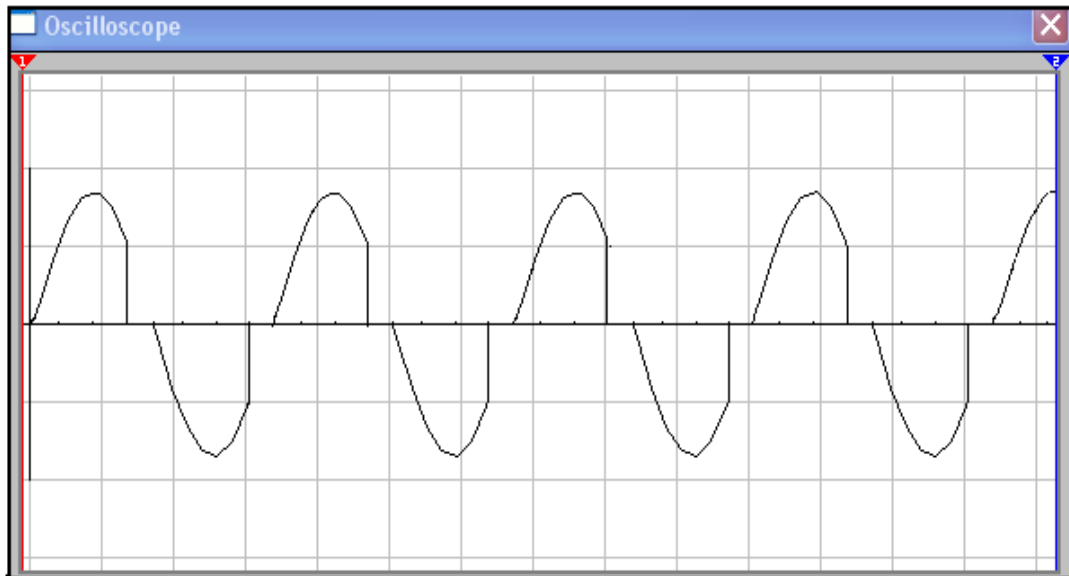


Figura 3.16. Tres partes de la onda por semiperiodo

3.-  $0.72 < \text{Outsys} < 1.08$  que corresponde a la tabla 3, el elemento resistivo tiene una onda con un semiperiodo de 2 partes por semiperiodo y un total de 4 partes de 2ms cada una por período (8 ms):

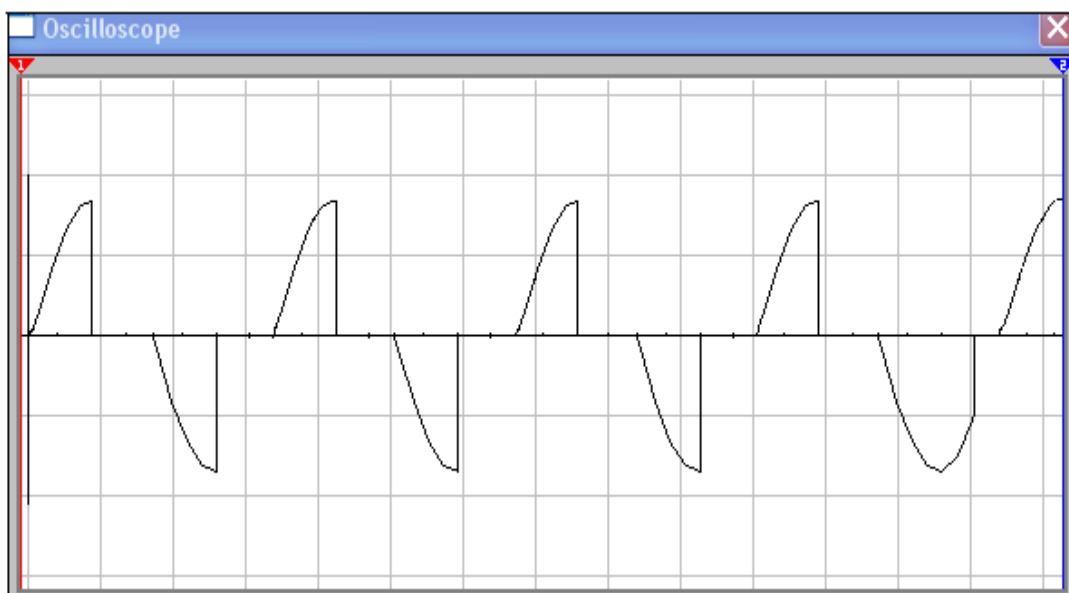


Figura 3.17 Dos partes de la onda por Semiperiodo

4.-  $0.36 < \text{Outsys} < 0.72$ , cuando tenemos este rango el elemento resistivo tendrá una onda con 1 parte por semiperiodo es decir, solo conducirá en total un tiempo de 4 ms, debido a que la temperatura esta cercana a la seteada.

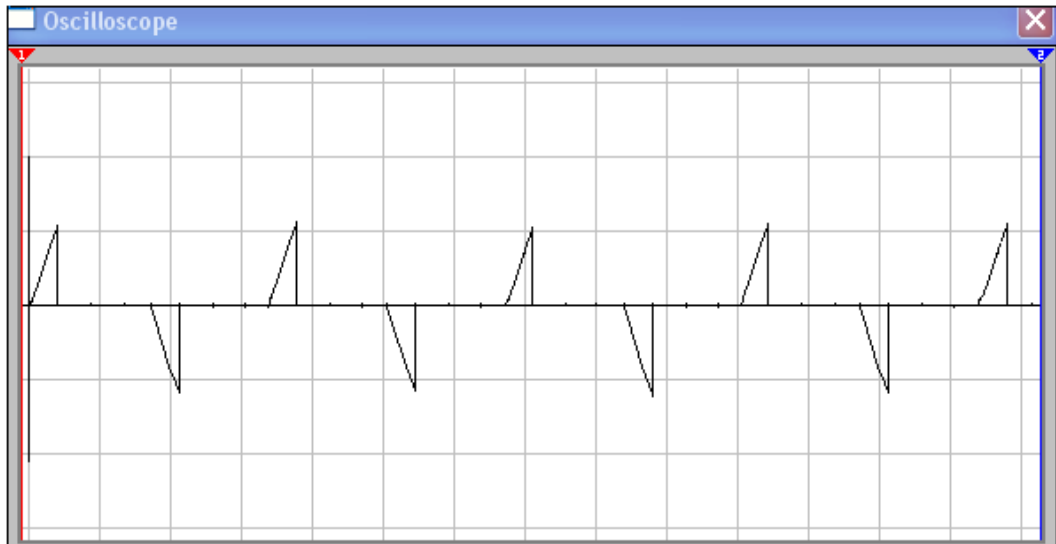


Figura 3.18 Onda con un periodo de conducción total de 2 ms.

5.-  $0 < \text{Outsys} < 0.36$  que corresponde a la Tabla 5 en donde el elemento resistivo se mantiene sin corriente ya que la temperatura en el tanque es igual a la seteada por el usuario, o es mayor a la misma.

Básicamente esto es lo que hace el programa en un lazo infinito, que solo podrá ser detenido o reiniciado por el usuario.

La primera vez que se setea la temperatura, el tiempo que se alcance la misma será único ya que el prototipo siempre estará encendido, logrando el ahorro de energía, pues al llegar a la temperatura seteada por el usuario el dispositivo deja de conducir, es obio que la temperatura del agua empezara a decrementarse; en este momento el dispositivo se encenderá nuevamente, realizará la comparación de temperaturas y conducirá el valor mas conveniente para llegar nuevamente a la temperatura seteada por el usuario, momento en el cual el prototipo no conducirá mas energía al elemento resistivo.

### 3.5 ESQUEMATICO DEL PROTOTIPO.

En esta parte se mostrara el diagrama de la placa construida del prototipo de este proyecto de titulación.

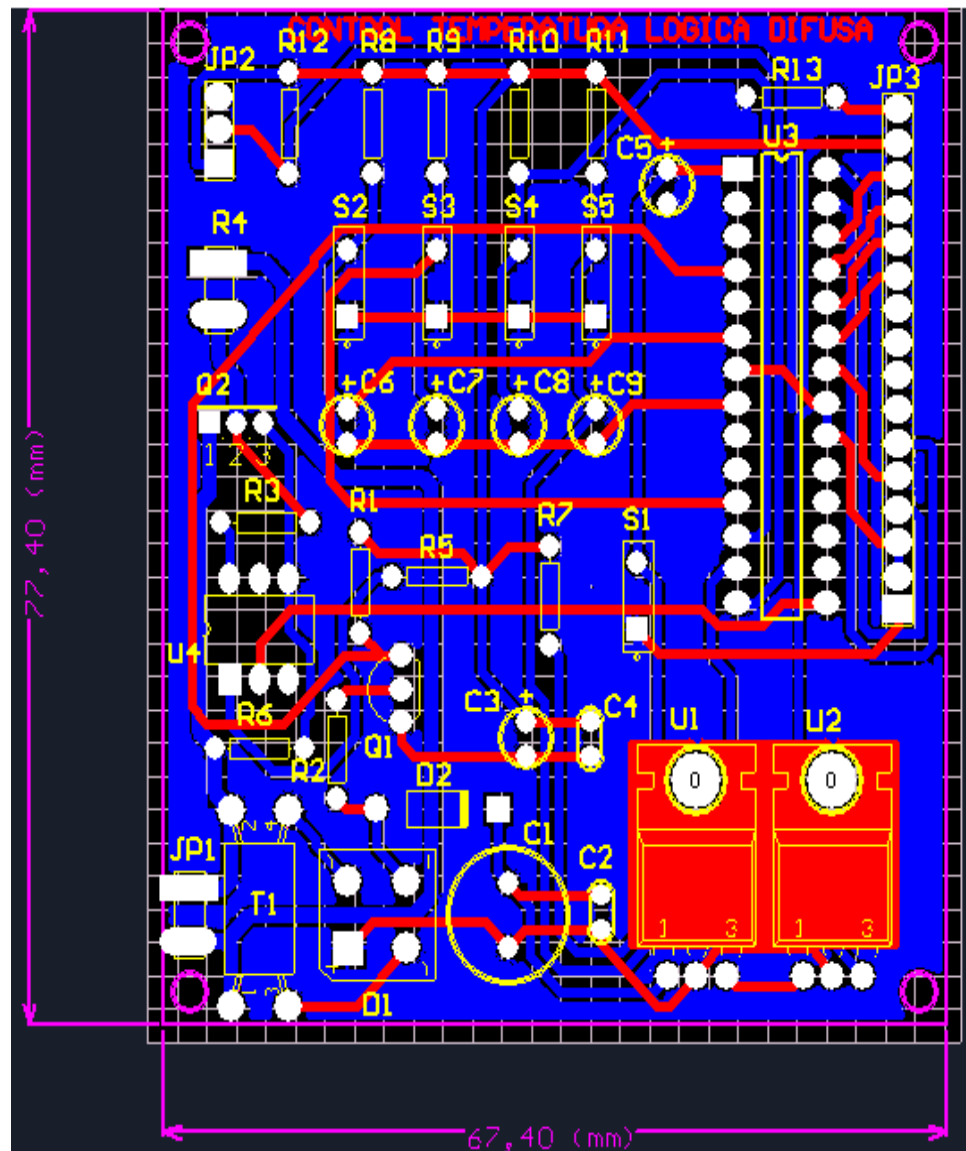


Diagrama 3.19 Diagrama de La Placa Del Protótipo.

A continuación se presenta el esquemático del circuito en el siguiente diagrama:

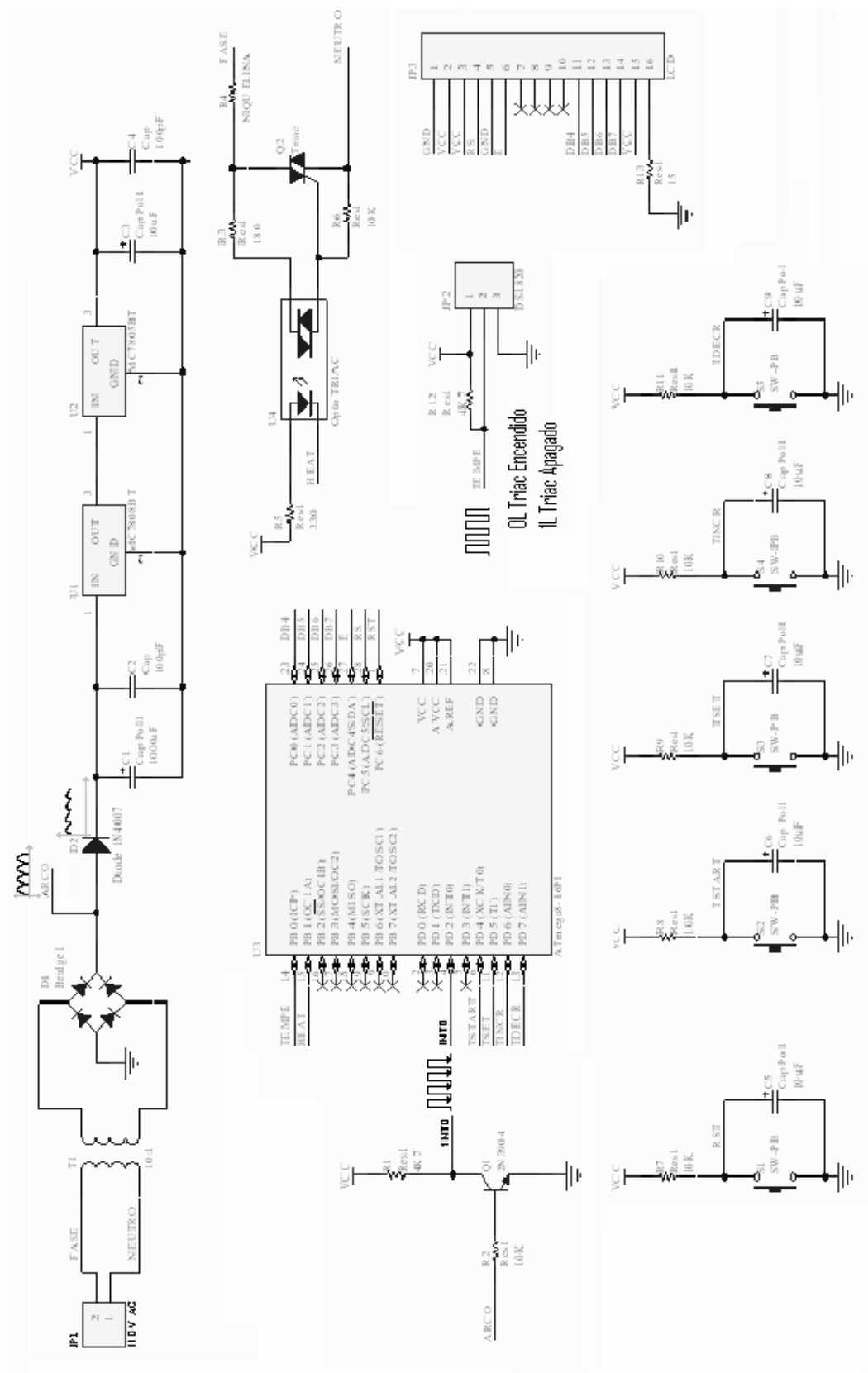


Diagrama 3.20 Esquemático Del Prototipo.



A continuación algunas se presentan algunas fotografías del prototipo y las partes que lo constituyen:

- En la siguiente fotografía se muestra el botón **Start** , que al ser presionado iniciara la toma de datos de temperatura, y por consiguiente todo el desarrollo del programa en si.



Fotografía 3.1 Pulsador que inicia el Programa

- La siguiente imagen muestra los botones para setear la temperatura, mismos que nos van ayudar a incrementar o decrementar la misma por parte del usuario.



Fotografía 3.2. Pulsadores de Seteo

- A continuación se muestra el elemento resistivo (Niquelina) que ira dentro del tanque.

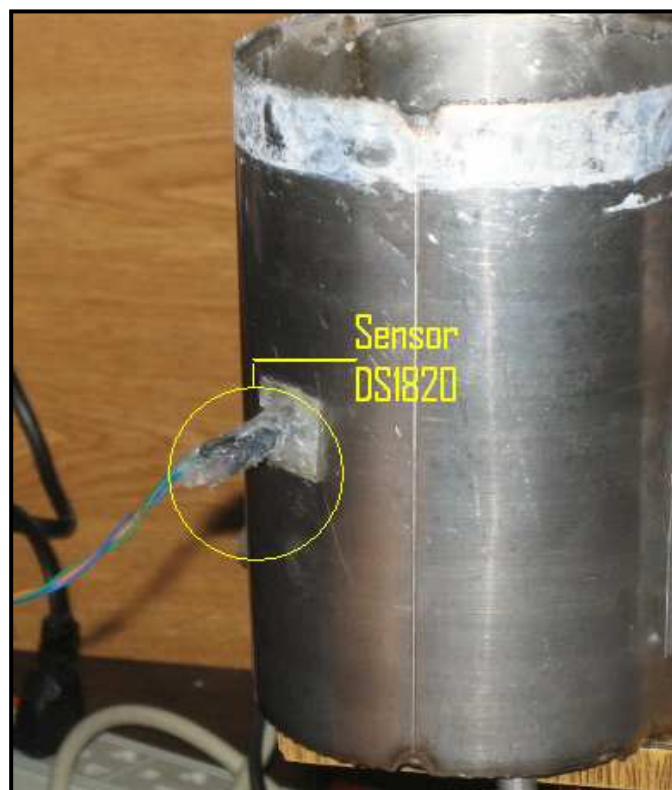


Fotografía 3.3 Elemento Resistivo

- En la siguiente fotografía se observa la ubicación que se le dio al sensor de temperatura dentro del tanque.



Fotografía 3.4. Ubicación Sensor DS1820



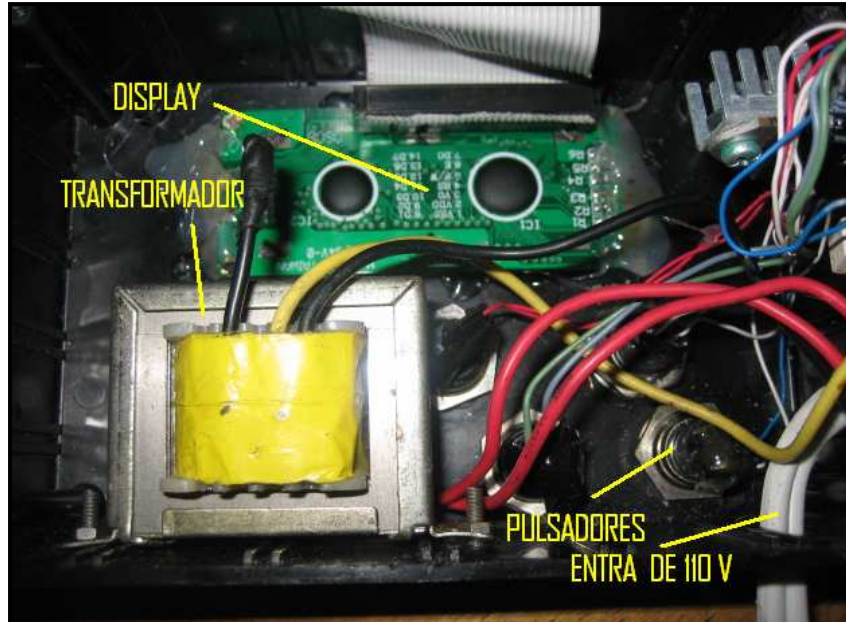
Fotografía 3.5. Ubicación DS1820 parte exterior del tanque.

- En las siguientes fotografías se puede apreciar el interior de la caja, donde se encuentra el prototipo, se observa el bus de datos que conecta el display con la placa y el transformador de 120 v a 9 v. Los 2 cables rojos conducen los 120 V y fue necesario colocarlos, ya que por estos circula una corriente de mas de 21 A, misma que no podía pasar por las pistas de la placa ya que se quemarían.
- Se puede apreciar también la silicona colocada en algunos puntos de la placa con el fin de aislar los mismos y de esta manera evitar un cortocircuito.

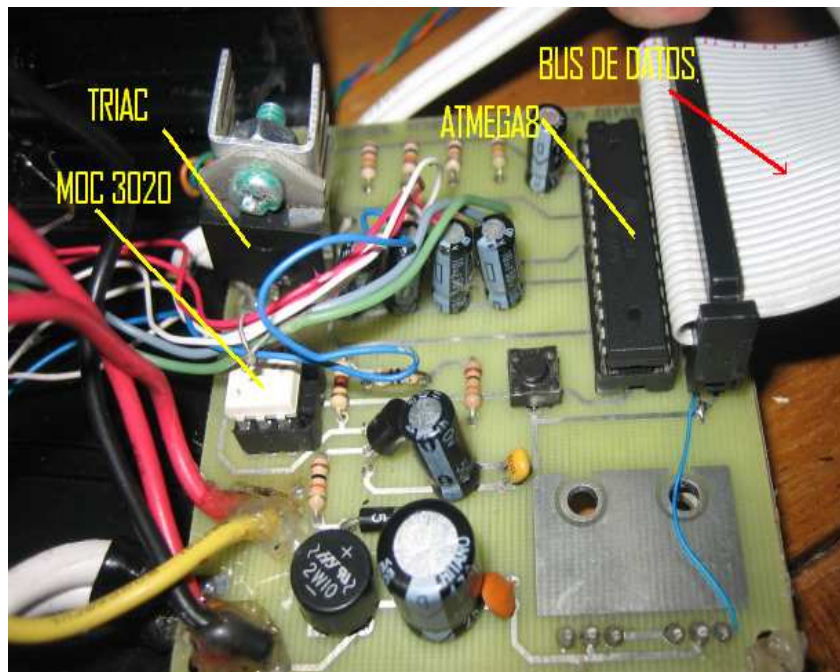


Fotografía 3.6. Placa Protótipo.

- En las siguientes fotografías se observa la parte posterior del display, el transformador y los pulsadores que nos ayudarán a la configuración del prototipo.



Fotografía 3.7. Parte posterior del display y pulsadores.



Fotografía 3.8. Circuitería, parte frontal de la placa.

**Fotografía 3.9. Reset y conexión DS1820**

El prototipo se realizo, tomando en cuenta las siguientes premisas de diseño:

1. La niquelina será ubicada en un lugar donde su energia calorífica sea aprovechada a su máximo, por esta razón se la ubico en la parte inferior del tanque.
2. El desfogue da agua será ubicado en la parte superior del tanque a una distancia prudente del sensor DS1820 para que los datos tomados por este sean lo mas reales posibles.
3. El tanque tiene una capacidad de  $890.6 \text{ cm}^3$ .
4. Se encuentra elaborado en Metal.
5. Cuenta con varias entradas, mismas que serán aisladas para el uso del prototipo.
6. Se lo sello con silicona, para tener acceso al mismo, cuando se desee realizar alguna modificación.





Fotografia 3.10. Tanque, diferentes angulos

## **CAPITULO 4**

### **PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS.**

En el presente capítulo se realizara un análisis del presupuesto del costo versus el ahorro del agua y energía eléctrica y sus ventajas como también los resultados de las pruebas de funcionamiento del prototipo del controlador de temperatura.

En este proyecto el control de temperatura permanece encendido mientras las llaves de agua estén cerradas. Este sistema le ofrece al usuario agua caliente instantánea. Si por razones de racionamiento usted se ve obligado a desenchufar su calentador tradicional, al regresar el agua deberá esperar a que se caliente el agua del tanque. Con este prototipo simplemente se abre una llave y se disfruta de **agua caliente sin tener que esperar** y sin tener la necesidad de encenderlo.

#### **4.2 PRUEBAS REALIZADAS.**

En esta sección con finalidad de asegurar un correcto funcionamiento, configurando todos los parámetros necesarios del prototipo se realizaron las debidas pruebas.

También en lo que es el consumo de energía eléctrica, primero se debe saber cuanta potencia utilizan tanto un termostato así como es una ducha eléctrica. En el caso de una ducha eléctrica el consumo es de alrededor de 3000 A 4000 vatios / hora.

##### **4.2.1 PRUEBAS Y ANALISIS CON RESPECTO AL DESPERDICIO DE AGUA**

A continuación se muestra una tabla con valores tomados experimentalmente utilizando una ducha eléctrica y por otra parte nuestro prototipo, para así tomar un



punto de relación entre los dos y encontrar de esta manera el ahorro que se tendrá especialmente con el agua o el líquido que se está utilizando.

Temperatura °C	Cantidad (lts)	Tiempo (s)
	Ducha Eléctrica	
42	1	21,5
42	1,2	26,4
42	1,25	30,47
42	1,3	36,12
42	1,5	32,43
42	1,15	24,42
42	1,35	26,45
42	0,95	21,68
42	1,4	29,13
42	2	40,42
42	2,3	45,01
42	1,5	35,98
42	1,2	30,88
42	1,7	38,45
42	1,3	25,27
42	1,55	27,43
42	1,2	23,29
42	1,75	32,23
42	1,6	30,9
42	0,8	19,95
42	1,25	27,37
42	1,3	29,96
42	1,3	32,33
42	1,45	34,46
42	1,6	36,72
42	1,8	38,45
42	2,1	40,5
42	2,4	42,26
42	2,2	41,53
<b>PROMEDIO</b>	<b>1,496551724</b>	<b>31,79275862</b>

**Tabla 4.1 Cantidad de Agua desperdiciada con la ducha eléctrica.**

En la tabla anterior se puede apreciar la cantidad de agua que se usa hasta llegar a una temperatura adecuada para el usuario, en este caso 42 °C, como dato adicional se toma el tiempo que demora en llegar a dicha temperatura.

Todas las medidas se realizaron con un periodo de 30 minutos entre estas , ya que es necesario que el elemento resistivo (niquelina), y el liquido vuelva a su estado inicial (temperatura ambiente).

	<b>Cantidad (lts)</b>	
<b>Temperatura</b>	<b>Prototipo</b>	<b>Tiempo (s)</b>
42	0,8	19,93
42	0,6	18,1
42	0,75	18,56
42	1,1	26,12
42	1,15	24,45
42	0,6	18,3
42	0,92	21,52
42	0,85	19,95
42	0,75	18,54
42	1,1	26,15
42	1,15	24,48
42	1,1	26,12
42	1,1	26,17
42	1,3	26,46
42	0,95	21,71
42	0,67	18,33
42	0,8	19,95
42	0,85	19,94
42	0,88	20
42	0,9	21,58
42	1,2	26,9
42	0,7	18,53
42	0,8	19,55
42	0,9	20,1
42	1,5	31,6
42	1	25,95
42	0,75	18,5
42	1,1	26,3
42	0,7	18,49
<b>PROMEDIO</b>	<b>0,93</b>	<b>22,14758621</b>

**Tabla 4.2 Cantidad de agua desperdiciada con el Prototipo.**

Se puede apreciar claramente que no existe un ahorro sustancial de agua en la primera vez que se utiliza el prototipo, esto es normal ya que el ahorro del mismo viene dado a partir de la segunda vez que se utilice, pues el prototipo siempre estará encendido para que de esta manera la temperatura del agua permanezca constante.

En este caso al igual que con la ducha eléctrica fue necesario dar un periodo de espera para tomar cada uno de los datos y así poder relacionarlos.

#### 4.2.1.1 CALCULO DEL AHORRO DE AGUA

Tomando en cuenta los valores tabulados en las tablas anteriores se procede a realizar el cálculo con respecto al ahorro y desperdicio de agua.

- ***Ducha Eléctrica***

Promedio de agua desperdiciada: 1.496 litros

Tiempo promedio: 31.79 segundos

- ***Ahorro Prototipo (1era vez)***

Promedio de agua desperdiciada: 1.39 litros

Tiempo promedio: 31 segundos

$$\% \text{ ahorro} = \frac{1.496 - 1.39}{1.496} \times 100 = 7.086\%$$

- ***Ahorro Prototipo ( A partir de la segunda vez)***

Promedio de agua desperdiciada: 0.02 litros

Tiempo promedio: 2 segundos

$$\% \text{ ahorro} = \frac{1.496 - 0.02}{1.496} \times 100 = 98\%$$

Entonces el ahorro aproximado que se obtendría con el prototipo en relación a la cantidad de agua desperdiciada es del 98%, a partir de la segunda vez de uso, es por esto la importancia de que el prototipo siempre se mantenga encendido.

#### 4.2.2 PRUEBAS CON RESPECTO AL USO DE ENERGÍA.

En esta sección de este capítulo se mostraran las pruebas que se realizaron en el laboratorio a fin de comprobar la existencia de un ahorro en la energía eléctrica al usar este prototipo.

A continuación se mostraran fotografías de las formas de onda obtenidas, de una serie de temperaturas que fueron medidas por el sensor de temperatura hasta alcanzar la temperatura seteada.

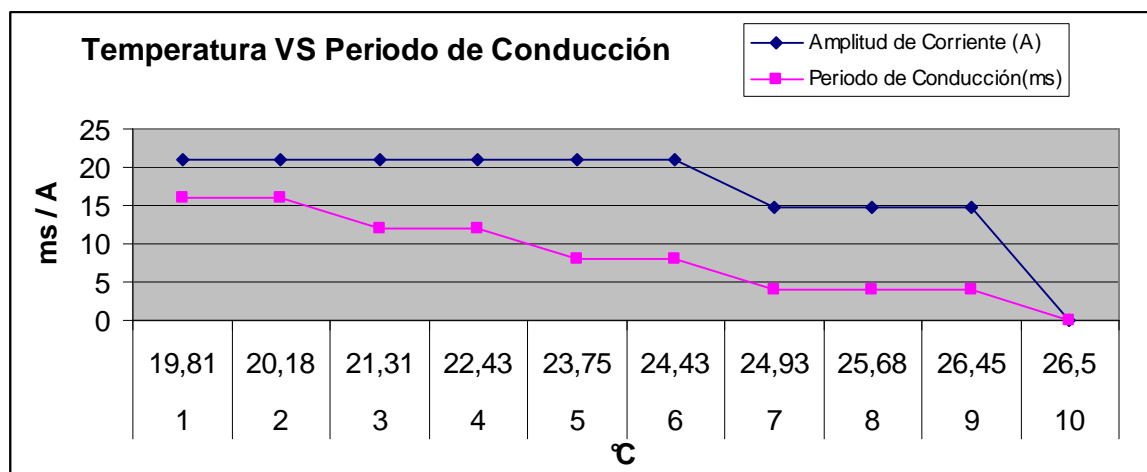
**T seteada =25 grados centígrados**

De las pruebas realizadas en el laboratorio podemos apreciar que el ancho del pulso va disminuyendo y haciéndose más pequeño cada vez que la temperatura se incrementa acercándose a la temperatura seteada por el usuario.

En la siguiente tabla con su respectivo gráfico se puede apreciar mejor el cambio en el periodo de conducción con respecto al aumento de la temperatura en el agua.

<b>Nº de Medida</b>	<b>Temperatura (°C)</b>	<b>Amplitud de Corriente (A)</b>	<b>Periodo de Conducción(ms)</b>
1	19,81	21	16
2	20,18	21	16
3	21,31	21	12
4	22,43	21	12
5	23,75	21	8
6	24,43	21	8
7	24,93	14.84	4
8	25,68	14.84	4
9	26,45	14.84	4
10	26,5	0	0

**Tabla 4.3. Relacion entre Ancho de Pulso y Aumento de Temperatura**



**Gráfico 4.1. Temperatura Vs Periodo de Conducción**

El ahorro de energía viene dado ya que en una ducha eléctrica la potencia a la carga (niquelina) es siempre constante durante todo el tiempo que es utilizada por un usuario, mientras que el microcontrolador Atmega8 de este prototipo en el instante que se acerca a la temperatura seteada por el usuario, la intensidad de corriente enviada a la niquelina disminuirá, y desde este punto siempre se mantendrá constante, obteniendo un ahorro de la siguiente forma:

Para los datos tomados sacamos el promedio de Corriente:

Tomando en cuenta que la ducha eléctrica absorbe constantemente 21 A;

$$\frac{(21 + 21 + 21 + 21 + 21 + 21 + 14.84 + 14.84 + 14.84 + 0)}{10} = 17.052$$

$$\% \text{Ahorro(Primera Vez)} = \frac{21 - 17.052}{21} \times 100 = 18.8\%$$

$$\frac{(14.84 + 0 + 14.84 + 0 + 14.84 + 0 + 14.84 + 0 + 14.84 + 0)}{10} = 7.42$$

$$\% \text{Ahorro(A partir de la 2da Vez)} = \frac{21 - 7.42}{21} \times 100 = 64.66\%$$

Se puede ver claramente que el ahorro a partir de la segunda vez que el usuario use el prototipo es muy notorio, llegando a ser máximo mientras más se utilice el mismo.

### **4.3 FACTORES QUE AFECTARIAN LA TEMPERATURA DE SALIDA DEL AGUA:**

#### **4.3.1 LA DISTANCIA ENTRE EL CALENTADOR Y EL PUNTO DE USO DE AGUA CALIENTE:**

Cuando el agua entra a la tubería que la va a transportar hasta su lugar de uso comienza un intercambio de calor entre la tubería y el ambiente. Estos intercambios de calor no son otra cosa que pérdidas de energía (en forma de calor para nuestro caso), los tres factores que influyen en el porcentaje de pérdida son: el largo de la tubería, el material de que está hecha la tubería y el aislamiento que esta pueda tener. Cuanto mas larga sea la tubería que conduce el agua caliente mayor es la pérdida y menor la temperatura de salida. El cuanto a los materiales es la tubería de cobre la que tiene más pérdida y la de PVC la que tiene menor pérdida. Las construcciones de buena calidad incluyen tubería de cobre por ser la más duradera, pero la envuelven con un aislante térmico. Aunque un aislante térmico puede parecer costoso a primera vista, el ahorro de energía (electricidad) que este proporciona paga su costo varias veces. Este ahorro es mayor en zonas de clima frío.

#### **4.3.2 EI CAUDAL A USAR.**

Los datos presentados a continuación fueron tomados con caudal constante y con una temperatura seteada de 33 °C durante todo el tiempo de la prueba, hay que tomar en cuenta que se tabularon un total de 20 valores sin ningún periodo de tiempo definido entre ellos. El objetivo es tratar de definir que cuanto mayor sea el caudal de agua, para el prototipo será más complicado mantener constante la temperatura seteada.

Caudal: 2 L/min

N° de Medida	Temperatura Seteada °C	Temperatura Prototipo °C	Litros por minuto (Caudal)
1	33	33.4	2
2	33	33.8	2
3	33	34.2	2
4	33	34	2
5	33	33.5	2
6	33	33	2
7	33	33.6	2
8	33	33.4	2
9	33	33.5	2
10	33	33.9	2
11	33	34.4	2
12	33	33.1	2
13	33	33.52	2
14	33	34.11	2
15	33	34.42	2
16	33	33.33	2
17	33	33.31	2
18	33	33.02	2
19	33	32.5	2
20	33	32.98	2

Tabla. 4.4. Comportamiento de la temperatura con un caudal constante

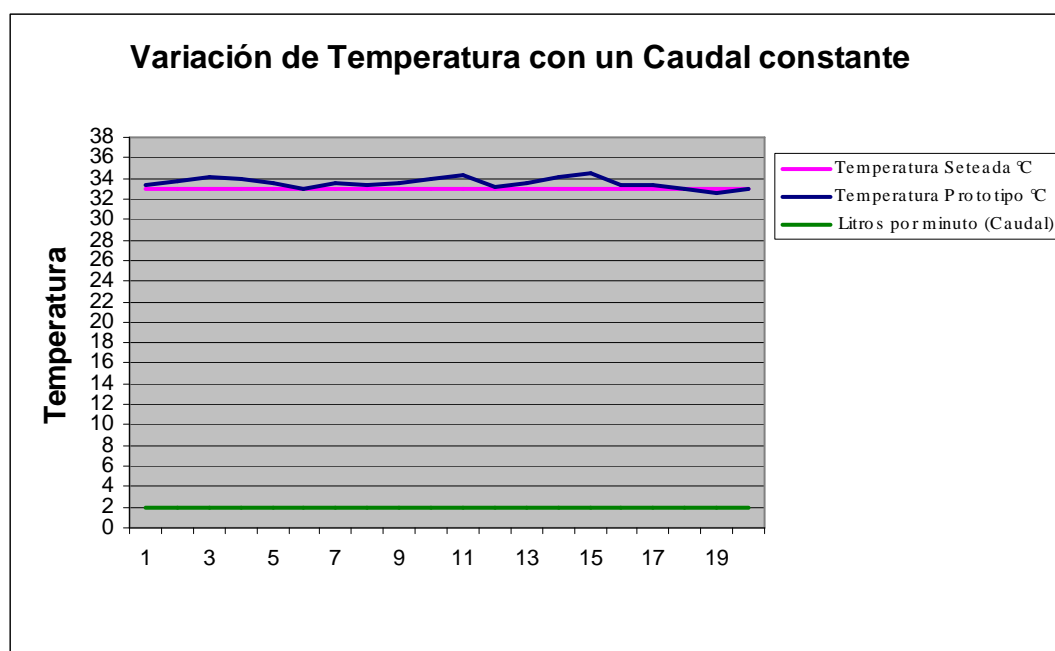


Gráfico 4.2. Variación de temperatura con un caudal constante

Caudal: 4L/min

N° de Medida	Temperatura Seteada °C	Temperatura Prototipo °C	Litros por minuto (Caudal)
1	33	34.51	4
2	33	34.4	4
3	33	33.22	4
4	33	33.49	4
5	33	32.65	4
6	33	32.2	4
7	33	32.12	4
8	33	32.98	4
9	33	32.65	4
10	33	32.44	4
11	33	32.69	4
12	33	32.78	4
13	33	33.5	4
14	33	33.15	4
15	33	32.86	4
16	33	32.67	4
17	33	32.12	4
18	33	31.76	4
19	33	31.98	4
20	33	32.1	4

Tabla 4.5 Pruebas Temperatura, Caudal, Tiempo.

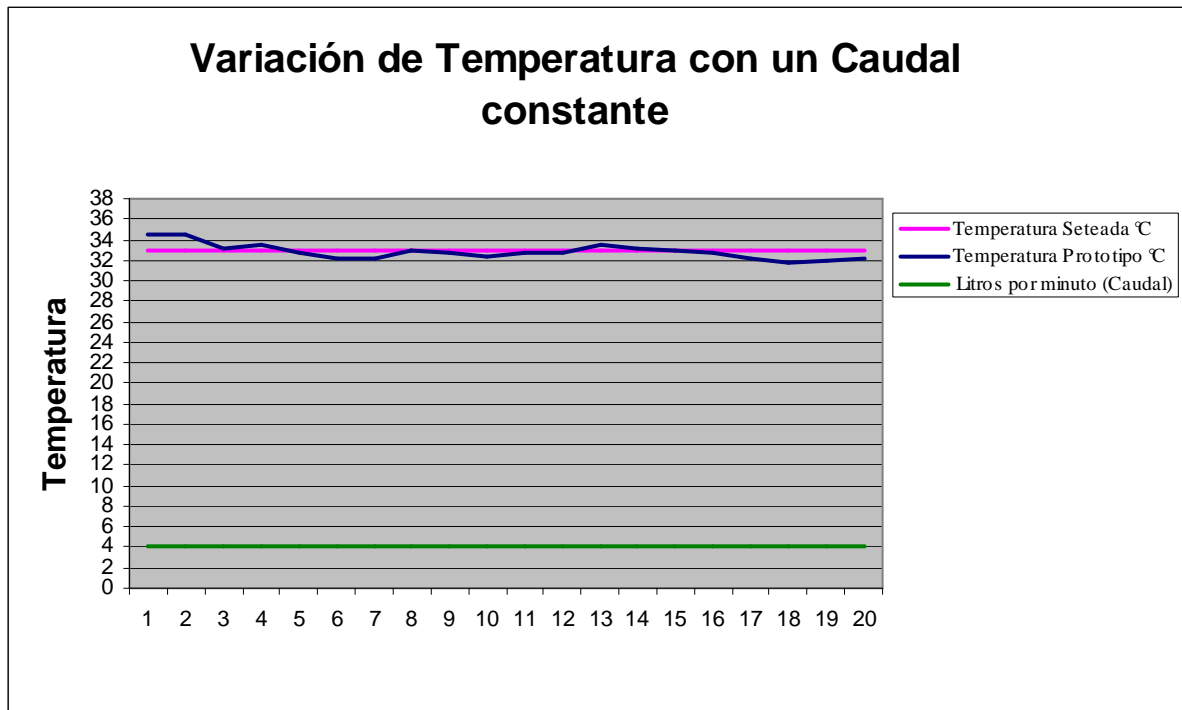


Gráfico 4.3. Variación de temperatura con un caudal constante.



#### 4.4 CONSTRUCCIÓN Y COSTOS.

En esta sección se realizara el estudio de costo, de la construcción y montaje del diseño del presente prototipo los precios de estos elementos son referenciales al los meses de noviembre del 2006 y enero del 2007, los cuales permiten calcular un valor muy aproximado para su producción unitaria. Para realizar esta estimación es imprescindible conocer la lista de materiales, suministros, etc., que se necesiten.

##### 4.4.1 COSTOS DE ELEMENTOS NORMALIZADOS UTILIZADOS.

La siguiente lista presentada en la tabla 3.2, son los elementos necesarios para la construcción.

<b>DENOMINACION</b>	<b>ESPECIFICACION</b>	<b>CANTIDAD</b>	<b>COSTO UNITARIO (dolares)</b>	<b>COSTO TOTAL (dolares)</b>
<b>ATMEGA8</b>	Microcontrolador	1	7.50	7.50
<b>CAPACITORES</b>	Electrolíticos y cerámico	4	0.20	0.80
<b>DISPLAY</b>	LCD	1	14.00	14.00
<b>OPTOTRIAC</b>	MOC 3020	1	1.20	1.20
<b>TRANSFORMADOR</b>	De 120 a 9 voltios	1	4.00	4.00
<b>RESISTENCIAS</b>		7	0.02	0.14
<b>PULSADORES</b>	Normales	4	0.12	0.48

<b>REGUARDOR</b>	7805	1	0.40	0.40
<b>TRIAC</b>	ECG5638	1	0.80	0.80
<b>RECTIFIADOR</b>	De onda completa tipo puente	1	0.30	0.30
<b>DS1820</b>	Sensor de Temperatura	1		

**Tabla 4.3 Precios de los Diferentes Elementos Empleados.**

En total el prototipo tiene el costo de ..... dólares sin mencionar el costo de realizar el impreso del mismo donde son 20 dólares, esto demuestra que para los beneficios de ahorro que podemos obtener en agua y energía eléctrica no es un costo muy alto.

## **CAPITULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

- Hay que tomar muy en cuenta el dimensionamiento de la fuente a utilizar ya que como hacemos uso de un transformador que soporta solo un amperio de corriente es preciso que el calculo del consumo total de corriente por parte de los elementos que serán alimentados por nuestra fuente se lo realice de forma precisa para no tener problemas de alimentación en nuestros dispositivos.
- El uso de un transformador de tan solo un amperio se debe al tamaño del mismo ya que transformadores de mayor amperaje serán siempre de tamaño mayor al utilizado. Es también preciso indicar que nuestro

transformador cumple eficazmente con los requisitos y propósitos para nuestro proyecto.

- El tanque utilizado en nuestro proyecto fue construido previamente a escala tomando en cuenta que el elemento resistivo a utilizado es una resistencia con un valor de 8 ohmios, calculada para aprovechar al máximo la potencia que brindara la misma, y así poder alcanzar la temperatura requerida en un tiempo mínimo.
- El tamaño de nuestro prototipo es muy importante ya que como se utiliza en un tanque de agua no debe llamar la atención, es por este motivo que el diseño del calentador electrónico será lo mas pequeño posible.
- Se decidió utilizar el LCD en vez de Displays debido a que se necesitaba un interfaz fácil de utilizar para el usuario, es decir que la entrada de datos y el inicio del programa sea lo mas amigable posible.
- Otra opción para realizar el envío de corriente hacia el elemento resistivo consiste en utilizar el principio de un DIMMER que no es más que un controlador de corriente.
- Es importante que al hacer uso del prototipo, el tanque se encuentre siempre lleno de agua ya que al no ser así el elemento resistivo no tardara en quemarse es por esto que se puede hacer uso de un sensor de nivel que indicara el nivel de agua que existe en el tanque.
- Debido a que el termómetro digital DS1820 medirá el agua en la salida del tanque es de vital importancia que se realice una correcta impermeabilización en este para evitar un corto circuito y de esta forma la perdida del dispositivo.
- El error que se tendrá con respecto a la temperatura seteada por el usuario en el prototipo es de aproximadamente 1,5 °C ya que como se ha definido

en el segundo capítulo la diferencia de temperatura que el cuerpo humano es capaz de sentir es de aproximadamente 2 °C motivo por el cual se encontrara trabajando dentro de los parámetros definidos.

- Para conseguir una mejor transferencia de calor por parte del elemento resistivo hacia el agua se pueden realizar arreglos de niquelina. Ya que la temperatura del agua en nuestro tanque dependerá en gran parte de la resistividad de la niquelina y por consiguiente de la potencia irradiada por esta.
- Es de vital importancia que la tabla de inferencia se encuentre construida de manera tal que los valores tomados de esta sean los ideales para optimizar el diseño de cualquier proyecto que trabaje con Lógica Difusa, ya que si existe algún error con estos valores el programa de dicho proyecto no funcionara de la manera más eficiente.
- El cálculo que se realizó con respecto a la lectura del termómetro digital 1wire DS1820 nos permite visualizar la temperatura del ambiente que se encuentre testeando con 2 decimales de precisión, ayudados así por ejemplo en un proyecto de Termocunas donde el control de temperatura resulta ser muy importante y a su vez muy crítico.

# ANEXO I



## Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 1K Byte Internal SRAM
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Three PWM Channels
  - 8-channel ADC in TQFP and MLF package
    - Six Channels 10-bit Accuracy
    - Two Channels 8-bit Accuracy
  - 6-channel ADC in PDIP package
    - Four Channels 10-bit Accuracy
    - Two Channels 8-bit Accuracy
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-lead PDIP, 32-lead TQFP, and 32-pad MLF
- Operating Voltages
  - 2.7 - 5.5V (ATmega8L)
  - 4.5 - 5.5V (ATmega8)
- Speed Grades
  - 0 - 8 MHz (ATmega8L)
  - 0 - 16 MHz (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C
  - Active: 3.6 mA
  - Idle Mode: 1.0 mA
  - Power-down Mode: 0.5  $\mu$ A



**8-bit AVR<sup>®</sup>**  
**with 8K Bytes**  
**In-System**  
**Programmable**  
**Flash**

**ATmega8**  
**ATmega8L**

**Summary**

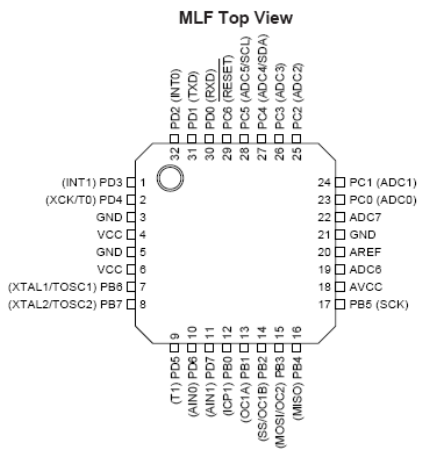
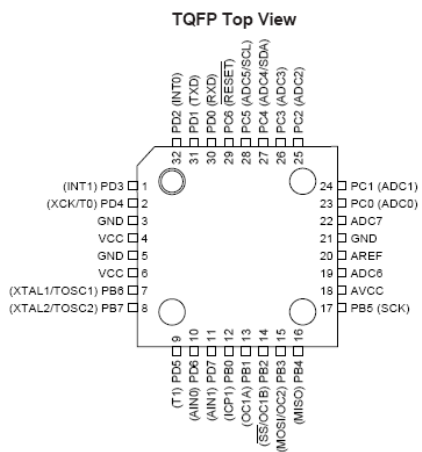
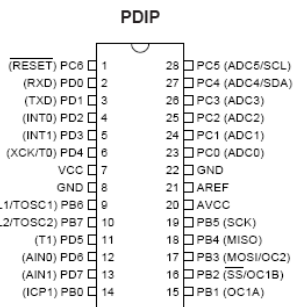
Rev. 2486MS-AVR-12/03



Note: This is a summary document. A complete document is available on our Web site at [www.atmel.com](http://www.atmel.com).



## Pin Configurations





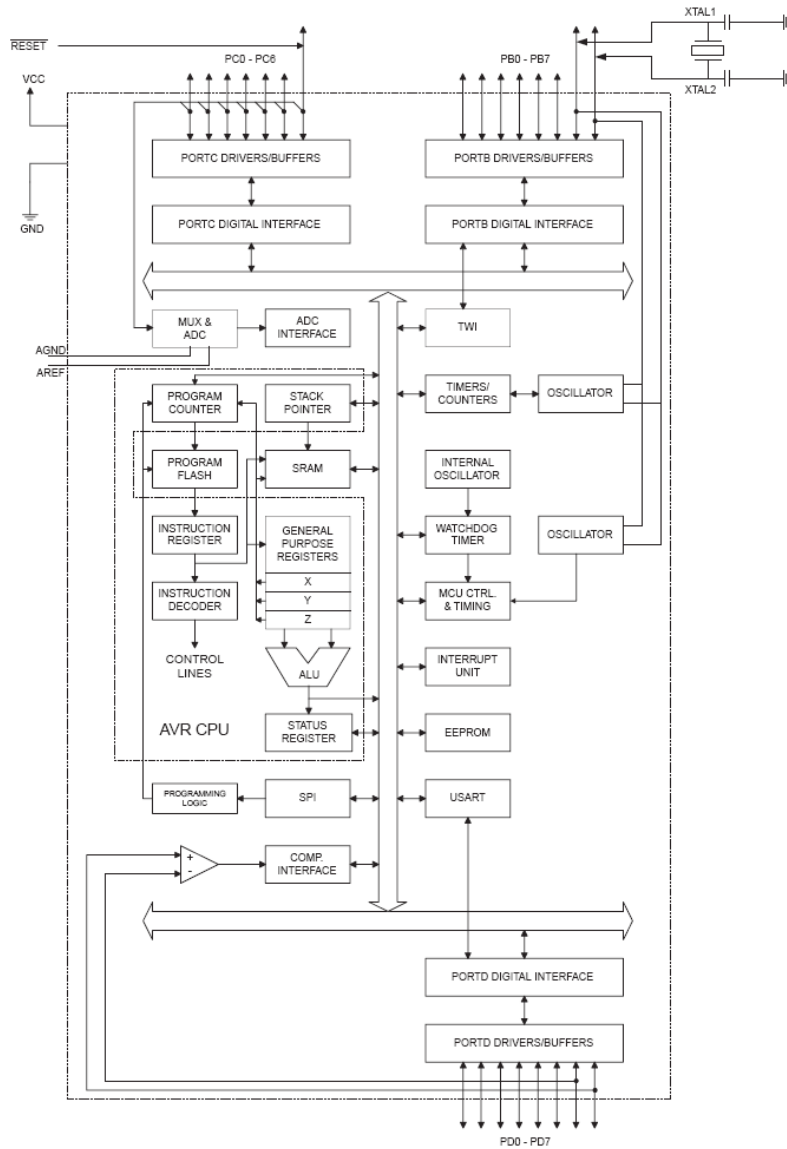
# ATmega8(L)

## Overview

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 1. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and MLF packages) where four (six) channels have 10-bit accuracy and two channels have 8-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 AVR is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

## Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.



<b>AVCC</b>	AVCC is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to $V_{CC}$ through a low-pass filter. Note that Port C (5..4) use digital supply voltage, $V_{CC}$ .
<b>AREF</b>	AREF is the analog reference pin for the A/D Converter.
<b>ADC7..6 (TQFP and MLF Package Only)</b>	In the TQFP and MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

# ATmega8(L)

## Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	9
0x3E (0x5E)	SPH	-	-	-	-	-	SP10	SP9	SP8	11
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	11
0x3C (0x5C)	Reserved									
0x3B (0x5B)	GICR	INT1	INT0	-	-	-	-	IVSEL	IVCE	47, 65
0x3A (0x5A)	GIFR	INTF1	INTF0	-	-	-	-	-	-	66
0x39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0	70, 100, 120
0x38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0	71, 101, 120
0x37 (0x57)	SPMCR	SPMIE	RWWSB	-	RWWSRE	BLSE	PGWRT	PGERS	SPMEN	210
0x36 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	168
0x35 (0x55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	31, 64
0x34 (0x54)	MCUCSR	-	-	-	-	WDRF	BORF	EXTRF	PORF	39
0x33 (0x53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	70
0x32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								70
0x31 (0x51)	OSCCAL	Oscillator Calibration Register								29
0x30 (0x50)	SFIOR	-	-	-	-	ACME	PUD	PSR2	PSR10	56, 73, 121, 190
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	95
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	96
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High byte								99
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low byte								99
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High byte								99
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low byte								99
0x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High byte								99
0x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low byte								99
0x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High byte								100
0x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low byte								100
0x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	115
0x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								117
0x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								117
0x22 (0x42)	ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	117
0x21 (0x41)	WDTCSR	-	-	-	WDCE	WDE	WDP2	WDP1	WDP0	41
0x20 <sup>(*)</sup> (0x40 <sup>(*)</sup> )	UBRRH	URSEL	-	-	-	-	UBRR[11:8]			155
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCS21	UCS20	UCPOL	153
0x1F (0x3F)	EEARH	-	-	-	-	-	-	-	EEAR8	18
0x1E (0x3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	18
0x1D (0x3D)	EEDR	EEPROM Data Register								18
0x1C (0x3C)	EEDR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	18
0x1B (0x3B)	Reserved									
0x1A (0x3A)	Reserved									
0x19 (0x39)	Reserved									
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	63
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	63
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	63
0x15 (0x35)	PORTC	-	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	63
0x14 (0x34)	DDRC	-	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	63
0x13 (0x33)	PINC	-	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	63
0x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	63
0x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	63
0x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	63
0x0F (0x2F)	SPDR	SPI Data Register								128
0x0E (0x2E)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	128
0x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	126
0x0C (0x2C)	UDR	USART I/O Data Register								150
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	151
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCS22	RXB8	TXB8	152
0x09 (0x29)	UBRRL	USART Baud Rate Register Low byte								155
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	191
0x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0	202
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	204
0x05 (0x25)	ADCH	ADC Data Register High byte								205
0x04 (0x24)	ADCL	ADC Data Register Low byte								205
0x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								170
0x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	170



## Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x01 (0x21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	170
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								168

- Notes:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
  2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

## Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd,K	Add Immediate to Word	$RdH:RdL \leftarrow RdH:RdL + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd,K	Subtract Immediate from Word	$RdH:RdL \leftarrow RdH:RdL - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \& Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \& K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \& (\text{0xFF} - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \& Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll \uparrow$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll \uparrow$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll \uparrow$	Z,C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	If $(Rd = Rr) PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	If $(Rr)(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	If $(Rr)(b)=1) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	If $(P)(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	If $(P)(b)=1) PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	If $(SREG)(s) = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	If $(SREG)(s) = 0) then PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	If $(Z = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	If $(Z = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	If $(C = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	If $(C = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	If $(C = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	If $(C = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	If $(N = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	If $(N = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	If $(N \oplus V = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	If $(N \oplus V = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	If $(H = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	If $(H = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	If $(T = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	If $(T = 0) then PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	If $(V = 1) then PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	If $(V = 0) then PC \leftarrow PC + k + 1$	None	1/2
Mnemonics	Operands	Description	Operation	Flags	#Clocks



## Instruction Set Summary (Continued)

BRIE	k	Branch if Interrupt Enabled	If (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	If (I = 0) then PC ← PC + k + 1	None	1 / 2
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBI	P, b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P, b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z, C, N, V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
<b>Mnemonics</b>	<b>Operands</b>	<b>Description</b>	<b>Operation</b>	<b>Flags</b>	<b>#Clocks</b>

## ATmega8(L)

### Instruction Set Summary (Continued)

CLT		Clear T In SREG	T ← 0	T	1
SEH		Set Half Carry Flag In SREG	H ← 1	H	1
CLH		Clear Half Carry Flag In SREG	H ← 0	H	1
<b>MCU CONTROL INSTRUCTIONS</b>					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1





## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5	ATmega8L-8AC ATmega8L-8PC ATmega8L-8MC	32A 28P3 32M1-A	Commercial (0°C to 70°C)
		ATmega8L-8AI ATmega8L-8PI ATmega8L-8MI	32A 28P3 32M1-A	Industrial (-40°C to 85°C)
16	4.5 - 5.5	ATmega8-16AC ATmega8-16PC ATmega8-16MC	32A 28P3 32M1-A	Commercial (0°C to 70°C)
		ATmega8-16AI ATmega8-16PI ATmega8-16MI	32A 28P3 32M1-A	Industrial (-40°C to 85°C)

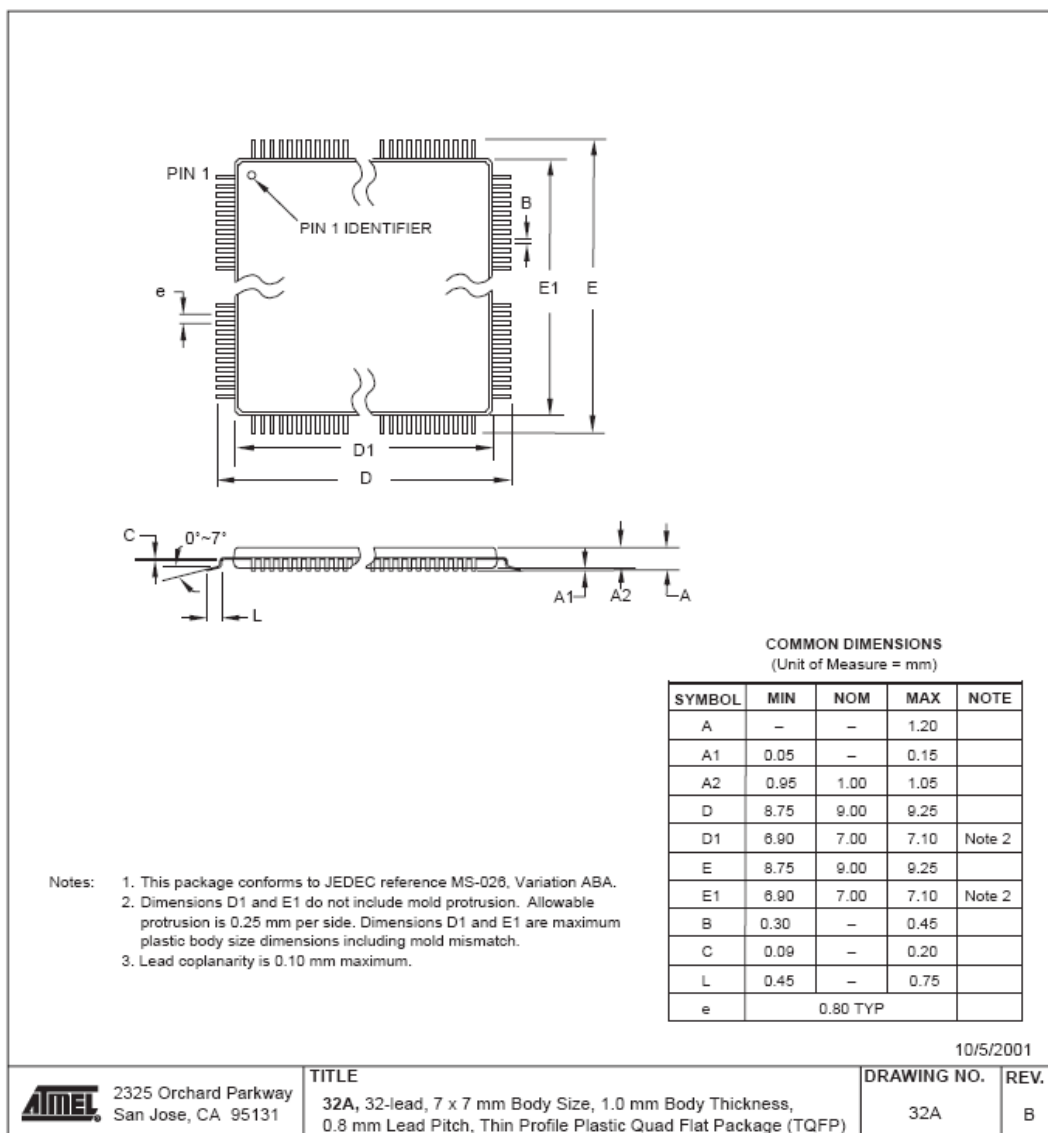
Note: This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
28P3	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
32M1-A	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF)

## ATmega8(L)

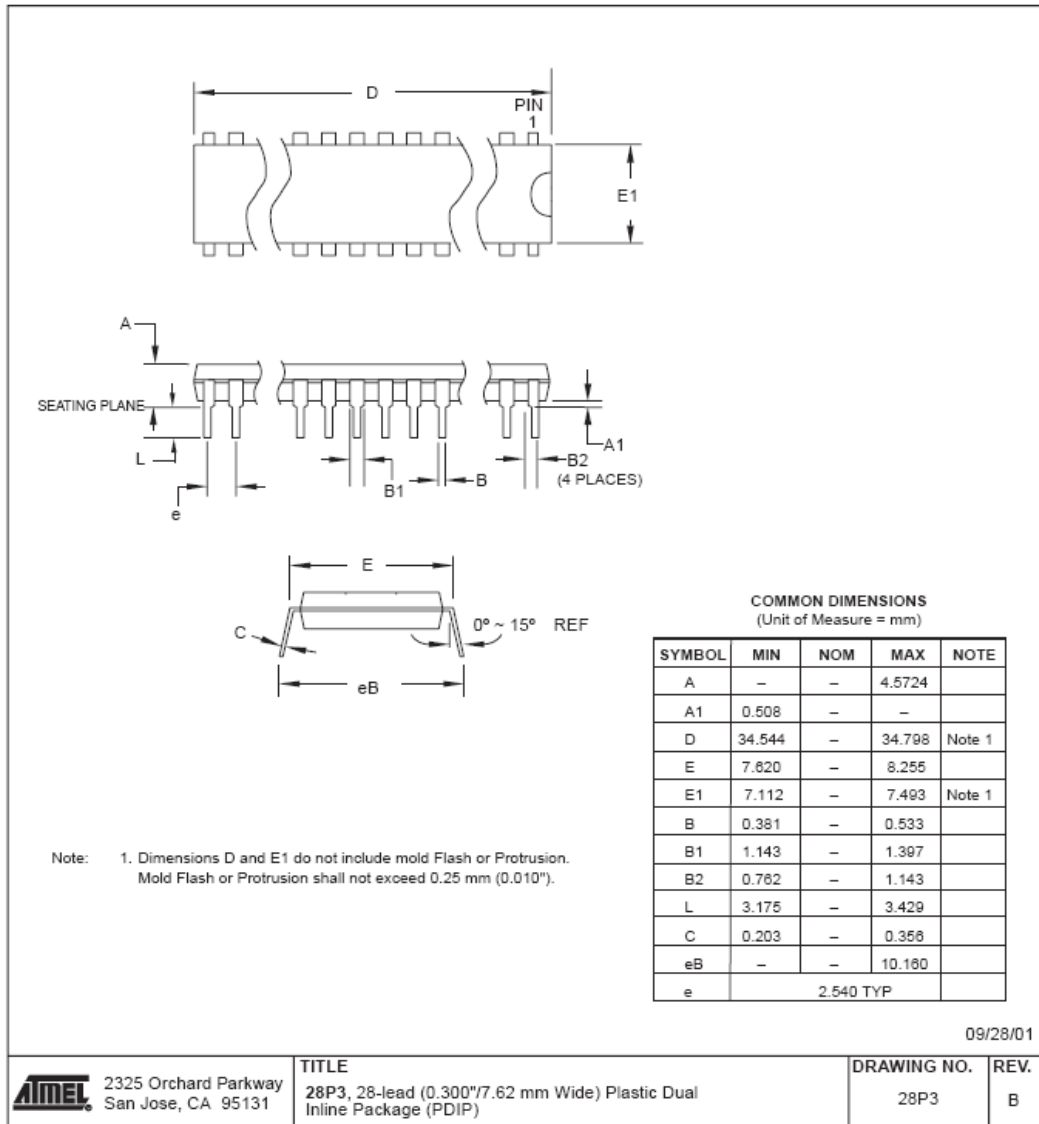
### Packaging Information

32A



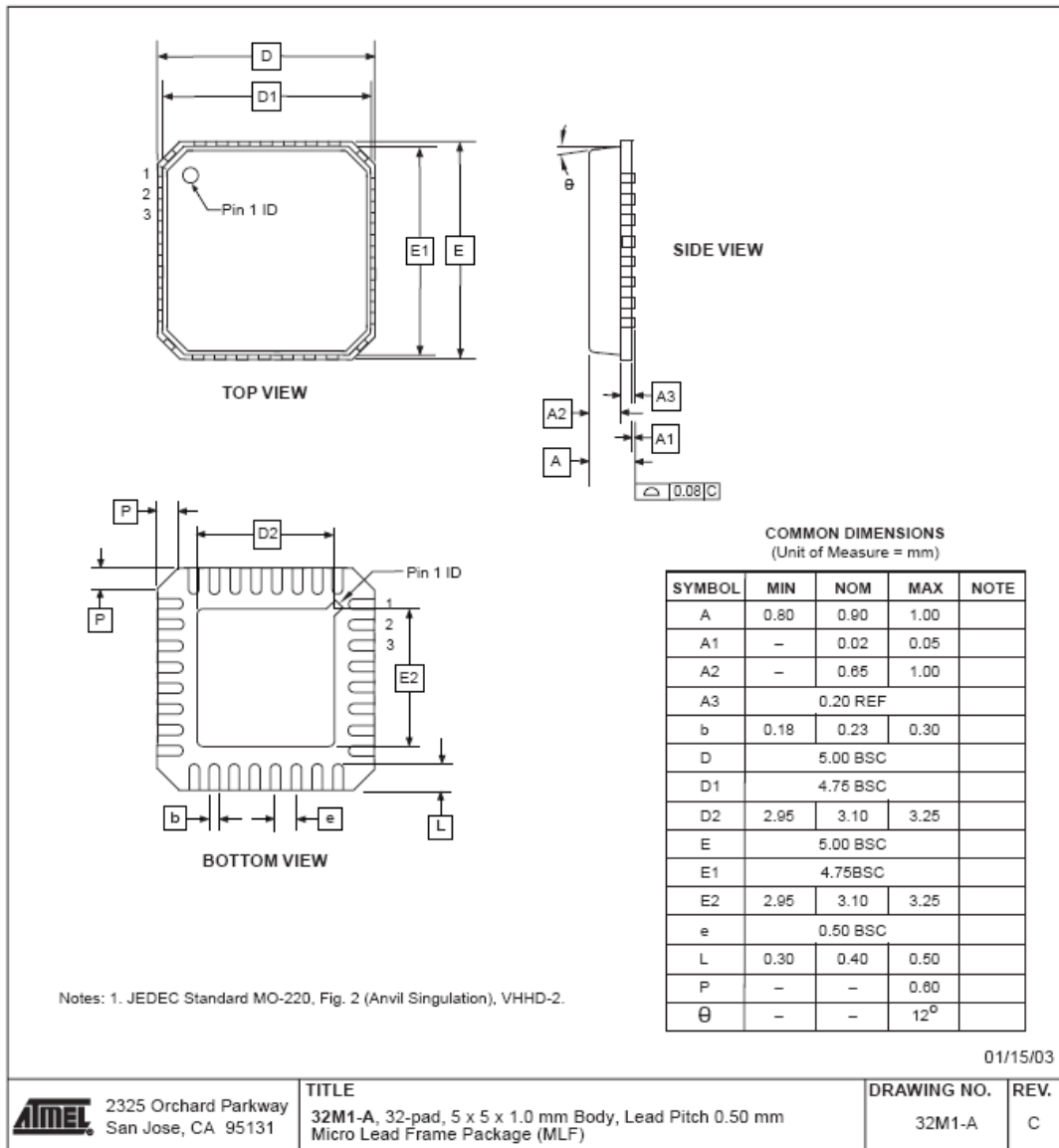


28P3



**ATmega8(L)**

**32M1-A**





## Erratas

### ATmega8 Rev. D, E, F, and G

The revision letter in this section refers to the revision of the ATmega8 device.

- CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2

#### 1. CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2

When the internal RC Oscillator is used as the main clock source, it is possible to run the Timer/Counter2 asynchronously by connecting a 32 KHz Oscillator between XTAL1/TOSC1 and XTAL2/TOSC2. But when the internal RC Oscillator is selected as the main clock source, the CKOPT Fuse does not control the internal capacitors on XTAL1/TOSC1 and XTAL2/TOSC2. As long as there are no capacitors connected to XTAL1/TOSC1 and XTAL2/TOSC2, safe operation of the Oscillator is not guaranteed.

##### Problem fix/Workaround

Use external capacitors in the range of 20 - 36 pF on XTAL1/TOSC1 and XTAL2/TOSC2. This will be fixed in ATmega8 Rev. G where the CKOPT Fuse will control internal capacitors also when internal RC Oscillator is selected as main clock source. For ATmega8 Rev. G, CKOPT = 0 (programmed) will enable the internal capacitors on XTAL1 and XTAL2. Customers who want compatibility between Rev. G and older revisions, must ensure that CKOPT is unprogrammed (CKOPT = 1).

# ANEXO II

**MOTOROLA**  
**SEMICONDUCTOR TECHNICAL DATA**

 Order this document  
 by MOC3020/D


## 6-Pin DIP Random-Phase Optoisolators Triac Driver Output (400 Volts Peak)

The MOC3020 Series consists of gallium arsenide infrared emitting diodes, optically coupled to a silicon bilateral switch.

- *To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.* They are designed for applications requiring isolated triac triggering.

**Recommended for 115/240 Vac(rms) Applications:**

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115 Vac Peripherals
- Motor Controls
- Static ac Power Switch
- Solid State Relays
- Incandescent Lamp Dimmers

**MAXIMUM RATINGS** ( $T_A = 25^\circ\text{C}$  unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

**INFRARED EMITTING DIODE**

Reverse Voltage	$V_R$	3	Volts
Forward Current — Continuous	$I_F$	60	mA
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Negligible Power in Triac Driver Derate above $25^\circ\text{C}$	$P_D$	100	mW
		1.33	mW/ $^\circ\text{C}$

**OUTPUT DRIVER**

Off-State Output Terminal Voltage	$V_{DRM}$	400	Volts
Peak Repetitive Surge Current ( $PW = 1\text{ ms}$ , 120 pps)	$I_{TSM}$	1	A
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	300	mW
		4	mW/ $^\circ\text{C}$

**TOTAL DEVICE**

Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 Second Duration)	$V_{ISO}$	7500	Vac(pk)
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	330	mW
		4.4	mW/ $^\circ\text{C}$
Junction Temperature Range	$T_J$	-40 to +100	$^\circ\text{C}$
Ambient Operating Temperature Range <sup>(2)</sup>	$T_A$	-40 to +85	$^\circ\text{C}$
Storage Temperature Range <sup>(2)</sup>	$T_{stg}$	-40 to +150	$^\circ\text{C}$
Soldering Temperature (10 s)	$T_L$	260	$^\circ\text{C}$

1. Isolation surge voltage,  $V_{ISO}$ , is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value. GlobalOptoisolator is a trademark of Motorola, Inc.

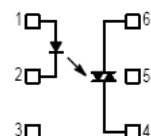
REV 1

© Motorola, Inc. 1995

**MOC3021**  
[IFT = 15 mA Max]  
**MOC3022**  
[IFT = 10 mA Max]  
**MOC3023\***  
[IFT = 5 mA Max]

\*Motorola Preferred Device

**STYLE 6 PLASTIC**

**STANDARD THRU HOLE  
CASE 730A-04**
**SCHEMATIC**


1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE  
DO NOT CONNECT
6. MAIN TERMINAL



**MOC3021 MOC3022 MOC3023****ELECTRICAL CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
<b>INPUT LED</b>					
Reverse Leakage Current ( $V_R = 3\text{ V}$ )	$I_R$	—	0.05	100	$\mu\text{A}$
Forward Voltage ( $I_F = 10\text{ mA}$ )	$V_F$	—	1.15	1.5	Volts
<b>OUTPUT DETECTOR (<math>I_F = 0</math> unless otherwise noted)</b>					
Peak Blocking Current, Either Direction (Rated $V_{DRM}^{(1)}$ )	$I_{DRM}$	—	10	100	nA
Peak On-State Voltage, Either Direction ( $I_{TM} = 100\text{ mA Peak}$ )	$V_{TM}$	—	1.8	3	Volts
Critical Rate of Rise of Off-State Voltage (Figure 7, Note 2)	$dv/dt$	—	10	—	$\text{V}/\mu\text{s}$
<b>COUPLED</b>					
LED Trigger Current, Current Required to Latch Output (Main Terminal Voltage = $3\text{ V}^{(3)}$ )	$I_{FT}$	—	8	15	mA
MOC3021	—	—	—	10	
MOC3022	—	—	—	5	
Holding Current, Either Direction	$I_H$	—	100	—	$\mu\text{A}$

1. Test voltage must be applied within  $dv/dt$  rating.
2. This is static  $dv/dt$ . See Figure 7 for test circuit. Commutating  $dv/dt$  is a function of the load-driving thyristor(s) only.
3. All devices are guaranteed to trigger at an  $I_F$  value less than or equal to max  $I_{FT}$ . Therefore, recommended operating  $I_F$  lies between max  $I_{FT}$  (15 mA for MOC3021, 10 mA for MOC3022, 5 mA for MOC3023) and absolute max  $I_F$  (60 mA).

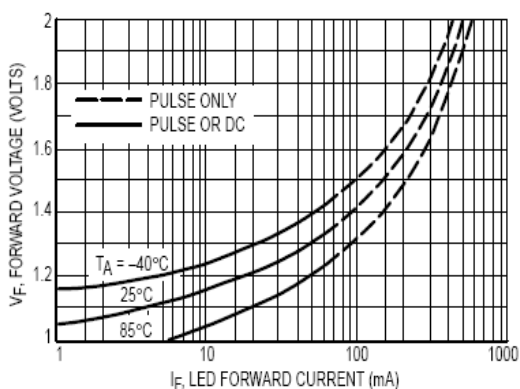
**TYPICAL ELECTRICAL CHARACTERISTICS** $T_A = 25^\circ\text{C}$ 

Figure 1. LED Forward Voltage versus Forward Current

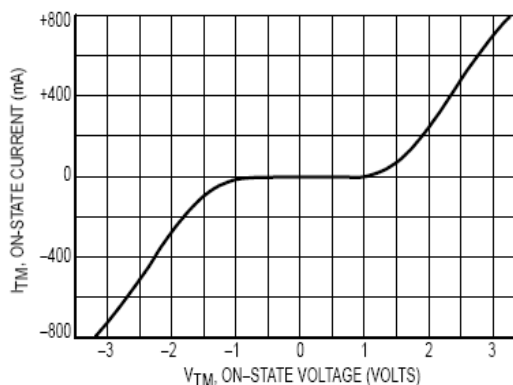
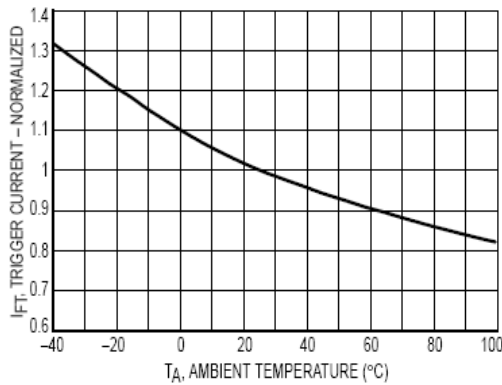


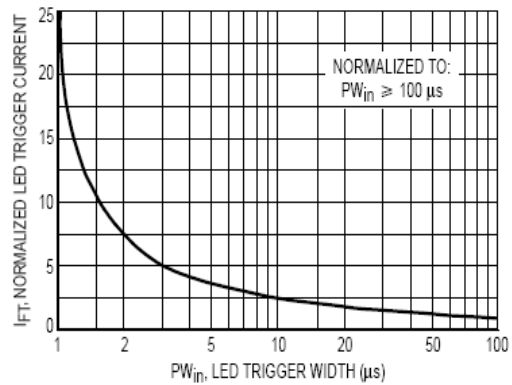
Figure 2. On-State Characteristics



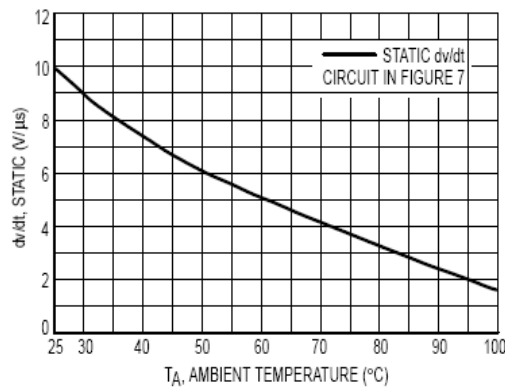
**MOC3021 MOC3022 MOC3023**



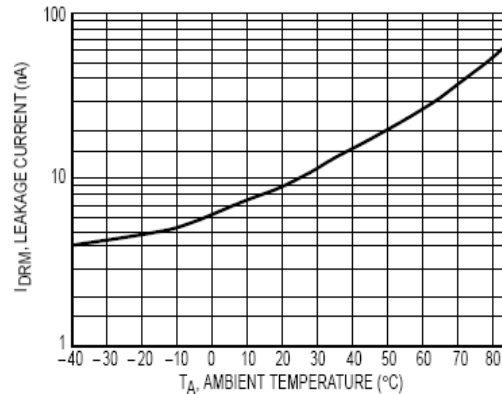
**Figure 3. Trigger Current versus Temperature**



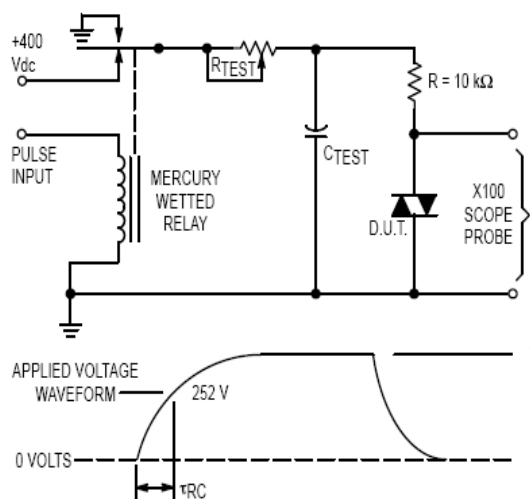
**Figure 4. LED Current Required to Trigger versus LED Pulse Width**



**Figure 5. dv/dt versus Temperature**



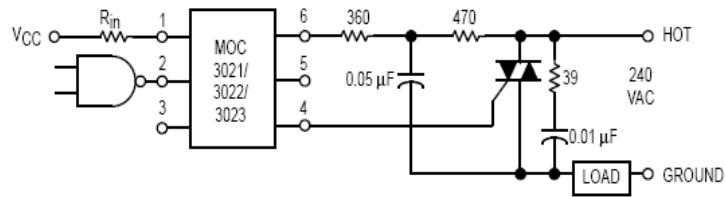
**Figure 6. Leakage Current, I<sub>DRM</sub> versus Temperature**



1. The mercury wetted relay provides a high speed repeated pulse to the D.U.T.
2. 100x scope probes are used, to allow high speeds and voltages.
3. The worst-case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable R<sub>TEST</sub> allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering. τ<sub>RC</sub> is measured at this point and recorded.

**Figure 7. Static dv/dt Test Circuit**

$$dv/dt = \frac{0.63 V_{max}}{\tau_{RC}} = \frac{252}{\tau_{RC}}$$

**MOC3021 MOC3022 MOC3023**

\* This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only.

Additional information on the use of optically coupled triac drivers is available in Application Note AN-780A.

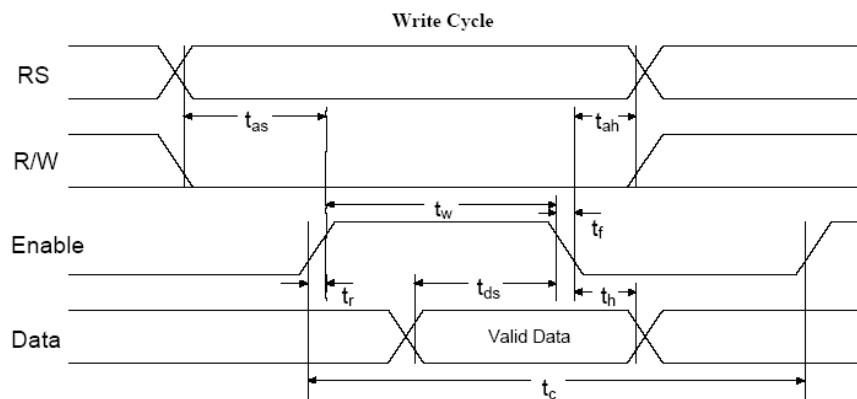
In this circuit the "hot" side of the line is switched and the load connected to the cold or ground side.

The 39 ohm resistor and 0.01  $\mu\text{F}$  capacitor are for snubbing of the triac, and the 470 ohm resistor and 0.05  $\mu\text{F}$  capacitor are for snubbing the coupler. These components may or may not be necessary depending upon the particular triac and load used.

**Figure 8. Typical Application Circuit**

# ANEXO III

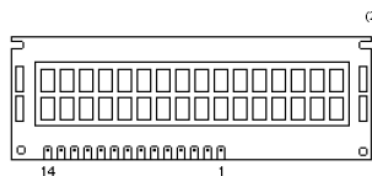
Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
NOP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero, returns shifted display to original position. DDRAM contents remains unchanged.	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction, and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N,M) and voltage generator control (G).	3
Set CGRAM Addr	0	0	0	1	Character Generator RAM					Sets CGRAM Address	3	
Set DDRAM Addr	0	0	1	Display Data RAM Address					Sets DDRAM Address	3		
Busy Flag & Addr	0	1	BF	Address Counter					Reads Busy Flag & Address Counter	0		
Read Data	1	0	Read Data					Reads data from CGRAM or DDRAM	3			
Write Data	1	1	Write Data					Writes data from CGRAM or DDRAM	3			



Parameter	Symbol	Min <sup>(1)</sup>	Typ <sup>(1)</sup>	Max <sup>(1)</sup>	Unit
Enable Cycle Time	$t_c$	500	-	-	ns
Enable Pulse Width (High)	$t_w$	230	-	-	ns
Enable Rise/Fall Time	$t_r, t_f$	-	-	20	ns
Address Setup Time	$t_{as}$	40	-	-	ns
Address Hold Time	$t_{ah}$	10	-	-	ns
Data Setup Time	$t_{ds}$	80	-	-	ns
Data Hold Time	$t_h$	10	-	-	ns

Note <sup>1</sup> The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

Note <sup>2</sup> A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x 16 character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x 20 character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD Module.



Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

# ANEXO IV



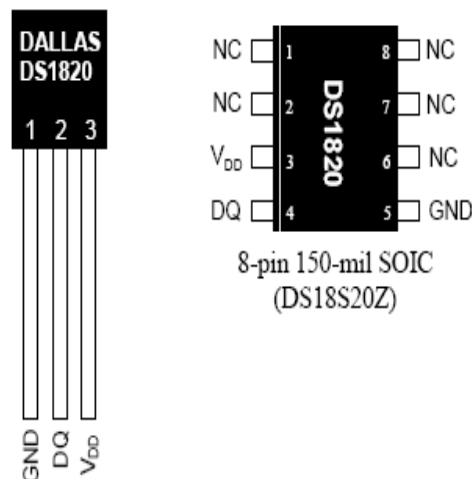
## DS18S20 High Precision 1-Wire<sup>®</sup> Digital Thermometer

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

### FEATURES

- Unique 1-wire interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an on-board ROM
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$ )
- $\pm 0.5^{\circ}\text{C}$  accuracy from  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User-definable nonvolatile alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

### PIN ASSIGNMENT



### PIN DESCRIPTION

GND - Ground  
DQ - Data In/Out  
V<sub>DD</sub> - Power Supply Voltage  
NC - No Connect

### DESCRIPTION

The DS18S20 Digital Thermometer provides 9-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  and is accurate to  $\pm 0.5^{\circ}\text{C}$  over the range of  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ . In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

**DETAILED PIN DESCRIPTIONS Table 1**

8-PIN SOIC*	TO-92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	<b>Data Input/Output pin.</b> Open-drain 1-wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	3	V <sub>DD</sub>	<b>Optional V<sub>DD</sub> pin.</b> V <sub>DD</sub> must be grounded for operation in parasite power mode.

\*All pins not specified in this table are "No Connect" pins.

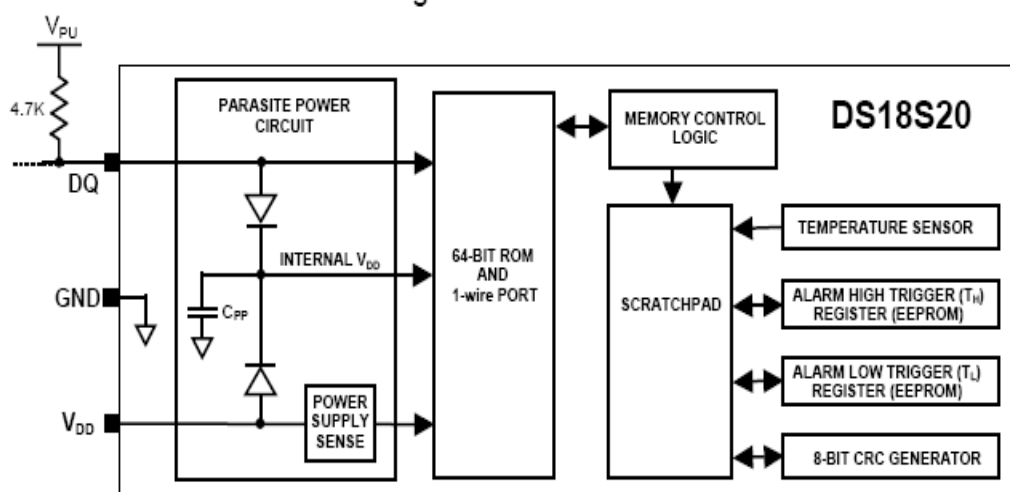
## OVERVIEW

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T<sub>H</sub> and T<sub>L</sub>). The T<sub>H</sub> and T<sub>L</sub> registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Dallas' exclusive 1-wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the 1-WIRE BUS SYSTEM section of this datasheet.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C<sub>PP</sub>), which then supplies power to the device when the bus is low. This method of deriving power from the 1-wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V<sub>DD</sub>.

**DS18S20 BLOCK DIAGRAM Figure 1**



## OPERATION – MEASURING TEMPERATURE

The core functionality of the DS18S20 is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to 0.5°C steps. The DS18S20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its idle state. If the DS18S20 is powered by an external supply, the master can issue “read time slots” (see the 1-WIRE BUS SYSTEM section) after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18S20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the POWERING THE DS18S20 section of this datasheet.

The DS18S20 output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two’s complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 2 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT REMAIN and COUNT PER °C registers in the scratchpad. Note that the COUNT PER °C register is hard-wired to 16 (10h). After reading the scratchpad, the TEMP\_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 2). The extended resolution temperature can then be calculated using the following equation:

$$TEMPERATURE = TEMP\_READ - 0.25 + \frac{COUNT\_PER\_C - COUNT\_REMAIN}{COUNT\_PER\_C}$$

Additional information about high-resolution temperature calculations can be found in Application Note 105: “High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors”.

### TEMPERATURE REGISTER FORMAT Figure 2

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	S	S	S

### TEMPERATURE/DATA RELATIONSHIP Table 2

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+85.0°C*	0000 0000 1010 1010	00AAh
+25.0°C	0000 0000 0011 0010	0032h
+0.5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1111	FFFFh
-25.0°C	1111 1111 1100 1110	FFCEh
-55.0°C	1111 1111 1001 0010	FF92h

\*The power-on reset value of the temperature register is +85°C



## OPERATION – ALARM SIGNALING

After the DS18S20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte  $T_H$  and  $T_L$  registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . The  $T_H$  and  $T_L$  registers are nonvolatile (EEPROM) so they will retain data when the device is powered down.  $T_H$  and  $T_L$  can be accessed through bytes 2 and 3 of the scratchpad as explained in the MEMORY section of this datasheet.

### $T_H$ AND $T_L$ REGISTER FORMAT Figure 3

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	$2^6$	$2^5$	$2^5$	$2^5$	$2^2$	$2^1$	$2^0$

Only bits 8 through 1 of the temperature register are used in the  $T_H$  and  $T_L$  comparison since  $T_H$  and  $T_L$  are 8-bit registers. If the result of a temperature measurement is higher than  $T_H$  or lower than  $T_L$ , an alarm condition exists and an alarm flag is set inside the DS18S20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18S20s on the bus by issuing an Alarm Search [ECh] command. Any DS18S20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18S20s have experienced an alarm condition. If an alarm condition exists and the  $T_H$  or  $T_L$  settings have changed, another temperature conversion should be done to validate the alarm condition.

## POWERING THE DS18S20

The DS18S20 can be powered by an external supply on the  $V_{DD}$  pin, or it can operate in “parasite power” mode, which allows the DS18S20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18S20's parasite-power control circuitry, which “steals” power from the 1-wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18S20 while the bus is high, and some of the charge is stored on the parasite power capacitor ( $C_{pp}$ ) to provide power when the bus is low. When the DS18S20 is used in parasite power mode, the  $V_{DD}$  pin must be connected to ground.

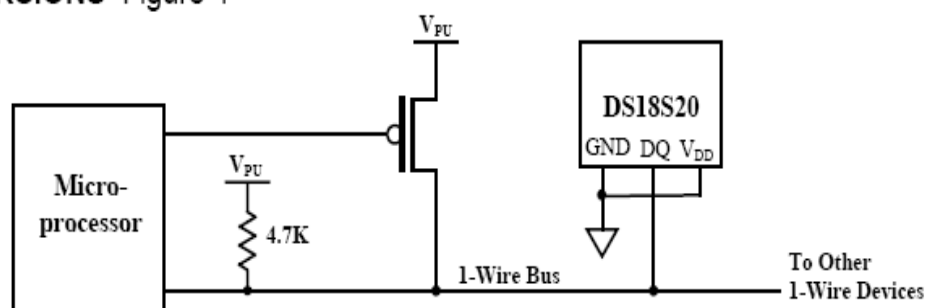
In parasite power mode, the 1-wire bus and  $C_{pp}$  can provide sufficient current to the DS18S20 for most operations as long as the specified timing and voltage requirements are met (refer to the DC ELECTRICAL CHARACTERISTICS and the AC ELECTRICAL CHARACTERISTICS sections of this data sheet). However, when the DS18S20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5 mA. This current can cause an unacceptable voltage drop across the weak 1-wire pullup resistor and is more current than can be supplied by  $C_{pp}$ . To assure that the DS18S20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-wire bus must be switched to the strong pullup within 10  $\mu$ s (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion ( $t_{conv}$ ) or data transfer ( $t_{wr} = 10$  ms). No other activity can take place on the 1-wire bus while the pullup is enabled.

The DS18S20 can also be powered by the conventional method of connecting an external power supply to the  $V_{DD}$  pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-wire bus is free to carry other traffic during the temperature conversion time.

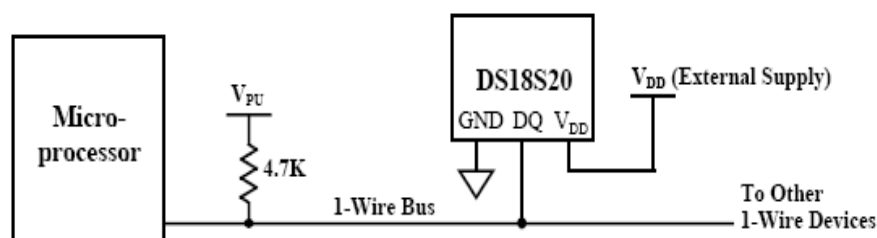
The use of parasite power is not recommended for temperatures above 100°C since the DS18S20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18S20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18S20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read time slot”. During the read time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-wire bus during temperature conversions.

### SUPPLYING THE PARASITE-POWERED DS18S20 DURING TEMPERATURE CONVERSIONS Figure 4



### POWERING THE DS18S20 WITH AN EXTERNAL SUPPLY Figure 5



### 64-BIT LASERED ROM CODE

Each DS18S20 contains a unique 64-bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20's 1-wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the CRC GENERATION section. The 64-bit ROM code and associated ROM function control logic allow the DS18S20 to operate as a 1-wire device using the protocol detailed in the 1-WIRE BUS SYSTEM section of this datasheet.

### 64-BIT LASERED ROM CODE Figure 6

8-BIT CRC		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

## MEMORY

The DS18S20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers ( $T_H$  and  $T_L$ ). Note that if the DS18S20 alarm function is not used, the  $T_H$  and  $T_L$  registers can serve as general-purpose memory. All memory commands are described in detail in the DS18S20 FUNCTION COMMANDS section.

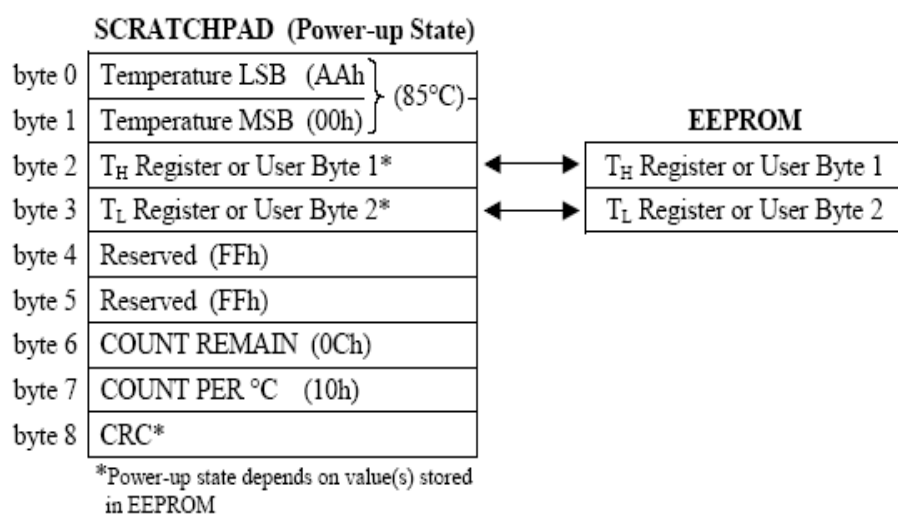
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to  $T_H$  and  $T_L$  registers. Bytes 4 and 5 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read. Bytes 6 and 7 contain the COUNT REMAIN and COUNT PER °C registers, which can be used to calculate extended resolution results as explained in the OPERATION – MEASURING TEMPERATURE section.

Byte 8 of the scratchpad is read-only and contains the cyclic redundancy check (CRC) code for bytes 0 through 7 of the scratchpad. The DS18S20 generates this CRC using the method described in the CRC GENERATION section.

Data is written to bytes 2 and 3 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18S20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-wire bus starting with the least significant bit of byte 0. To transfer the  $T_H$  and  $T_L$  data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E<sup>2</sup> [B8h] command. The master can issue “read time slots” (see the 1-WIRE BUS SYSTEM section) following the Recall E<sup>2</sup> command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

## DS18S20 MEMORY MAP



## CRC GENERATION

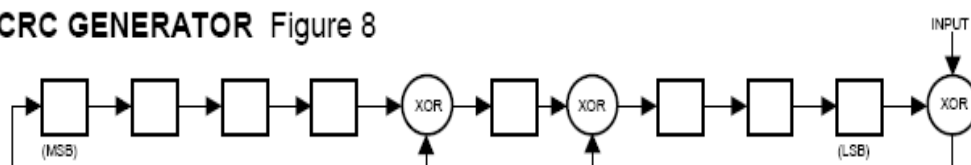
CRC bytes are provided as part of the DS18S20's 64-bit ROM code and in the 9<sup>th</sup> byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18S20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18S20 that prevents a command sequence from proceeding if the DS18S20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18S20 using the polynomial generator shown in Figure 8. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56<sup>th</sup> bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18S20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Dallas 1-wire cyclic redundancy check is available in Application Note 27 entitled "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products."

**CRC GENERATOR** Figure 8



## 1-WIRE BUS SYSTEM

The 1-wire bus system uses a single bus master to control one or more slave devices. The DS18S20 is always a slave. When there is only one slave on the bus, the system is referred to as a “single-drop” system; the system is “multi-drop” if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-wire bus.

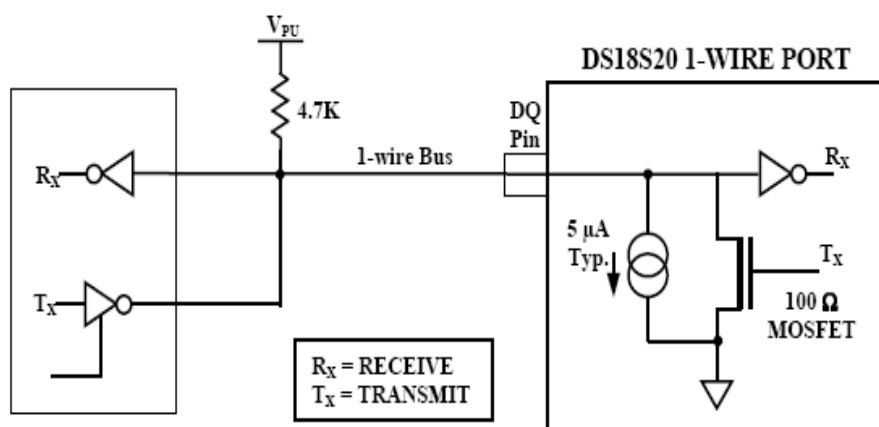
The following discussion of the 1-wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-wire signaling (signal types and timing).

## HARDWARE CONFIGURATION

The 1-wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-wire port of the DS18S20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 9.

The 1-wire bus requires an external pullup resistor of approximately 5 k $\Omega$ ; thus, the idle state for the 1-wire bus is high. If for any reason a transaction needs to be suspended, the bus **MUST** be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480  $\mu$ s, all components on the bus will be reset.

## HARDWARE CONFIGURATION Figure 9



## TRANSACTION SEQUENCE

The transaction sequence for accessing the DS18S20 is as follows:

- Step 1. Initialization
- Step 2. ROM Command (followed by any required data exchange)
- Step 3. DS18S20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18S20 is accessed, as the DS18S20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

## INITIALIZATION

All transactions on the 1-wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18S20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the 1-WIRE SIGNALING section.

## ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18S20 function command. A flowchart for operation of the ROM commands is shown in Figure 14.

### SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the iButton Book of Standards at [www.ibutton.com/ibuttons/standard.pdf](http://www.ibutton.com/ibuttons/standard.pdf). After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

### READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

### MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multi-drop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

### SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18S20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command. Note, however, that the Skip ROM command can only be followed by the Read Scratchpad [BEh] command when there is one slave on the bus. This sequence saves time by allowing the master to read from the device without sending its 64-bit ROM code. This sequence will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

### ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18S20s experienced an alarm condition during the most recent temperature conversion. After

every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. Refer to the OPERATION – ALARM SIGNALING section for an explanation of alarm flag operation.

## DS18S20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18S20 with which it wishes to communicate, the master can issue one of the DS18S20 function commands. These commands allow the master to write to and read from the DS18S20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18S20 function commands, which are described below, are summarized in Table 4 and illustrated by the flowchart in Figure 15.

### CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10  $\mu$ s (max) after this command is issued the master must enable a strong pullup on the 1-wire bus for the duration of the conversion ( $t_{conv}$ ) as described in the POWERING THE DS18S20 section. If the DS18S20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

### WRITE SCRATCHPAD [4Eh]

This command allows the master to write 2 bytes of data to the DS18S20's scratchpad. The first byte is written into the  $T_H$  register (byte 2 of the scratchpad), and the second byte is written into the  $T_L$  register (byte 3 of the scratchpad). Data must be transmitted least significant bit first. Both bytes MUST be written before the master issues a reset, or the data may be corrupted.

### READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9<sup>th</sup> byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

### COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad  $T_H$  and  $T_L$  registers (bytes 2 and 3) to EEPROM. If the device is being used in parasite power mode, within 10  $\mu$ s (max) after this command is issued the master must enable a strong pullup on the 1-wire bus for at least 10 ms as described in the POWERING THE DS18S20 section.

### RECALL E<sup>2</sup> [B8h]

This command recalls the alarm trigger values ( $T_H$  and  $T_L$ ) from EEPROM and places the data in bytes 2 and 3, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E<sup>2</sup> command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

### READ POWER SUPPLY [B4h]

The master device issues this command followed by a read time slot to determine if any DS18S20s on the bus are using parasite power. During the read time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. Refer to the POWERING THE DS18S20 section for usage information for this command.

**DS18S20 FUNCTION COMMAND SET Table 4**

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued	Notes
<b>TEMPERATURE CONVERSION COMMANDS</b>				
Convert T	Initiates temperature conversion.	44h	DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s).	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18S20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2 and 3 ( $T_H$ and $T_L$ ).	4Eh	Master transmits 2 data bytes to DS18S20.	3
Copy Scratchpad	Copies $T_H$ and $T_L$ data from the scratchpad to EEPROM.	48h	None	1
Recall E <sup>2</sup>	Recalls $T_H$ and $T_L$ data from EEPROM to the scratchpad.	B8h	DS18S20 transmits recall status to master.	
Read Power Supply	Signals DS18S20 power supply mode to the master.	B4h	DS18S20 transmits supply status to master.	

**NOTES:**

1. For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
2. The master can interrupt the transmission of data at any time by issuing a reset.
3. Both bytes must be written before a reset is issued.



## 1-WIRE SIGNALING

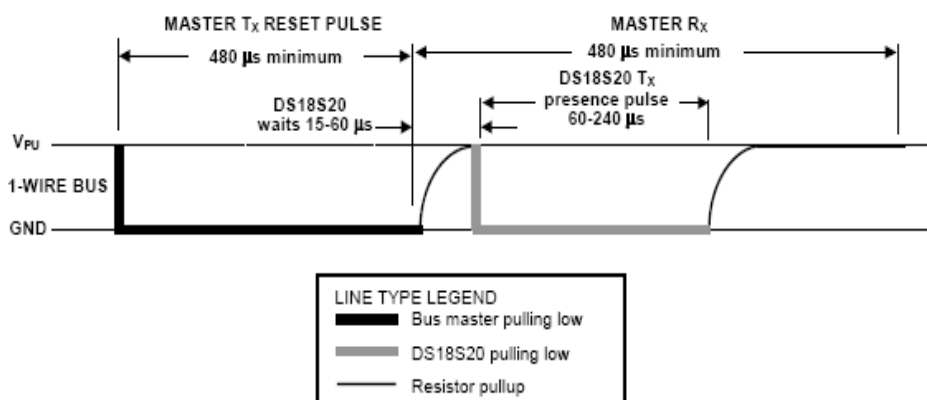
The DS18S20 uses a strict 1-wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

### INITIALIZATION PROCEDURE: RESET AND PRESENCE PULSES

All communication with the DS18S20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18S20. This is illustrated in Figure 10. When the DS18S20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits ( $T_x$ ) the reset pulse by pulling the 1-wire bus low for a minimum of 480  $\mu\text{s}$ . The bus master then releases the bus and goes into receive mode ( $R_x$ ). When the bus is released, the 5k pullup resistor pulls the 1-wire bus high. When the DS18S20 detects this rising edge, it waits 15–60  $\mu\text{s}$  and then transmits a presence pulse by pulling the 1-wire bus low for 60–240  $\mu\text{s}$ .

### INITIALIZATION TIMING Figure 10



### READ/WRITE TIME SLOTS

The bus master writes data to the DS18S20 during write time slots and reads data from the DS18S20 during read time slots. One bit of data is transmitted over the 1-wire bus per time slot.

### WRITE TIME SLOTS

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18S20 and a Write 0 time slot to write a logic 0 to the DS18S20. All write time slots must be a minimum of 60  $\mu\text{s}$  in duration with a minimum of a 1  $\mu\text{s}$  recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-wire bus low (see Figure 11).

To generate a Write 1 time slot, after pulling the 1-wire bus low, the bus master must release the 1-wire bus within 15  $\mu\text{s}$ . When the bus is released, the 5k pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60  $\mu\text{s}$ ). The DS18S20 samples the 1-wire bus during a window that lasts from 15  $\mu\text{s}$  to 60  $\mu\text{s}$  after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18S20. If the line is low, a 0 is written to the DS18S20.

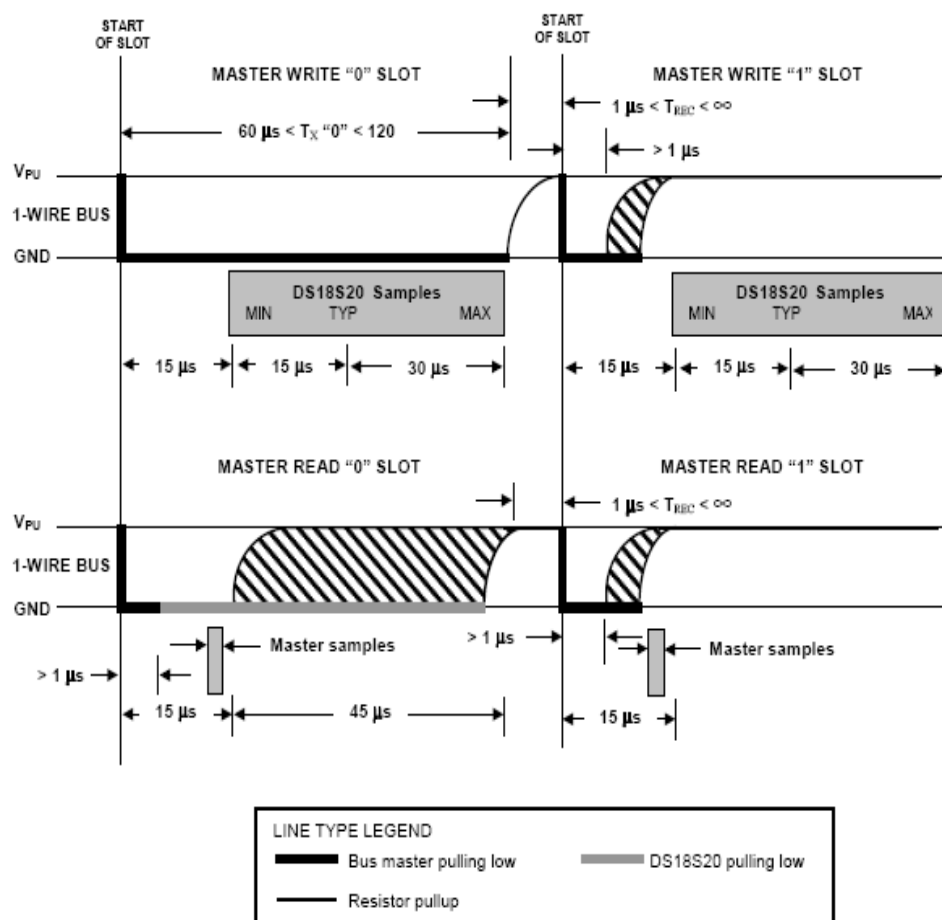
## READ TIME SLOTS

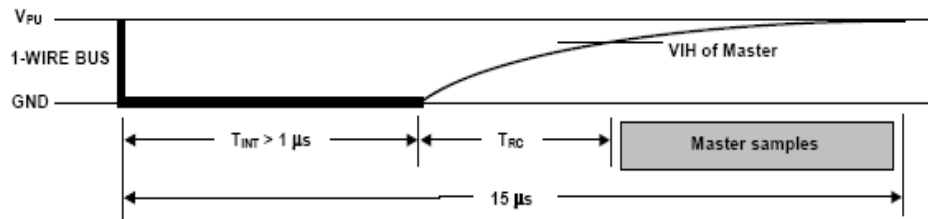
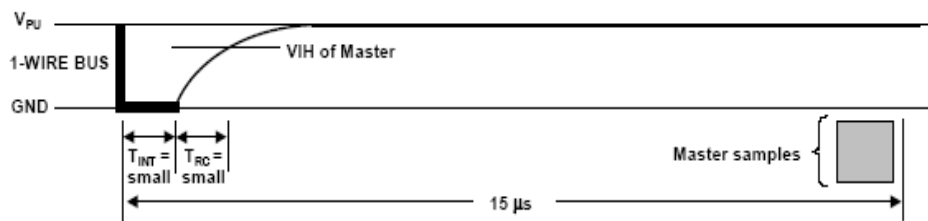
The DS18S20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18S20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E<sup>2</sup> [B8h] commands to find out the status of the operation as explained in the DS18S20 FUNCTION COMMAND section.

All read time slots must be a minimum of 60  $\mu\text{s}$  in duration with a minimum of a 1  $\mu\text{s}$  recovery time between slots. A read time slot is initiated by the master device pulling the 1-wire bus low for a minimum of 1  $\mu\text{s}$  and then releasing the bus (see Figure 11). After the master initiates the read time slot, the DS18S20 will begin transmitting a 1 or 0 on bus. The DS18S20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18S20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output data from the DS18S20 is valid for 15  $\mu\text{s}$  after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15  $\mu\text{s}$  from the start of the slot.

Figure 12 illustrates that the sum of  $T_{\text{INIT}}$ ,  $T_{\text{RC}}$ , and  $T_{\text{SAMPLE}}$  must be less than 15  $\mu\text{s}$  for a read time slot. Figure 13 shows that system timing margin is maximized by keeping  $T_{\text{INIT}}$  and  $T_{\text{RC}}$  as short as possible and by locating the master sample time during read time slots towards the end of the 15  $\mu\text{s}$  period.

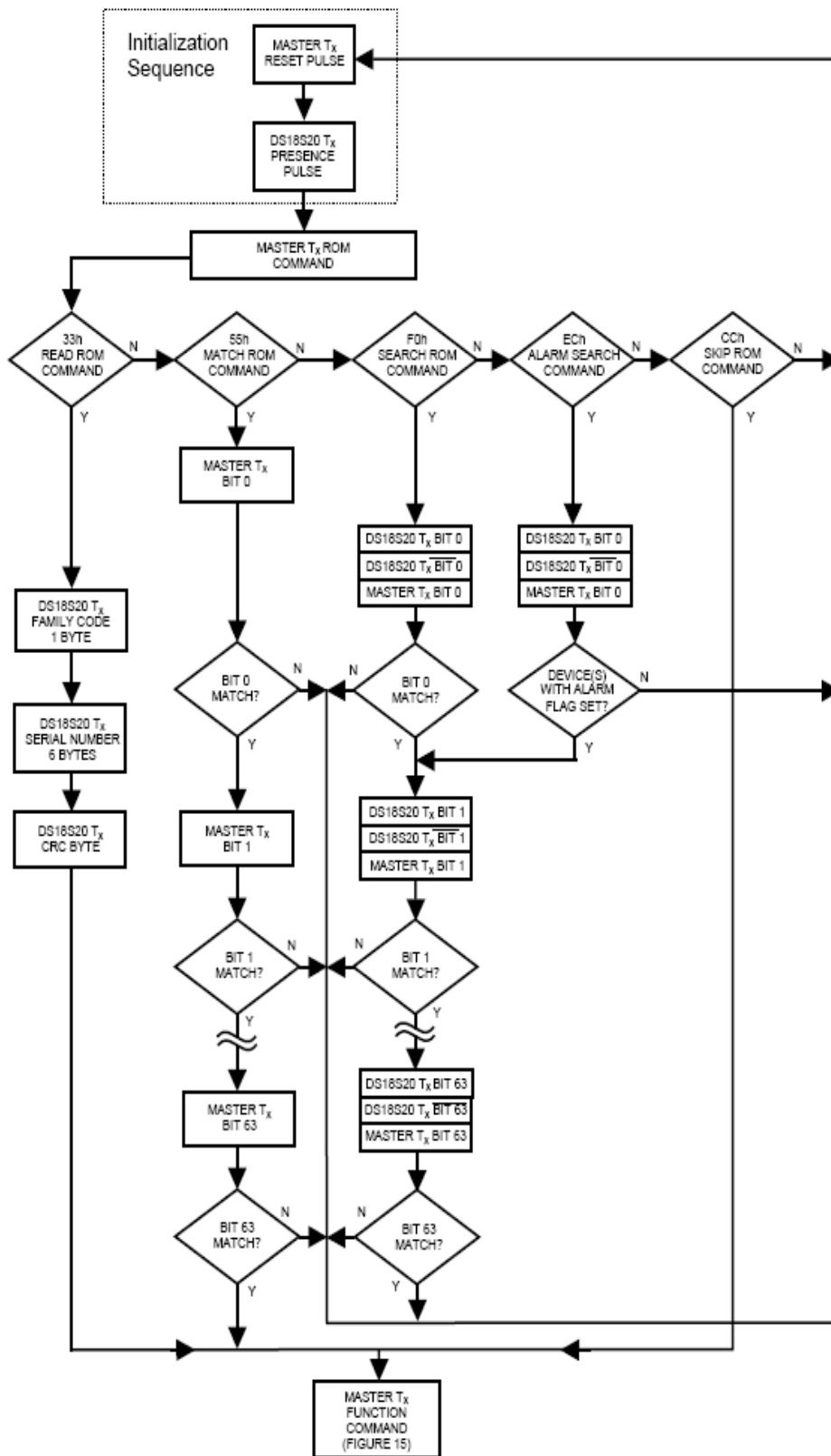
## READ/WRITE TIME SLOT TIMING DIAGRAM Figure 11



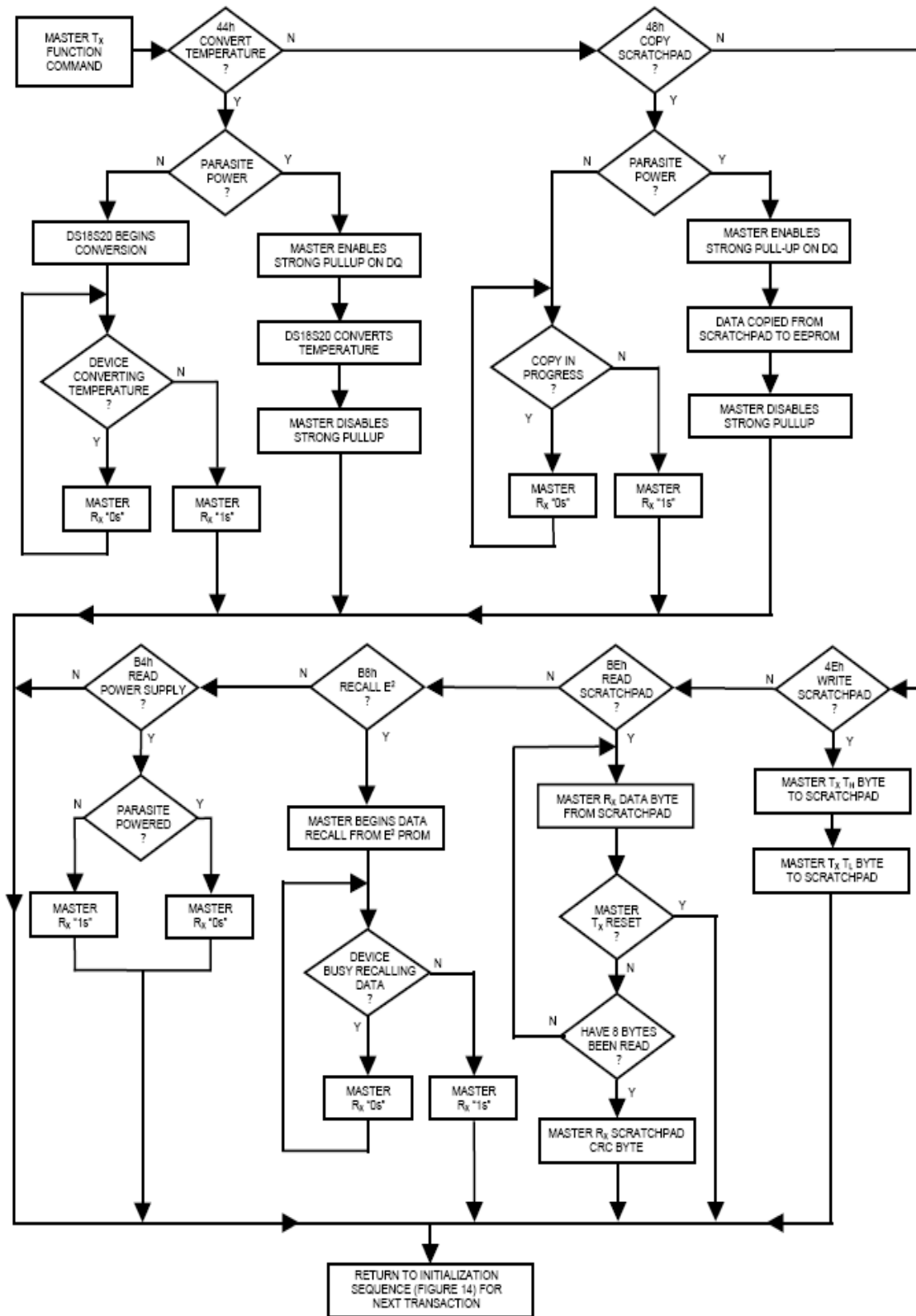
**DETAILED MASTER READ 1 TIMING** Figure 12**RECOMMENDED MASTER READ 1 TIMING** Figure 13

LINE TYPE LEGEND  
 — Bus master pulling low  
 — Resistor pullup

ROM COMMANDS FLOW CHART Figure 14



DS18S20 FUNCTION COMMANDS FLOW CHART Figure 15



**DS18S20 OPERATION EXAMPLE 1**

In this example there are multiple DS18S20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18S20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18S20 ROM code.
TX	44h	Master issues Convert T command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion ( $t_{conv}$ ).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20s respond with presence pulse.
TX	55h	Master issues Match ROM command.
TX	64-bit ROM code	Master sends DS18S20 ROM code.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

**DS18S20 OPERATION EXAMPLE 2**

In this example there is only one DS18S20 on the bus and it is using parasite power. The master writes to the  $T_H$  and  $T_L$  registers in the DS18S20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	4Eh	Master issues Write Scratchpad command.
TX	2 data bytes	Master sends two data bytes to scratchpad ( $T_H$ and $T_L$ )
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	48h	Master issues Copy Scratchpad command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10 ms while copy operation is in progress.

### DS18S20 OPERATION EXAMPLE 3

In this example there is only one DS18S20 on the bus and it is using parasite power. The bus master initiates a temperature conversion then reads the DS18S20 scratchpad and calculates a higher resolution result using the data from the temperature, COUNT REMAIN and COUNT PER °C registers.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
TR	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	44h	Master issues Convert T command.
TX	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion ( $t_{conv}$ ).
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
TX	CCh	Master issues Skip ROM command.
TX	BEh	Master issues Read Scratchpad command.
RX	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. The master also calculates the TEMP_READ value and stores the contents of the COUNT REMAIN and COUNT PER °C registers.
TX	Reset	Master issues reset pulse.
RX	Presence	DS18S20 responds with presence pulse.
-	-	CPU calculates extended resolution temperature using the equation in the OPERATION - MEASURING TEMPERATURE section of this datasheet.

### RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS18S20. These notes can be obtained from the Dallas Semiconductor "Application Note Book," via the Dallas website at <http://www.dalsemi.com/>, or through our faxback service at (214) 450-0441.

Application Note 27: "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Product"

Application Note 55: "Extending the Contact Range of Touch Memories"

Application Note 74: "Reading and Writing Touch Memories via Serial Interfaces"

Application Note 104: "Minimalist Temperature Control Demo"

Application Note 105: "High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors"

Application Note 106: "Complex MicroLANs"

Application Note 108: "MicroLAN – In the Long Run"

Sample 1-wire subroutines that can be used in conjunction with AN74 can be downloaded from the Dallas website or anonymous FTP Site.

**ABSOLUTE MAXIMUM RATINGS\***

Voltage on any pin relative to ground	-0.5V to +6.0V
Operating temperature	-55°C to +125°C
Storage temperature	-55°C to +125°C
Soldering temperature	See J-STD-020A Specification

\*These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

**DC ELECTRICAL CHARACTERISTICS** (-55°C to +125°C;  $V_{DD}=3.0V$  to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	$V_{DD}$	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	$V_{PU}$	Parasite Power	+3.0		+5.5	V	1,2
		Local Power	+3.0		$V_{DD}$		
Thermometer Error	$t_{ERR}$	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic Low	$V_{IL}$		-0.3		+0.8	V	1,4,5
Input Logic High	$V_{IH}$	Local Power	+2.2		The lower of 5.5 or $V_{DD} + 0.3$	V	1, 6
		Parasite Power	+3.0				
Sink Current	$I_L$	$V_{IO}=0.4V$	4.0			mA	1
Standby Current	$I_{DDs}$			750	1000	nA	7,8
Active Current	$I_{DD}$	$V_{DD}=5V$		1	1.5	mA	9
DQ Input Current	$I_{DQ}$			5		μA	10
Drift				±0.2		°C	11

**NOTES:**

- All voltages are referenced to ground.
- The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to  $V_{PU}$ . In order to meet the  $V_{IH}$  spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus:  $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$ .
- See typical performance curve in Figure 16
- Logic low voltages are specified at a sink current of 4 mA.
- To guarantee a presence pulse under low voltage parasite power conditions,  $V_{ILMAX}$  may have to be reduced to as low as 0.5V.
- Logic high voltages are specified at a source current of 1 mA.
- Standby current specified up to 70°C. Standby current typically is 3 μA at 125°C.
- To minimize  $I_{DDs}$ , DQ should be within the following ranges:  $GND \leq DQ \leq GND + 0.3V$  or  $V_{DD} - 0.3V \leq DQ \leq V_{DD}$ .
- Active current refers to supply current during active temperature conversions or EEPROM writes.
- DQ line is high ("hi-Z" state).
- Drift data is based on a 1000 hour stress test at 125°C with  $V_{DD} = 5.5V$ .



**AC ELECTRICAL CHARACTERISTICS: NV MEMORY**(-55°C to +100°C;  $V_{DD}=3.0V$  to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
NV Write Cycle Time	$t_{WT}$			2	10	ms
EEPROM Writes	$N_{EEWR}$	-55°C to +55°C	50k			writes
EEPROM Data Retention	$t_{EDR}$	-55°C to +55°C	10			years

**AC ELECTRICAL CHARACTERISTICS** (-55°C to +125°C;  $V_{DD}=3.0V$  to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	$t_{CONV}$				750	ms	1
Time to Strong Pullup On	$t_{SPON}$	Start Convert T Command Issued			10	$\mu s$	
Time Slot	$t_{SLOT}$		60		120	$\mu s$	1
Recovery Time	$t_{REC}$		1			$\mu s$	1
Write 0 Low Time	$t_{LOW0}$		60		120	$\mu s$	1
Write 1 Low Time	$t_{LOW1}$		1		15	$\mu s$	1
Read Data Valid	$t_{RDV}$				15	$\mu s$	1
Reset Time High	$t_{RSTH}$		480			$\mu s$	1
Reset Time Low	$t_{RSTL}$		480			$\mu s$	1,2
Presence Detect High	$t_{PDHIGH}$		15		60	$\mu s$	1
Presence Detect Low	$t_{PDLOW}$		60		240	$\mu s$	1
Capacitance	$C_{IN/OUT}$				25	pF	

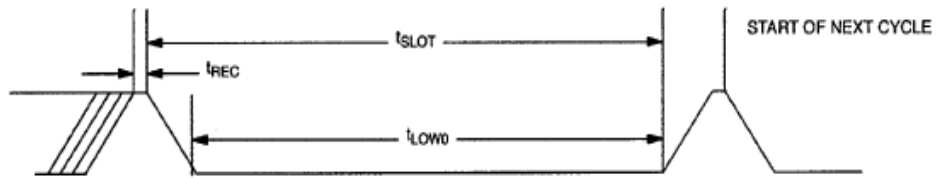
**NOTES:**

1. Refer to timing diagrams in Figure 17.
2. Under parasite power, if  $t_{RSTL} > 960 \mu s$ , a power on reset may occur.

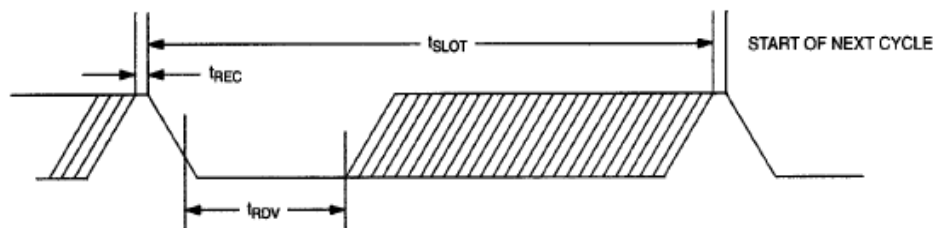
**TYPICAL PERFORMANCE CURVE** Figure 16

## TIMING DIAGRAMS Figure 17

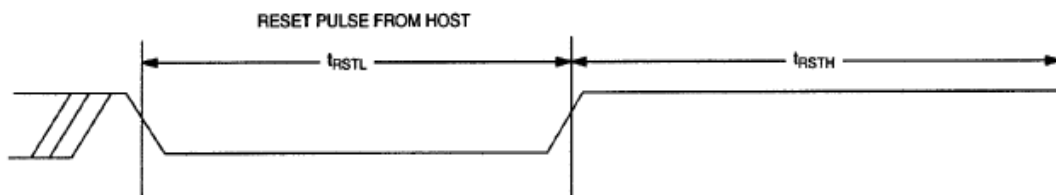
## 1-WIRE WRITE ZERO TIME SLOT



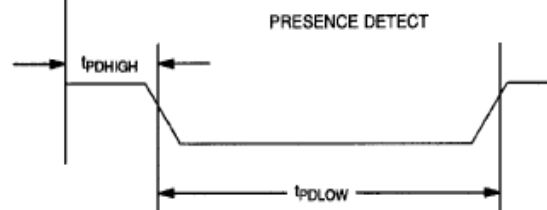
## 1-WIRE READ ZERO TIME SLOT



## 1-WIRE RESET PULSE



## 1-WIRE PRESENCE DETECT



# ANEXO V

## Triacs

## BT136 series

## GENERAL DESCRIPTION

Glass passivated triacs in a plastic envelope, intended for use in applications requiring high bidirectional transient and blocking voltage capability and high thermal cycling performance. Typical applications include motor control, industrial and domestic lighting, heating and static switching.

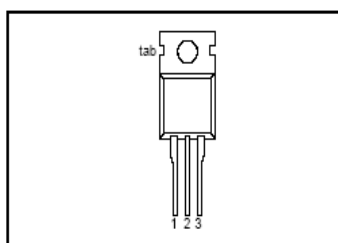
## QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	MAX.	MAX.	UNIT
$V_{DRM}$	BT136-	500	600	800	V
	BT136-	500F	600F	800F	
	BT136-	500G	600G	800G	
$I_{T(RMS)}$	Repetitive peak off-state voltages	500	600	800	A
$I_{TSM}$	RMS on-state current	4	4	4	A
	Non-repetitive peak on-state current	25	25	25	A

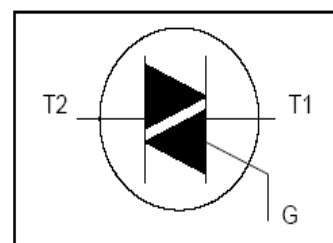
## PINNING - TO220AB

PIN	DESCRIPTION
1	main terminal 1
2	main terminal 2
3	gate
tab	main terminal 2

## PIN CONFIGURATION



## SYMBOL



## LIMITING VALUES

Limiting values in accordance with the Absolute Maximum System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.			UNIT
				-500	-600	-800	
$V_{DRM}$	Repetitive peak off-state voltages		-	500 <sup>1</sup>	600 <sup>1</sup>	800	V
$I_{T(RMS)}$	RMS on-state current	full sine wave; $T_{mb} \leq 107^\circ\text{C}$	-	4			A
$I_{TSM}$	Non-repetitive peak on-state current	full sine wave; $T_j = 25^\circ\text{C}$ prior to surge	-	25			A
		$t = 20\text{ ms}$	-	27			A
$I^2t$	$I^2t$ for fusing	$t = 10\text{ ms}$	-	3.1			A <sup>2</sup> s
$di_T/dt$	Repetitive rate of rise of on-state current after triggering	$I_{TM} = 6\text{ A}$ ; $I_G = 0.2\text{ A}$ ; $di_G/dt = 0.2\text{ A}/\mu\text{s}$	-	50			A/ $\mu\text{s}$
$I_{GM}$	Peak gate current	T2+ G+	-	50			A/ $\mu\text{s}$
		T2+ G-	-	50			A/ $\mu\text{s}$
		T2- G-	-	50			A/ $\mu\text{s}$
		T2- G+	-	10			A/ $\mu\text{s}$
$V_{GM}$	Peak gate voltage		-	2			V
$P_{GM}$	Peak gate power		-	5			W
$P_{G(AV)}$	Average gate power	over any 20 ms period	-	0.5			W
$T_{stg}$	Storage temperature		-40	150			$^\circ\text{C}$
$T_j$	Operating junction temperature		-	125			$^\circ\text{C}$

<sup>1</sup> Although not recommended, off-state voltages up to 800V may be applied without damage, but the triac may switch to the on-state. The rate of rise of current should not exceed 3 A/ $\mu\text{s}$ .

## Triacs

## BT136 series

## THERMAL RESISTANCES

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$R_{th\ j-mb}$	Thermal resistance junction to mounting base	full cycle	-	-	3.0	K/W
$R_{th\ j-a}$	Thermal resistance junction to ambient	half cycle in free air	-	60	3.7	K/W
			-		-	K/W

## STATIC CHARACTERISTICS

$T_j = 25\text{ }^\circ\text{C}$  unless otherwise stated

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.			UNIT
$I_{GT}$	Gate trigger current	<b>BT136-</b> $V_D = 12\text{ V}; I_T = 0.1\text{ A}$	-	-	...	...F	...G	
		T2+ G+	-	5	35	25	50	mA
		T2+ G-	-	8	35	25	50	mA
		T2- G-	-	11	35	25	50	mA
		T2- G+	-	30	70	70	100	mA
$I_L$	Latching current	$V_D = 12\text{ V}; I_{GT} = 0.1\text{ A}$	-	-	-	-	-	
		T2+ G+	-	7	20	20	30	mA
		T2+ G-	-	16	30	30	45	mA
		T2- G-	-	5	20	20	30	mA
		T2- G+	-	7	30	30	45	mA
$I_H$	Holding current	$V_D = 12\text{ V}; I_{GT} = 0.1\text{ A}$	-	5	15	15	30	mA
$V_T$	On-state voltage	$I_T = 5\text{ A}$	-	1.4	1.70			V
$V_{GT}$	Gate trigger voltage	$V_D = 12\text{ V}; I_T = 0.1\text{ A}$	-	0.7	1.5			V
		$V_D = 400\text{ V}; I_T = 0.1\text{ A};$ $T_j = 125\text{ }^\circ\text{C}$	0.25	0.4	-			V
$I_D$	Off-state leakage current	$V_D = V_{DRM(max)}$ ; $T_j = 125\text{ }^\circ\text{C}$	-	0.1	0.5			mA

## DYNAMIC CHARACTERISTICS

$T_j = 25\text{ }^\circ\text{C}$  unless otherwise stated

SYMBOL	PARAMETER	CONDITIONS	MIN.			TYP.	MAX.	UNIT
$dV_D/dt$	Critical rate of rise of off-state voltage	<b>BT136-</b> $V_{DM} = 67\% V_{DRM(max)}$ ; $T_j = 125\text{ }^\circ\text{C}$ ; exponential waveform; gate open circuit	...	...F	...G	250	-	V/ $\mu\text{s}$
$dV_{com}/dt$	Critical rate of change of commutating voltage	$V_{DM} = 400\text{ V}; T_j = 95\text{ }^\circ\text{C};$ $I_{T(RMS)} = 4\text{ A};$ $di_{com}/dt = 1.8\text{ A/ms}$ ; gate open circuit	-	-	10	50	-	V/ $\mu\text{s}$
$t_{gt}$	Gate controlled turn-on time	$I_{TM} = 6\text{ A}; V_D = V_{DRM(max)}$ ; $I_G = 0.1\text{ A}; di_G/dt = 5\text{ A}/\mu\text{s}$	-	-	-	2	-	$\mu\text{s}$

Triacs

BT136 series

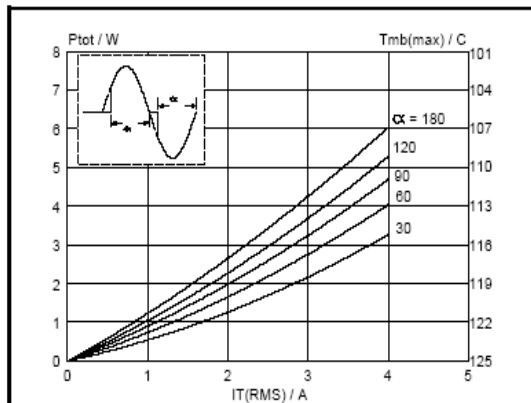


Fig. 1. Maximum on-state dissipation,  $P_{tot}$ , versus rms on-state current,  $I_{T(RMS)}$ , where  $\alpha$  = conduction angle.

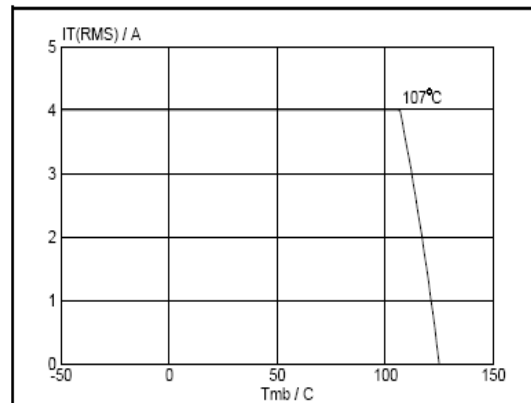


Fig. 4. Maximum permissible rms current  $I_{T(RMS)}$ , versus mounting base temperature  $T_{mb}$ .

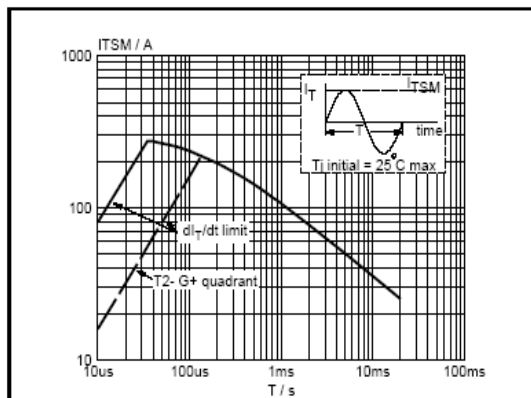


Fig. 2. Maximum permissible non-repetitive peak on-state current  $I_{TSM}$ , versus pulse width  $t_p$ , for sinusoidal currents,  $t_p \leq 20$ ms.

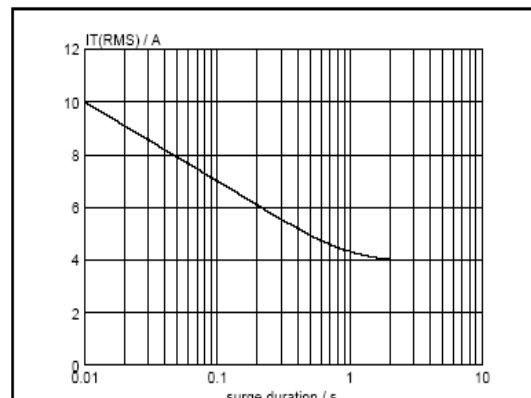


Fig. 5. Maximum permissible repetitive rms on-state current  $I_{T(RMS)}$ , versus surge duration, for sinusoidal currents,  $f = 50$  Hz;  $T_{mb} \leq 107$ °C.

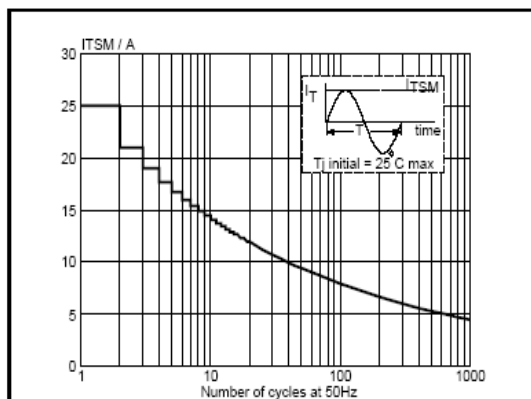


Fig. 3. Maximum permissible non-repetitive peak on-state current  $I_{TSM}$ , versus number of cycles, for sinusoidal currents,  $f = 50$  Hz.

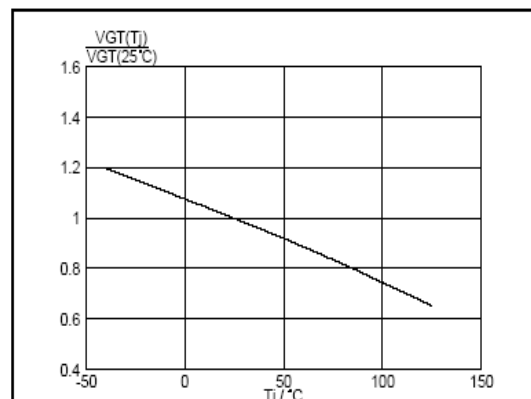
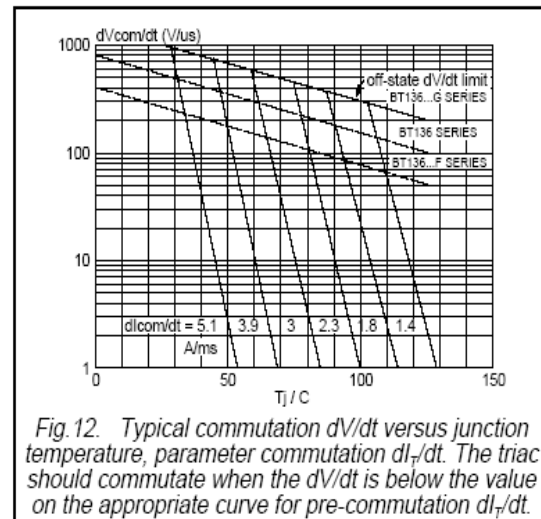
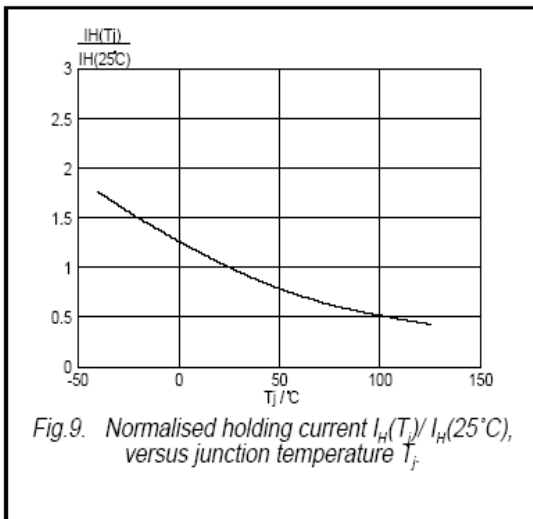
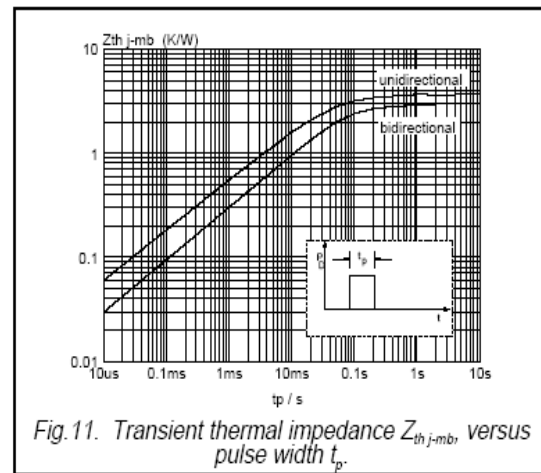
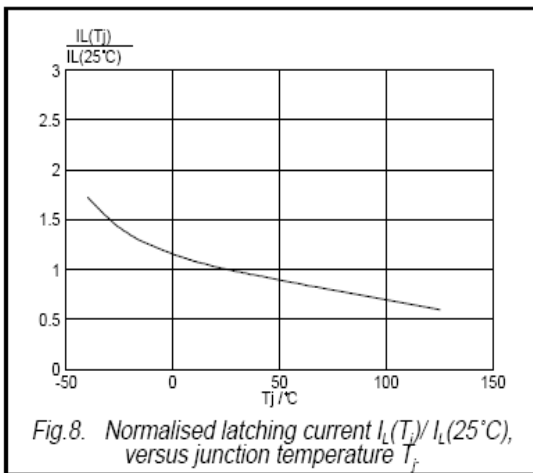
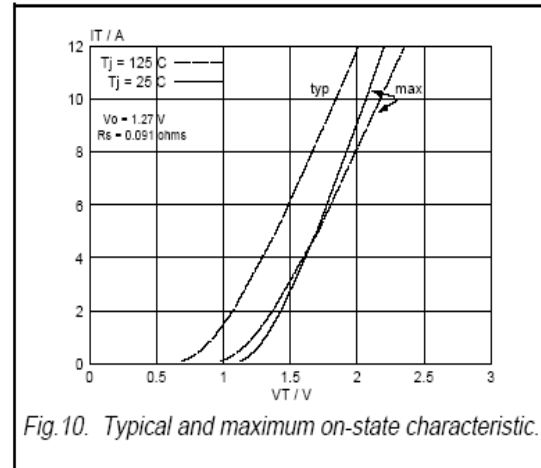
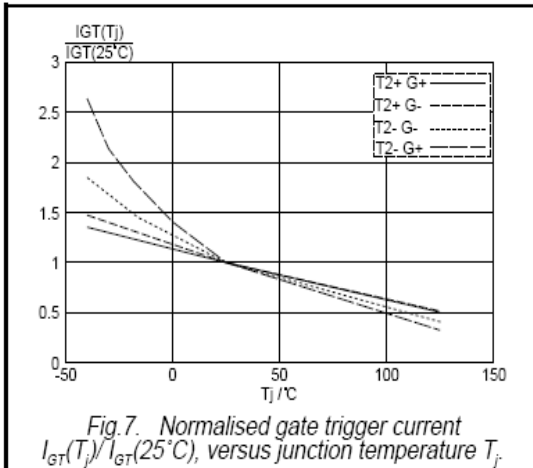


Fig. 6. Normalised gate trigger voltage  $V_{GT}(T_j)/V_{GT}(25^\circ C)$ , versus junction temperature  $T_j$ .

Triacs

BT136 series



## Triacs

## BT136 series

## MECHANICAL DATA

Dimensions in mm

Net Mass: 2 g

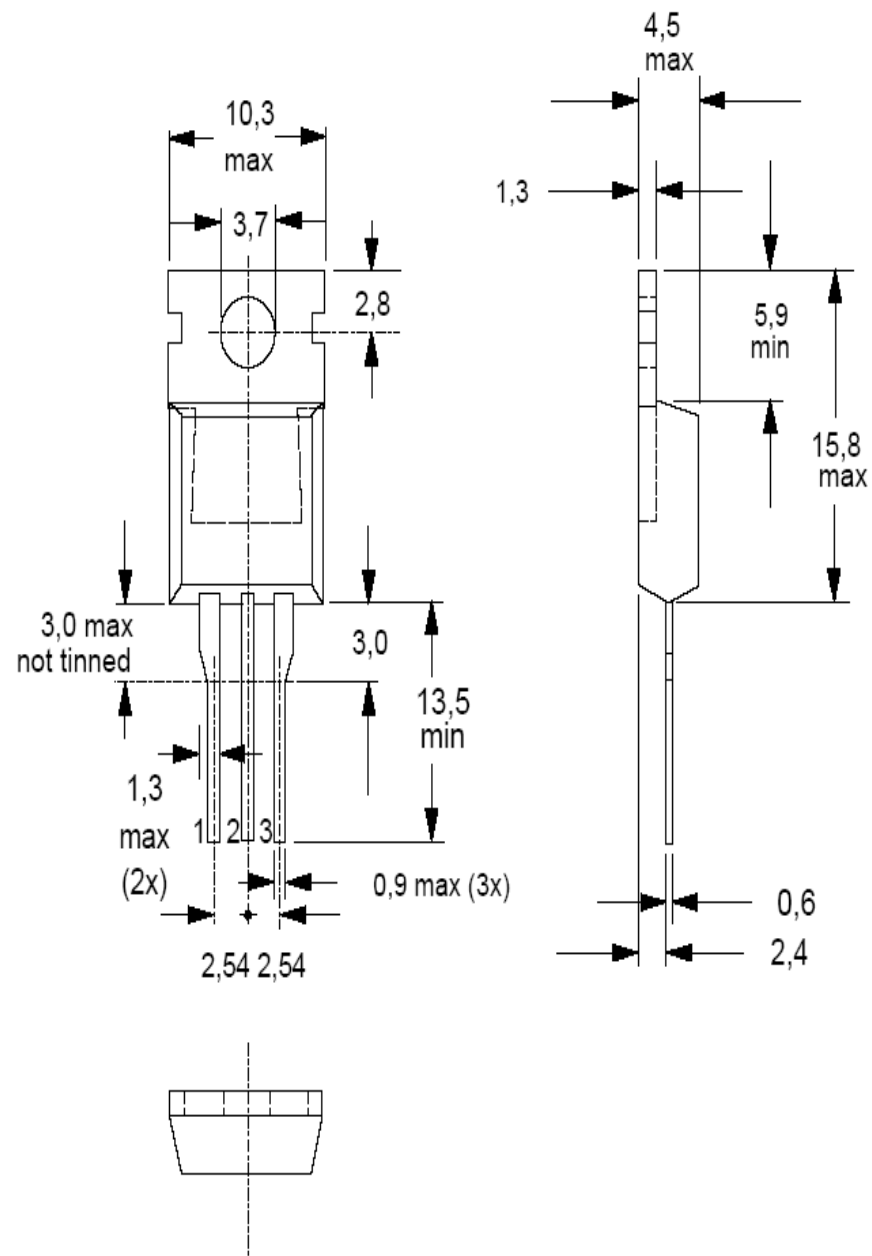


Fig. 13. TO220AB; pin 2 connected to mounting base.

**Notes**

1. Refer to mounting instructions for TO220 envelopes.
2. Epoxy meets UL94 V0 at 1/8".



## Triacs

## BT136 series

**DEFINITIONS**

<b>Data sheet status</b>	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
<b>Limiting values</b>	
Limiting values are given in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of this specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
<b>Application information</b>	
Where application information is given, it is advisory and does not form part of the specification.	
© Philips Electronics N.V. 1997	
All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.	
The information presented in this document does not form part of any quotation or contract, it is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights.	

**LIFE SUPPORT APPLICATIONS**

These products are not designed for use in life support appliances, devices or systems where malfunction of these products can be reasonably expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

# ANEXO VI



## BIBLIOGRAFÍA.

- [www.datasheetcatalog.com](http://www.datasheetcatalog.com)
- [www.control-systems-principles.co.uk](http://www.control-systems-principles.co.uk)
- [www.stiebel-eltron-usa.com](http://www.stiebel-eltron-usa.com)
- [www.termotronic.com/termoe](http://www.termotronic.com/termoe)
- [www.atmel.com](http://www.atmel.com)
- [www.puntolog.com/actual/ESPECIAL\\_LOGICA\\_BORROSA/](http://www.puntolog.com/actual/ESPECIAL_LOGICA_BORROSA/)
- [www.imse.cnm.es/Xfuzzy/Fleb/Fleb.htm](http://www.imse.cnm.es/Xfuzzy/Fleb/Fleb.htm)
  
- Introducción y Conceptos Básicos de Lógica Difusa (Parte I).  
<http://tesla.cuao.edu.co/proyectos/ic/fuzzyintr> .
  
- [www.lsi.us.es/joaquinp/doc/fuzzy.htm](http://www.lsi.us.es/joaquinp/doc/fuzzy.htm)
  
- Zadeh. "Fuzzy Sets". Information and Control; Vol 8 págs:338-353; 1965
  
- L. A. Zadeh. "Outline of a new approach to the analysis of complex systems and decision processes". IEEE: Trans. Syst., Man, Cybern.; Vol 3 págs:28-44; January-1973
  
- L. A. Zadeh. "Fuzzy Logic". IEEE Computer Mag; págs:83-93; Apr-1988
  
- E.H. Mamdani, J.J. Odtengard, E. Lembessis. "An experiment in Linguistic synthesis with a fuzzy logic controller". Int J. Man. Machine Studies.; Vol 7 págs:1-13; 1975
  
- E.H. Mamdani, J.J. Odtengard, E. Lembessis. "Use of fuzzy logic for implementing rule-based control of industrial processes". Advances in Fuzzy Sets.; Possibility Theory and Applications (Wang eds) Plenum Press 1983

- Antonio Morillas Raya. "INTRODUCCION AL ANÁLISIS DE DATOS DIFUSOS" Departamento de Estadística y Econometría Universidad de Málaga.
- Muñoz y A. Pardo, "Revista Colombiana de computación". Volumen 2 Numero 1. Pags 35 – 42.
- C. Vera, "Diseño de sistema de control de un horno rotatorio, productor de carbón activado", Trabajo de Título, Departamento de Ingeniería Eléctrica, Universidad de La Frontera, 2004.
- Alvarez López, I: Sistemas para el control automático de procesos aplicando lógica difusa. Memoria para obtención de la suficiencia investigadora. Universidad de Granada (2002)
- Ares, Dulce Maria; García H.; Naranjo S.; Peláez I. "Caracterización parcial de las fracciones proteicas extraídas de las hojas de tabaco". Cuba tabaco 1(1): 55-61, 1999.
- Driankov, D.; Hellendoorn, H.; Reinfrank, M.; *"An Introduction to Fuzzy Control"*, Springer-Verlag,1993.
- Kalfrisa. "Manual de usuario", 1988. [6] Kosko B., "Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machinery Intelligence" Prentice Hall Inc. 1992.