

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE INGENIERÍA**

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE CONTROL DE  
UNA TORRE DE DESTILACIÓN**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES.**

**MARCO ANTONIO CHÁVEZ PEÑA**

**DIRECTOR: ING. OSWALDO BUITRÓN**

**Quito, Abril 2005**

## DECLARACIÓN

Yo, Marco Antonio Chávez Peña, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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



Marco Antonio Chávez Peña

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Marco Antonio Chávez Peña, bajo mi supervisión.



---

Ing. Oswaldo Buitrón  
DIRECTOR DE PROYECTO

## **AGRADECIMIENTOS**

Primeramente quiero agradecer a Dios, el cual con su infinito amor me ha guiado y ayudado a través de los senderos de la vida.

A mis padres Jorge Aníbal y Ana Elisa que en todo momento no dejaron de apoyarme y darme siempre su consejo sabio. Siempre estuvieron ahí con una voz de aliento en los momentos difíciles.

A mis hermanas Ana Cristina y Sandra Paola a las que quiero mucho y agradezco la paciencia que han tenido conmigo.

A Liz Thornton esa persona muy especial que me ha traído mucha felicidad y me ha enseñado a ver la vida de un modo diferente.

Al Ing. Oswaldo Buitrón a quien lo considero un buen amigo y quien me supo guiar a través de todo este proyecto.

Al Ing. Marcelo Acosta quien aportó mucho con sus conocimientos de Ingeniería Química.

***Marco Antonio Chávez***

## DEDICATORIA

Este trabajo lo dedico a Dios el cual me ha dado la fortaleza para seguir adelante en los momentos más difíciles.

A mis padres Jorge Aníbal y Ana Elisa que son un ejemplo de amor y perseverancia.

A mis hermanas Ana Cristina y Sandra Paola a las que espero no haber defraudado.

Y a Liz Thornton a la que considero una persona muy inteligente, bondadosa y amorosa.

***Marco Antonio Chávez***

# CONTENIDO

## 1. CAPÍTULO 1: GENERALIDADES

1.1. INTRODUCCIÓN.....	1
1.2. TECNOLOGÍA DE PRODUCCIÓN DEL ALCOHOL.....	2
1.2.1. CORTADO.....	3
1.2.2. EXPRIMIDO.....	3
1.2.3. FERMENTADO.....	3
1.2.4. DESTILADO.....	4
1.3. IMPACTO DEL PROYECTO EN LA SOCIEDAD.....	4
1.4. JUSTIFICACIÓN DEL PROYECTO.....	5

## 2. CAPITULO 2: DISEÑO DEL PROTOTIPO

2.1. PROCESO A SER CONTROLADO.....	6
2.1.1. PARTES DE LA TORRE DE DESTILACIÓN.....	7
2.1.1.1. CALDERÍN O REBOILER.....	7
2.1.1.2. DISPOSITIVO DE CALENTAMIENTO.....	7
2.1.1.3. TUBOS DE VIDRIO.....	8
2.1.1.4. TERMOSIFÓN.....	9
2.1.1.5. ALIMENTACIÓN.....	9
2.1.1.6. CONDENSADOR.....	10
2.1.1.7. ACUMULADOR.....	11
2.1.2. FUNCIONAMIENTO DE LA TORRE.....	11
2.2. OBJETIVOS.....	13
2.3. LOS SENSORES.....	13
2.3.1. SENSOR DE TEMPERATURA.....	14
2.3.1.1. CARACTERÍSTICAS DEL CIRCUITO DS1821.....	14
2.3.1.2. DESCRIPCIÓN DE LOS PINES.....	15
2.3.1.3. DESCRIPCIÓN GENERAL.....	16
2.3.1.4. OPERACIÓN DEL DS1821 PARA MEDICIONES DE TEMPERATURA.....	17
2.3.1.5. REGISTRO DE CONFIGURACIÓN Y DE ESTADO.....	20

2.3.1.5.1. DESCRIPCIÓN DE LOS BITS DEL REGISTRO DE CONFIGURACIÓN Y DE ESTADO.....	21
2.3.1.6. PROTOCOLO 1-WIRE.....	22
2.3.1.6.1. HARDWARE.....	23
2.3.1.6.2. SECUENCIA DE TRANSACCIÓN.....	24
2.3.1.6.2.1. INICIALIZACIÓN.....	24
2.3.1.6.2.2. COMANDOS DE FUNCIONES DEL DS1821...24	
2.3.1.6.3. TIPOS DE SEÑALES DEL PROTOCOLO 1-WIRE....	25
2.3.1.6.3.1. PROCESO DE INICIALIZACIÓN, PULSOS DE RESET Y DE PRESENCIA.....	26
2.3.1.6.3.2. SLOTS DE TIEMPO DE LECTURA Y ESCRITURA.....	27
2.3.1.6.3.2.1. SLOTS DE TIEMPO DE ESCRITURA.....	27
2.3.1.6.3.2.2. SLOTS DE TIEMPO DE LECTURA.....	27
2.3.2. SENSORES DE NIVEL.....	28
2.3.3. INTERRUPTORES ÓPTICOS.....	31
2.3.4. TECLADO.....	32
2.3.5. INTERRUPTORES MAGNÉTICOS.....	33
2.4. LOS ACTUADORES.....	34
2.4.1. ACTUADOR PARA EL DISPOSITIVO DE CALENTAMIENTO...34	
2.4.1.1. DETECTOR DE CRUCE POR CERO.....	35
2.4.1.2. CIRCUITO DE DISPARO DEL TRIAC.....	40
2.4.2. MOTOR PASO A PASO.....	43
2.4.2.1. PRINCIPIO DE FUNCIONAMIENTO.....	44
2.4.3. ELECTROVÁLVULA.....	46
2.4.4. BOMBAS.....	47
2.4.5. LCD.....	48
2.4.5.1. DESCRIPCIÓN DE LOS PINES DE UN LCD.....	49
2.4.5.1.1. PINES GND Y VCC.....	49
2.4.5.1.2. PIN CONT.....	49
2.4.5.1.3. PIN RS.....	49
2.4.5.1.4. PIN R/W.....	49
2.4.5.1.5. PIN E.....	50
2.4.5.1.6. BUS DE DATOS.....	50

2.4.5.2.	DIAGRAMAS DE TIEMPO DE ESCRITURA Y LECTURA.....	50
2.4.5.2.1.	ESCRITURA DE UNA INSTRUCCIÓN.....	50
2.4.5.2.2.	ESCRITURA DE UN DATO.....	51
2.4.5.2.3.	LECTURA DE UN DATO.....	51
2.4.5.3.	INICIALIZACIÓN DEL MÓDULO LCD.....	53
2.4.5.4.	INSTRUCCIONES DEL LCD.....	54
2.4.6.	LED.....	56
2.4.7.	RELAYS.....	56
2.5.	LAS COMUNICACIONES.....	59
2.5.1.	INTERFAZ SERIAL.....	60
2.5.1.1.	MODO 0.....	60
2.5.1.2.	MODO 1.....	61
2.5.1.3.	MODO 2.....	61
2.5.1.4.	MODO 3.....	61
2.5.2.	COMUNICACIONES EN UN AMBIENTE MULTIPROCESADOR.....	62
2.5.3.	CONTROL DE ERRORES.....	63
2.5.3.1.	CÓDIGOS DE REDUNDANCIA CÍCLICA.....	64
2.6.	CONFIGURACIÓN E INTERFACES.....	76
2.6.1.	CIRCUITO DEL PANEL PRINCIPAL.....	78
2.6.2.	CIRCUITO PARA EL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE.....	79
2.6.3.	CIRCUITO DE POTENCIA PARA EL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE.....	80
2.6.4.	CIRCUITO PARA EL CONTROL DEL NIVEL DENTRO DEL REBOILER.....	80
2.6.5.	CIRCUITO DE POTENCIA DEL CONTROL DEL NIVEL DEL REBOILER.....	81
2.6.6.	CIRCUITO DE CONTROL PARA EL ACUMULADOR, VENTILADOR Y TANQUES DE ALIMENTACIÓN Y DE RESERVA.....	81

2.6.7. CIRCUITO DE POTENCIA PARA EL CONTROL DEL ACUMULADOR VENTILADOR Y TANQUES DE RESERVA Y ALIMENTACIÓN.....	82
2.7. POSIBILIDADES DE CONTROL.....	82
2.7.1. OPCIONES DEL PANEL PRINCIPAL.....	88
2.7.2. OPCIONES DEL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE.....	91
2.7.3. OPCIONES DEL CONTROL DE NIVEL DEL REBOILER.....	92
2.7.4. OPCIONES DE CONTROL DEL ACUMULADOR, VENTILADOR, BOMBAS DE REFLUJO Y ALIMENTACIÓN.....	92
2.8. PERTURBACIONES.....	93

### 3. CAPÍTULO 3: IMPLEMENTACIÓN DEL SOFTWARE

3.1. SUBROUTINAS DEL PANEL PRINCIPAL.....	101
3.1.1. PROGRAMA PRINCIPAL.....	101
3.1.2. SUBROUTINA INI_4BITS.....	102
3.1.3. SUBROUTINA COMANDO.....	102
3.1.4. SUBROUTINA CLS.....	104
3.1.5. SUBROUTINA HOME.....	104
3.1.6. SUBROUTINA CARÁCTER.....	105
3.1.7. SUBROUTINA LCD.....	105
3.1.8. SUBROUTINA TECLADO.....	105
3.1.9. SUBROUTINA BEEPER.....	106
3.1.10. SUBROUTINA COD2HEX.....	106
3.1.11. SUBROUTINA HEX2ASCII.....	106
3.1.12. SUBROUTINA HEX2BCD.....	107
3.1.13. SUBROUTINA BCD2HEX.....	107
3.1.14. SUBROUTINA INGRESE_DATO.....	108
3.1.15. SUBROUTINA VER_TEMPERATURA.....	108
3.1.16. SUBROUTINA INT_SERIAL.....	109
3.1.17. SUBROUTINA ENVIAR.....	109
3.1.18. SUBROUTINA ENVIAR_DATO.....	109
3.1.19. SUBROUTINA BORRAR.....	110
3.1.20. SUBROUTINA CONTROL.....	110

3.1.21.	SUBROUTINA BAJAR_MENU.....	111
3.1.22.	SUBROUTINA SUBIR_MENU.....	111
3.1.23.	SUBROUTINA CLAVE.....	111
3.2. SUBROUTINAS DEL CONTROL DE TEMPERATURA DEL		
	REBOILER.....	141
3.2.1.	PROGRAMA PRINCIPAL.....	141
3.2.2.	SUBROUTINA AUTOMÁTICO.....	141
3.2.3.	SUBROUTINA MANUAL.....	142
3.2.4.	SUBROUTINA INT_SERIAL.....	142
3.2.5.	SUBROUTINA SERIAL_AUTOMÁTICO.....	143
3.2.6.	SUBROUTINA SERIAL_MANUAL.....	144
3.2.7.	SUBROUTINA DALLAS_RESET.....	144
3.2.8.	SUBROUTINA DALLAS_WRITE.....	145
3.2.9.	SUBROUTINA DALLAS_READ.....	145
3.2.10.	SUBROUTINA MEDIR.....	146
3.2.11.	SUBROUTINA CONTROL_TEMPE.....	147
3.2.12.	SUBROUTINA INCRE_ANGULO.....	147
3.2.13.	SUBROUTINA DECRE_ANGULO.....	148
3.2.14.	SUBROUTINA ACT_ENABLE.....	148
3.2.15.	SUBROUTINA ENVIAR_DATOS.....	148
3.2.16.	SUBRTUINA ENVIAR.....	149
3.3. SUBROUTINA PARA EL CONTROL DEL NIVEL DEL		
	REBOILER.....	165
3.3.1.	PROGRAMA PRINCIPAL.....	165
3.3.2.	SUBROUTINA AUTOMÁTICO.....	165
3.3.3.	SUBROUTINA MANUAL.....	166
3.3.4.	SUBROUTINA INT_SERIAL.....	166
3.3.5.	SUBROUTINA SERIAL_AUTOMÁTICO.....	167
3.3.6.	SUBROUTINA SERIAL_MANUAL.....	168
3.3.7.	SUBROUTINA CALIBRAR.....	168
3.3.8.	SUBROUTINA SEN_NIVEL.....	169
3.3.9.	SUBROUTINA ACT_SENSORES.....	169
3.3.10.	SUBROUTINA ABRIR.....	170
3.3.11.	SUBROUTINA CERRAR.....	170
3.3.12.	SUBROUTINA INCREMENTAR.....	171

3.3.13.	SUBROUTINA DECREMENTAR.....	171
3.3.14.	SUBROUTINA ACT_MOTOR.....	171
3.4.	SUBROUTINAS PARA EL CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS.....	187
3.4.1.	PROGRAMA PRINCIPAL.....	187
3.4.2.	SUBROUTINA AUTOMÁTICO.....	187
3.4.3.	SUBROUTINA MANUAL.....	188
3.4.4.	SUBROUTINA INT_SERIAL.....	189
3.4.5.	SUBROUTINA SERIAL_AUTOMÁTICO.....	189
3.4.6.	SUBROUTINA SERIAL MANUAL.....	190

#### **4. PRUEBAS Y MEDICIONES DEL EQUIPO COSTRUIDO**

4.1.	PUESTA EN MARCHA DEL SISTEMA EN MODO AUTOMÁTICO.....	197
4.2.	PUESTA EN MARCHA DEL SISTEMA EN MODO MANUAL.....	206
4.3.	TABLAS DE MEDIDAS.....	209
4.3.1.	TABLA DEL CIRCUITO DE DETECCIÓN DE CRUCE POR CERO.....	209
4.3.2.	TABLA DE CONVERSIÓN DE LA SUBROUTINA COD2HEX.....	210
4.3.3.	TABLA DE CONVERSIÓN DE LA SUBROUTINA HEX2ASCII.....	211
4.3.4.	TABLA PARA EL MANEJO DEL DISPLAY DE 7 SEGMENTOS.....	212
4.3.5.	TABLA PARA LA CONVERSIÓN DE PORCENTAJE DE POTENCIA A GRADOS DE DISPARO DEL TRIAC.....	213
4.3.6.	TABLA PARA LA CONVERSIÓN DE ANGULO DE DISPARO DEL TRIAC A PORCENTAJE DE POTENCIA.....	214

## **5. CAPÍTULO 5: ANÁLISIS FINANCIERO DE LA SOLUCIÓN IMPLEMENTADA**

5.1. DETALLE DE COSTO DEL CIRCUITO DE CONTROL DE TEMPERATURA DEL REBOILER.....	217
5.2. DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DE TEMPERATURA.....	218
5.3. DETALLE DE COSTO DEL CIRCUITO DE CONTROL DEL ACUMULADOR VENTILADOR Y BOMBAS.....	218
5.4. DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS.....	220
5.5. DETALLE DE COSTO DEL CIRCUITO DE CONTROL DE NIVEL DEL REBOILER.....	220
5.6. DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DE NIVEL DEL REBOILER.....	222
5.7. DETALLE DE COSTO DEL CIRCUITO DEL PANEL PRINCIPAL.....	222
5.8. COSTO DE INGENIERÍA E IMPLEMENTACIÓN.....	224
5.9. CÁLCULO DE PRECIO DE VENTA AL PÚBLICO.....	224

## **6. CAPITULO 6: COMENTARIOS Y CONCLUSIONES**

6.1. COMENTARIOS.....	227
6.2. CONCLUSIONES.....	228

### **ANEXO 1:**

Programa en lenguaje assembler del panel principal.

### **ANEXO 2:**

Programa en lenguaje assembler del control de temperatura del reboiler.

### **ANEXO 3:**

Programa en lenguaje assembler del control de nivel del reboiler.

### **ANEXO 4:**

Programa en lenguaje assembler del control del acumulador, ventilador y bombas.

**ANEXO 5:**

Circuito y programa para medir el periodo de la señal de cruce por cero.

**ANEXO 6:**

Hojas de datos de los fabricantes de los dispositivos electrónicos utilizados.

## RESUMEN

En el capítulo 1 de este proyecto se da una breve introducción que habla de la producción de alcohol en el Ecuador. Después se discute el proceso de fabricación de alcohol, describiendo sus diferentes etapas. Por último se platica de cómo influenciará el proyecto en la sociedad y la razón por la que se realizó el proyecto.

El capítulo 2 aborda el diseño en si. Comienza describiendo el proceso que será controlado para luego determinar los objetivos de control. Después hace un estudio de los sensores y actuadores utilizados; para después pasar a describir las comunicaciones entre los microcontroladores. Terminado este estudio se pasa a describir el funcionamiento de cada uno de los circuitos implementados y las posibilidades de controlar la torre de destilación. En la parte final de este capítulo se muestran los diagramas circuitales de todos los controles implementados así como de los diagramas circuitales de los circuitos de potencia.

El capítulo 3 aborda todo lo que es el software implementado en cada uno de los microcontroladores por medio de diagramas de flujo y en adición se da una explicación de lo que hace cada programa.

El capítulo 4 describe el funcionamiento de todo el equipo tanto en modo de control manual como en el modo de control automático. Además se incluye todas las tablas utilizadas por las diferentes subrutinas así como también los valores con los que a través de las pruebas se ha tenido un buen funcionamiento de la torre.

El capítulo 5 hace un breve análisis financiero de la solución implementada. En la primera parte se ofrece un detalle de costos de todos los materiales utilizados en cada uno de los controles implementados. Después se analiza el costo de implementación e ingeniería para por último dar un posible precio de venta al público.

El capítulo 6 contiene las conclusiones de la realización del proyecto, así como comentarios de cómo mejorar el equipo.

Los anexos del 1 al 4 contienen los programas en lenguaje ensamblador que utilizan cada uno de los 4 microcontroladores utilizados en este proyecto.

El anexo 5 contiene el software y el hardware para generar la tabla de valores que servirá para determinar el tiempo en que la señal de salida del detector de cruce por cero pasa en el estado 1L.

El anexo 6 contiene las hojas de datos de los fabricantes de los dispositivos electrónicos utilizados en este proyecto.

## PRESENTACIÓN

La producción de alcohol de alta concentración en el Ecuador es escasa, la mayor parte de este producto proviene de la importación. Por ende la intención de este proyecto es la de motivar a los pequeños productores de caña de azúcar a la fabricación de alcohol industrial.

En los capítulos subsiguientes se describirá el proceso de fabricación del alcohol, centrándose principalmente en la etapa de destilación.

A través del conocimiento de las partes de las cuales está constituida la torre de destilación y su función; es que se determinó los sensores y actuadores a utilizarse para después mediante el uso de microcontroladores realizar la automatización de la misma.

También se realiza un análisis financiero en la perspectiva de vender equipos similares. En el capítulo final se presentan los comentarios y conclusiones de la solución implementada en donde se analiza que el presente proyecto puede ser usado como herramienta didáctica tanto para los estudiantes de Ingeniería Química, Ingeniería en Control e Ingeniería en Telecomunicaciones.

Cabe decir que el proyecto aquí presentado es un prototipo al que se le puede hacer muchas mejoras; pero que a su vez puede ser usado como base para realizar el control de torres de destilación mucho más grandes.

# CAPÍTULO 1: GENERALIDADES

## 1.1 INTRODUCCIÓN

La producción de alcohol etílico en el Ecuador se inició en épocas de la colonia por la necesidad, de alcoholes para la fabricación de barnices, colorantes y medicinas. El mal uso de este producto dado en el alto consumo de licores y su significación económica forzó a los gobiernos, desde la época de Flores ya en la etapa de la República, a valerse del mismo para financiar el presupuesto del estado con lo que se ha denominado impuestos de estancos, vigentes hasta ahora. Sin embargo, el alcohol utilizado hasta recientemente en el Ecuador era de un grado alcohólico reducido (alrededor de 50 grados gay lussac) y el alcohol puro que se utiliza en pocas aplicaciones industriales y farmacéuticas, es hasta ahora proveniente de la importación.

Es importante tener en cuenta que el alcohol a condición atmosférica, obtenido en torres de destilación especializadas puede alcanzar una concentración denominada azeotrópica que corresponde al 97% de concentración, y que además la industria a conseguido alcohol del 99.9%, a través de varios procedimientos entre otros: la destilación más separación, procedimientos de membrana y destilación ternaria.

Como es obvio el alcohol del 99.9% tiene un precio mayor y mayor demanda, ya que constituye actualmente un producto estratégico como aditivo de los combustibles de automotores a gasolina; es así como este compuesto deberá ser obligatoriamente utilizado debido al proyecto del gobierno actual (Presidente Lucio Gutiérrez Borbúa) el mismo que ha decidido entrar en un programa de siembra de grandes extensiones de caña de azúcar para su posterior fabricación de alcohol para ser mezclado con las gasolinas y a conseguido interesar en el proyecto tanto a sectores internos, como externos.

## 1.2 TECNOLOGÍA DE PRODUCCIÓN DEL ALCOHOL

El alcohol se puede obtener por medio de la fermentación de diferentes materias primas como son: la papa, el maíz y la caña de azúcar.

La caña de azúcar es una de las principales materias primas para la obtención del alcohol. Su tallo contiene entre un 11% y 15% de sacarosa en peso. La sacarosa constituye el elemento principal para la obtención del alcohol.

A continuación en la figura 1.1, se presenta un diagrama de flujo que muestra la producción de alcohol industrial en sus diferentes fases o etapas del proceso.

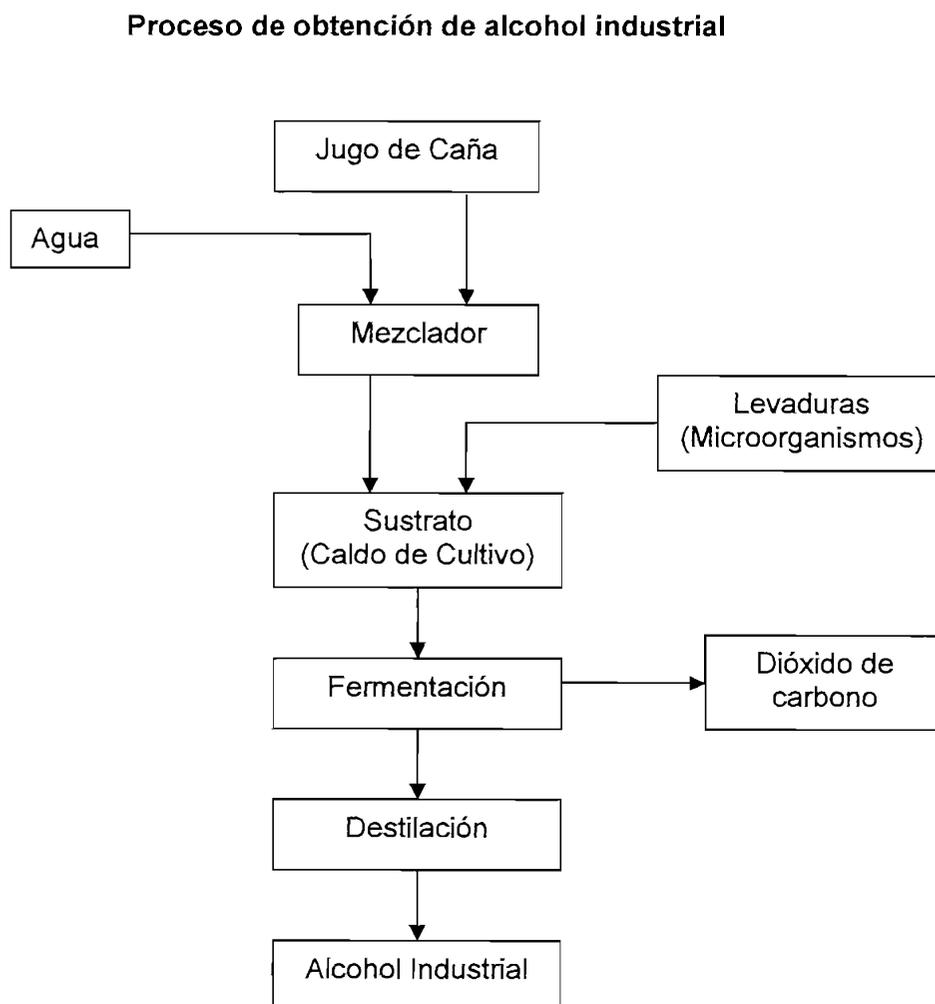


Figura 1.1

### 1.2.1 CORTADO

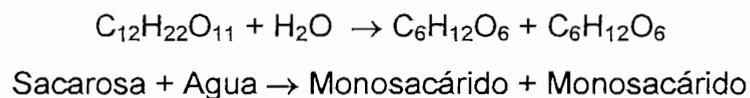
Antes de enviar las cañas de azúcar a los molinos, es necesario que pasen por un proceso previo que básicamente consiste en lavar la caña, retirar la hoja seca adherida, cortarla en trozos y rasgarlas, para que de esa manera se facilite la obtención del jugo de caña cuando pase la misma a través de los molinos.

### 1.2.2 EXPRIMIDO

Una vez que se ha lavado, trozado y rasgado las cañas, pasan a través de los denominados molinos los cuales consisten de tres tambores estriados que ejercen una gran presión sobre los trozos de caña. Agua y algunos jugos pueden adicionarse en este paso del proceso para facilitar la posterior maceración de la caña y de esa forma ayudar a los propósitos del proceso.

### 1.2.3 FERMENTADO

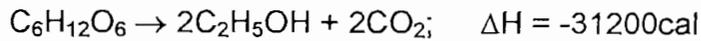
Una vez obtenido el jugo de caña se lo diluye en agua, debido a que los microorganismos no pueden vivir en ambientes con alto contenido de azúcar y se necesita de estos para conseguir la fermentación. La mezcla tiene que contener entre un 10% y 14% de concentración de azúcares. Una vez hecho esto hay que adicionar levadura (microorganismos) a la mezcla. Lo primero que hacen los microorganismos es romper el enlace de la sacarosa que es un polisacárido, y la convierte en dos monosacáridos. La fórmula química que corresponde a esta reacción esta dada por la ecuación 1.1.



Ecuación 1.1

Después proceden a destruir gradualmente los monosacáridos y convertirlos en alcohol liberando una cierta cantidad de CO<sub>2</sub> y energía como resultado de una

reacción exotérmica. Esta reacción está determinada por la ecuación de Gay Lussac, presentada en la ecuación 1.2.



Monosacárido → Alcohol + Dióxido de carbono

Ecuación 1.2

#### **1.2.4 DESTILADO**

La mezcla obtenida tiene una baja concentración de alcohol debido a lo cual para elevar esta concentración se utiliza un proceso denominado destilación el cual consiste en separar de la mezcla sólo el alcohol, por medio de la aplicación de calor y gracias a la diferencia entre los puntos de ebullición de los componentes de la que esta formada.

El presente proyecto está orientado a controlar una torre de destilación de alcohol por lo cual, la descripción en detalle de la etapa de destilación se abordará en los subsiguientes capítulos.

### **1.3 IMPACTO DEL PROYECTO EN LA SOCIEDAD**

El impacto que tendrá el proyecto en la sociedad será el de motivar la producción de alcohol industrial para el consumo interno del país. Es importante empezar a concienciar a las personas de que en el Ecuador se pueden producir muchas cosas. Pero no sólo es importante motivar a las personas a la producción sino orientar a la producción con normas de calidad, para de esa forma tener un producto competitivo que no sólo sirva para el mercado interno sino que también se lo pueda exportar.

También motivará a los ingenieros de todas las ramas a buscar soluciones a los problemas de la sociedad; así como también innovadoras formas de hacer las cosas.

## 1.4 JUSTIFICACIÓN DEL PROYECTO

*Al inicio* este proyecto se lo realizó con la idea de facilitar el manejo de la torre de destilación del Departamento de Ingeniería de Procesos y Apoyo a la Producción. Pero después se observó que puede ser utilizado como una herramienta muy eficaz para la enseñanza de la obtención del alcohol para los estudiantes de Ingeniería Química; así como también protocolos de comunicación entre microprocesadores para los estudiantes de Ingeniería Electrónica y Telecomunicaciones y control automático para los estudiantes de Ingeniería Electrónica y Control.

Además si el proyecto del gobierno actual (Presidente Lucio Gutiérrez Borbúa) se pone en marcha, se necesitarán torres de destilación mucho más grandes que pueden ser diseñadas a partir de la torre de destilación del presente proyecto.

## CAPÍTULO 2: DISEÑO DEL PROTOTIPO

### 2.1 PROCESO A SER CONTROLADO

El sistema desarrollado corresponde a un sistema de control para automatización del funcionamiento de una torre de destilación diseñada y construida en el Departamento de Ingeniería de Procesos y Apoyo a la producción.

Una torre de destilación es un dispositivo que permite separar los componentes de una mezcla, por medio del calentamiento de la misma y aprovechando la diferencia que existe entre los puntos de ebullición de los diferentes componentes de dicha mezcla. Así, los componentes con menor punto de ebullición se evaporarán más rápido y se los condensa en la parte superior de la torre, obteniéndose una parte de ellos en un recipiente, lo que se denomina el *destilado*, y el resto reingresa a la torre como realimentación. Como resultado de este proceso en la parte inferior de la torre quedan los componentes con mayor punto de ebullición, y una parte de estos se extraen para obtener lo que se denomina *fondos*.

Una mezcla puede tener 2 o más componentes. Una mezcla binaria tiene dos componentes así por ejemplo, aquella que tiene un 65% de agua y 45% de alcohol, que es lo que se obtiene del fermento del jugo de la caña de azúcar.

Una mezcla de múltiples componentes es por ejemplo el petróleo, y en este caso a estos componentes se los llama hidrocarburos. En una torre de destilación de petróleo se obtiene en la parte alta de la torre componentes como gas butano, gas propano; en la parte media se obtiene gasolinas, y en la parte baja aceites pesados.

Como la torre de destilación disponible en el Laboratorio es una torre cuya orientación es a demostrar el proceso para fines didácticos, la destilación se encuentra orientada a mezclas binarias.

Con el propósito de establecer en que etapas de la torre de destilación se puede realizar un control del proceso, a continuación se presenta una breve descripción de cada una de las partes que integran dicho equipo.

## **2.1.1 PARTES DE LA TORRE DE DESTILACIÓN**

### **2.1.1.1 CALDERÍN O REBOILER**

El calderín es el recipiente en donde se calienta la mezcla o el dispositivo donde va la materia prima, ver figura 2.1. Tiene entradas y salidas. Las entradas son las que permiten el ingreso de la mezcla ha ser separada, en cambio las salidas permiten la salida a la torre de los gases calientes de la ebullición.

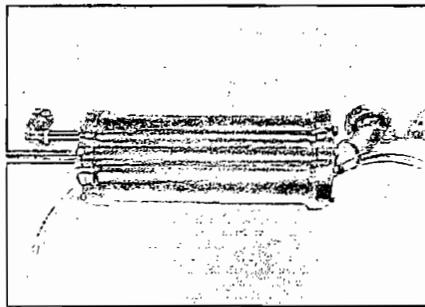


Figura 2.1

### **2.1.1.2 DISPOSITIVO DE CALENTAMIENTO**

Como es de suponer dentro del calderín se encuentra el dispositivo de calentamiento, el mismo que permitirá la ebullición de la mezcla. Un aspecto importante que hay que tomar en cuenta, es que nunca se debe conectar el dispositivo de calentamiento si el mismo no está sumergido en un suficiente nivel de líquido, porque caso contrario la resistencia se destruiría. En la figura 2.2, se puede observar la resistencia de calentamiento, dentro del calderín.

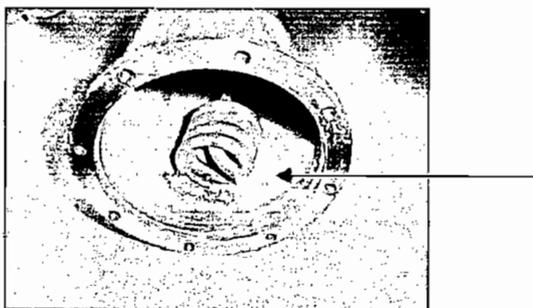


Figura 2.2

### 2.1.1.3 TUBOS DE VIDRIO

Sirven para guiar tanto el líquido así como también los gases calientes de la ebullición. Tiene tomas equidistantes en toda su longitud, su propósito es para alimentar la torre a diferentes niveles, o para tomar muestras de temperatura en esos niveles.

Dentro del tubo de vidrio existen separados equidistantemente grupos de bolitas de arcilla, cuya finalidad es permitir que haya un mejor contacto entre el líquido que baja y los gases de ebullición que suben, para que de esa forma se evapore de forma más rápida el líquido con menor índice de ebullición.

Cabe señalar que entre más altura tenga la torre, es decir en este caso, más tubos de vidrio conectados en serie, se va a tener un destilado más puro. Pero debido a que es una torre a presión atmosférica, siempre se va a tener un límite en el grado de pureza, por más altura que tenga la torre.

Esta limitación sólo puede ser superada con una modificación en el proceso, mediante una torre de destilación al vacío o con una torre de destilación en paralelo. Ver figura 2.3.

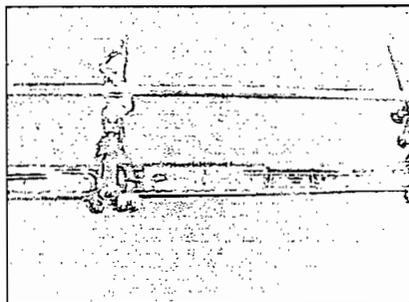


Figura 2.3

#### 2.1.1.4 **TERMOSIFÓN**

Es una válvula que se ubica en la parte baja entre el reboiler y el tubo de vidrio. Su objetivo es proveer una salida de emergencia del líquido entrante. En caso de que haya un excesivo nivel de líquido en el reboiler, se abre ésta válvula permitiendo desalojar el exceso de líquido. Que el líquido no tenga por donde salir, provocaría que se inunden las salidas, lo que provocaría que los gases de ebullición no tengan por donde salir. Ver figura 2.4.

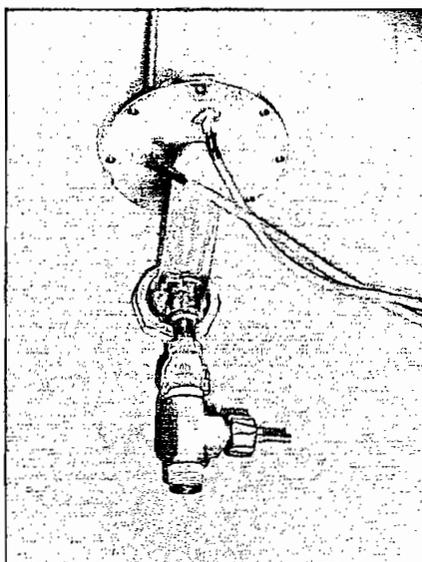


Figura 2.4

#### 2.1.1.5 **ALIMENTACIÓN**

Esta etapa del proceso se realiza con la ayuda de una válvula que permitirá el ingreso de la mezcla alcohol agua. Una característica importante de la

alimentación es que su caudal debe ser constante. Debido a ello se dispone de dos tanques uno de alimentación y otro de reserva. El tanque de alimentación es aquel que está conectado a la válvula de alimentación, en cambio el tanque de reserva es el encargado de proveer la materia prima para mantener el tanque de alimentación en un nivel constante de líquido. Debido a ello se dispone de sensores de nivel tanto en el tanque de reserva como en el de alimentación. Por lo general la alimentación se realiza a la mitad del tubo de vidrio del primer nivel. Ver figura 2.5.

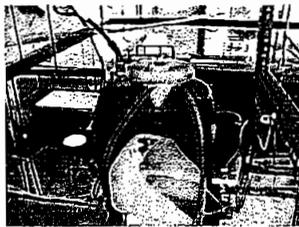


Figura 2.5

#### 2.1.1.6 CONDENSADOR

Se encuentra ubicado en la parte superior de la torre, el proceso requiere de un dispositivo que permita enfriar los gases de ebullición hasta el punto de condensación. En este caso el condensador trabaja con ayuda de un ventilador. Ver figura 2.6.

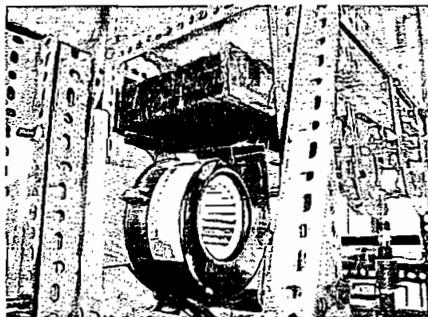


Figura 2.6

### 2.1.1.7 ACUMULADOR

El acumulador es el dispositivo que receipta el líquido condensado que viene del condensador. Dentro de este recipiente denominado acumulador existe una bomba de realimentación, la cual se activa cuando en el acumulador se ha alcanzado un suficiente nivel de líquido, y como su nombre lo indica esta bomba realimenta parte del líquido que existe en el acumulador a la torre.

Para evitar que el acumulador se inunde existe una bomba que se denomina bomba de cosecha; su objetivo es mantener un nivel constante dentro del acumulador. El líquido que se obtiene de esta bomba se denomina *producto* de la destilación. Ver figura 2.7.

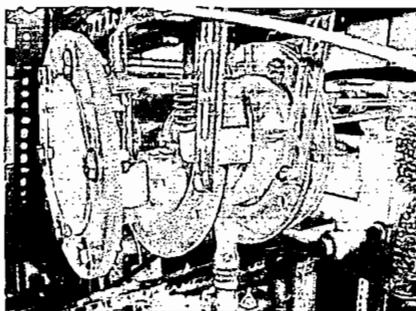


Figura 2.7

### 2.1.2 FUNCIONAMIENTO DE LA TORRE

La torre de destilación del Departamento de Ingeniería de Procesos y Apoyo a la Producción (Séptimo piso edificio Eléctrica – Química) se presenta en el esquema de la figura 2.8.

### Columna de destilación binaria

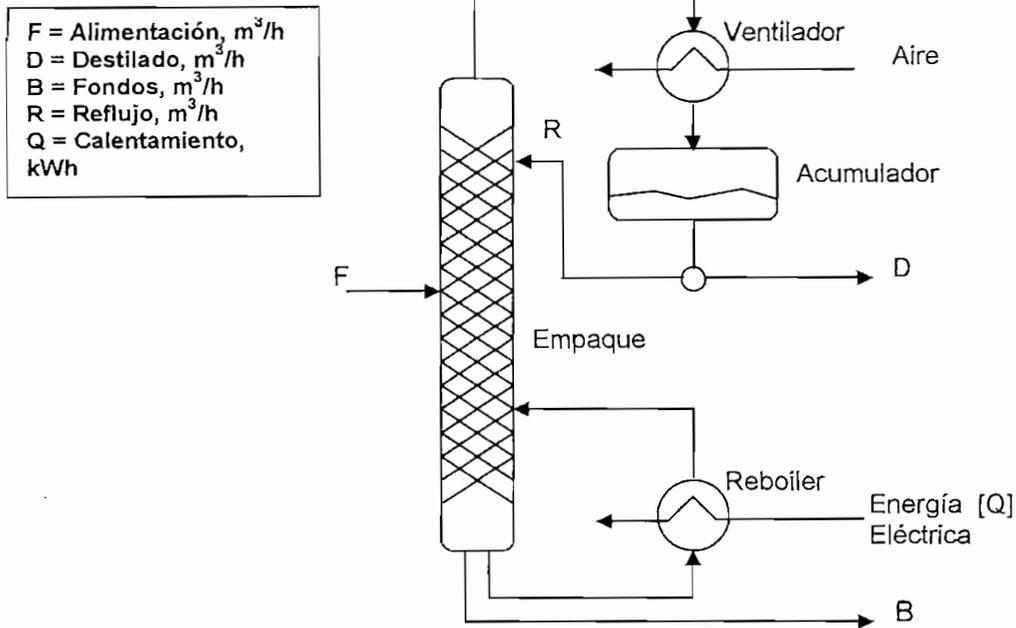


Figura 2.8

Se ha considerado el obtener alcohol etílico al 97% (aziótropo) a partir de una mezcla binaria (agua – alcohol) en una destilación a temperatura variable y presión atmosférica. En la figura 2.8 el fluido a ser separado ingresa por la porción media de la torre y se desliza a la parte inferior denominada calderín, donde hierve generando vapor, el mismo que asciende a lo largo de la torre por los intersticios que quedan entre las partículas del empaque. Al chocar se produce la separación del agua y el alcohol por causa de la diferencia de los puntos de ebullición de las dos sustancias. El vapor con diferente contenido de agua se va purificando y enriqueciendo en alcohol. En efecto se reconocen dos secciones de la torre: la zona de agotamiento (bajo la línea de alimentación) y la zona de enriquecimiento (sobre la línea de alimentación). El vapor enriquecido en alcohol es enfriado y condensado en un dispositivo (condensador) que utiliza el aire ambiental que circula a través de las placas en un enfriador que dispone de elevada área de transferencia de calor (radiador). El fluido enfriado llega al acumulador del mismo que salen dos corrientes: una que regresa al tope de la torre (reflujo) y otra que constituye el producto (destilado). La corriente de reflujo aumenta el

enriquecimiento de alcohol en el vapor que asciende al tope de la torre, efectuando un lavado en contra corriente que arrastra el agua residual.

## **2.2 OBJETIVOS**

Una de las principales razones por la cual se decidió realizar el proyecto, es porque para la operación de la torre de destilación, en un proceso denominado batch o por tandas se requiere de la presencia de dos personas para un correcto funcionamiento, mientras que al automatizar parte del proceso, pese que el mismo sea semicontinuo, sólo se va a necesitar un operador. Con estos antecedentes y de acuerdo a las distintas etapas de la torre, se tendría que realizar lo siguiente.

1. Controlar la corriente que circula por el dispositivo de calentamiento, para de esa forma garantizar el control de la temperatura de la mezcla dentro del reboiler.
2. Controlar la apertura y el cierre de la válvula del termosifón para de esa manera controlar el nivel de mezcla que existe dentro del reboiler, por medio de la medición del nivel de líquido que exista dentro del reboiler.
3. Controlar que exista un nivel constante en el tanque de alimentación, para de esa forma disminuir las fluctuaciones de caudal que pueden existir; así como el trabajo innecesario de la válvula de alimentación.
4. Controlar tanto la bomba de alimentación como la bomba de cosecha que se encuentran en el acumulador, por medio de la medición del nivel de líquido que exista en el acumulador.
5. Controlar el encendido y el apagado del ventilador.

## **2.3 LOS SENSORES**

De acuerdo con lo expuesto en el numeral anterior y la descripción de las etapas de la torre de destilación, se requiere de dispositivos que permitan realizar la traducción tanto de temperatura como de nivel, para poder medirlas. A estos dispositivos se los denomina sensores y son los encargados de transformar los cambios o una variable física en general, en una señal eléctrica que puede ser

tratada, con la finalidad de conocer el estado del proceso y poder realizar su control.

En el caso del sensor de temperatura se trata de un dispositivo electrónico. En cambio el sensor de nivel es un dispositivo magneto mecánico.

### **2.3.1 SENSOR DE TEMPERATURA**

En la actualidad la tendencia es a realizar todo procesamiento en forma digital: la imagen, el video, el sonido; y es la razón por la cual se ha dado mucha importancia al desarrollo de dispositivos que permitan la transformación de señales analógicas a digitales y aun como en el caso de esta aplicación que entreguen en forma directa la lectura de temperatura en un formato digital.

Dentro de los dispositivos que se han utilizado en el presente trabajo se cuenta con el DS1821, que es un dispositivo de 3 terminales que mediante una programación inicial, entrega lecturas continuas en un formato digital de 8 bits. A continuación se presenta una descripción del funcionamiento de este circuito integrado.

#### **2.3.1.1 CARACTERISTICAS GENERALES DEL CIRCUITO DS1821**

1. No requiere de componentes externos adicionales a excepción de una resistencia de PULL-UP, debido a que la salida del DS1821 es de drenaje abierto.
2. Para comunicarse con el DS1821 sólo se necesita un pin de cualquier pòrtico del microcontrolador, debido a que trabaja en base al protocolo 1-WIRE que se detallará más adelante.
3. Opera en un rango de temperatura que va desde los  $-55^{\circ}\text{C}$  a los  $+125^{\circ}\text{C}$ .
4. Puede ser la referencia del sistema para que éste funcione como un termostato, pudiendo definirse los valores de temperatura a los cuales se desconecta, o activa el dispositivo de calentamiento.

5. También funciona como termómetro en donde devuelve 8 bits que representan la temperatura a la que se encuentra el dispositivo.
6. Le toma como máximo 1 segundo el convertir una temperatura a una palabra digital de 8 bits.
7. Esta disponible en dos encapsulados PR35 de 3 pines y SOIC de 8 pines, mostrados en la figura 2.9.

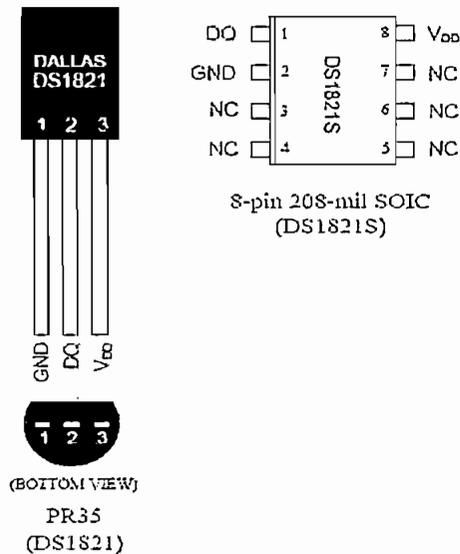


Figura 2.9

### 2.3.1.2 DESCRIPCION DE LOS PINES

El circuito que se ha utilizado en la aplicación es el de encapsulado PR35, por la facilidad de su instalación en la torre de destilación, en el mismo sus pines corresponden a lo siguiente, de acuerdo con lo especificado por el fabricante Dallas Semiconductors; sin embargo, en los anexos se presenta las hojas de datos completas de este circuito.

1. GND Terminal de polarización que se conecta a tierra.
2. VDD Terminal de polarización que se conecta a la fuente de voltaje.

3. DQ                    Para la aplicación en un termostato funciona como una salida y como termómetro, funciona como bus bidireccional de datos.

### 2.3.1.3            DESCRIPCION GENERAL

Como se indicó el circuito DS1821 se le puede hacer funcionar de dos formas:

1. En aplicaciones donde el sistema opera como termostato; ó
2. Como el elemento de medida de la temperatura; es decir, como un termómetro.

Como termostato el usuario puede programar los puntos de disparo, los cuales se guardan en una memoria no volátil dentro del dispositivo. Muy similar al dispositivo mecánico de una plancha eléctrica. Si el dispositivo se calienta en exceso y supera un nivel establecido se envía una señal para desconectar al dispositivo de calentamiento. En cambio si desciende a un nivel de temperatura por debajo del programado, se envía una señal para que lo vuelva a conectar.

También funciona como el elemento central de un termómetro; en este caso el pin DQ funciona como un bus bidireccional usando el protocolo 1-WIRE que se describirá en detalle más adelante.

Cuando el terminal DQ es usado como bus de comunicaciones se puede ingresar a la memoria no volátil del DS1821 y programar por ejemplo, los puntos de disparo cuando funciona como termostato o la palabra de estado cuando funciona como termómetro.

A continuación en la figura 2.10 se presenta un diagrama de bloques dado por el fabricante, el cual describe la arquitectura interna del circuito DS1821.

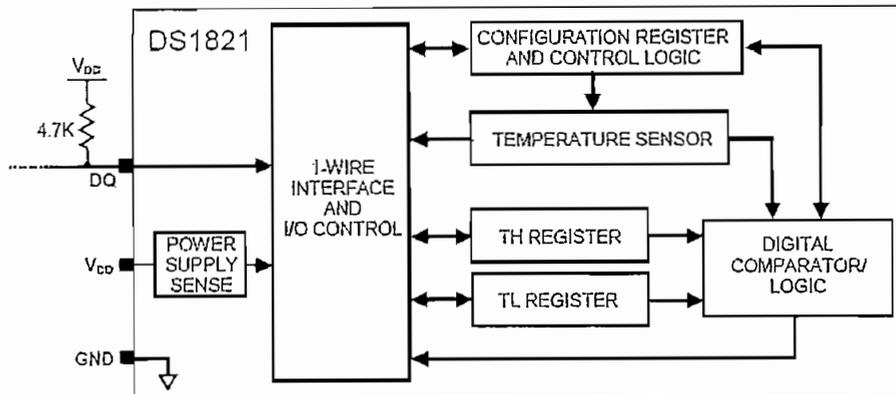


Figura 2.10

### 2.3.1.4 OPERACIÓN DEL DS1821 PARA MEDICIONES DE TEMPERATURA

Uno de los aspectos importantes en la operación del DS1821, es su característica de convertir cualquier temperatura que esté en el rango de  $-55^{\circ}\text{C}$  a  $125^{\circ}\text{C}$  directamente en una palabra digital de 8 bits.

Pero ¿Cómo se realiza esto? En la figura 2.11 se muestra un diagrama de bloques del circuito de medición de temperatura. Este circuito mide la temperatura contando el número de ciclos de reloj generados por un oscilador de bajo coeficiente de temperatura, dentro de un tiempo determinado por otro contador cuyos pulsos de reloj son generados por un oscilador de coeficiente de temperatura mayor.

El contador, cuyos pulsos de conteo son generados por el oscilador con menor coeficiente de temperatura, es cargado con un valor que corresponde a  $-55^{\circ}\text{C}$ . Si el contador llega a cero antes del tiempo determinado por el oscilador de coeficiente de temperatura mayor, el registro de temperatura que estaba cargado con el valor de  $-55^{\circ}\text{C}$  ahora se incrementa en un grado, y se vuelve a cargar el contador con otro valor que corresponde a la siguiente temperatura. Todos estos valores para cargar el contador se encuentran en una tabla en el circuito acumulador (slope accumulator). Cada valor a ser cargado en el contador es único

para cada incremento de temperatura y compensa el comportamiento no lineal que presenta las oscilaciones versus la temperatura.

Este proceso de cargar el contador y cuenta descendente se repite hasta que el tiempo que le toma al contador cuyos pulsos son generados por el oscilador de bajo coeficiente de temperatura, sea lo más cercano al tiempo determinado por el oscilador con coeficiente de temperatura más alto.

Cuando todo este proceso iterativo se haya terminado, quedará almacenado en el registro de temperatura; la temperatura del medio en que se encuentra el dispositivo expresado en grados centígrados.

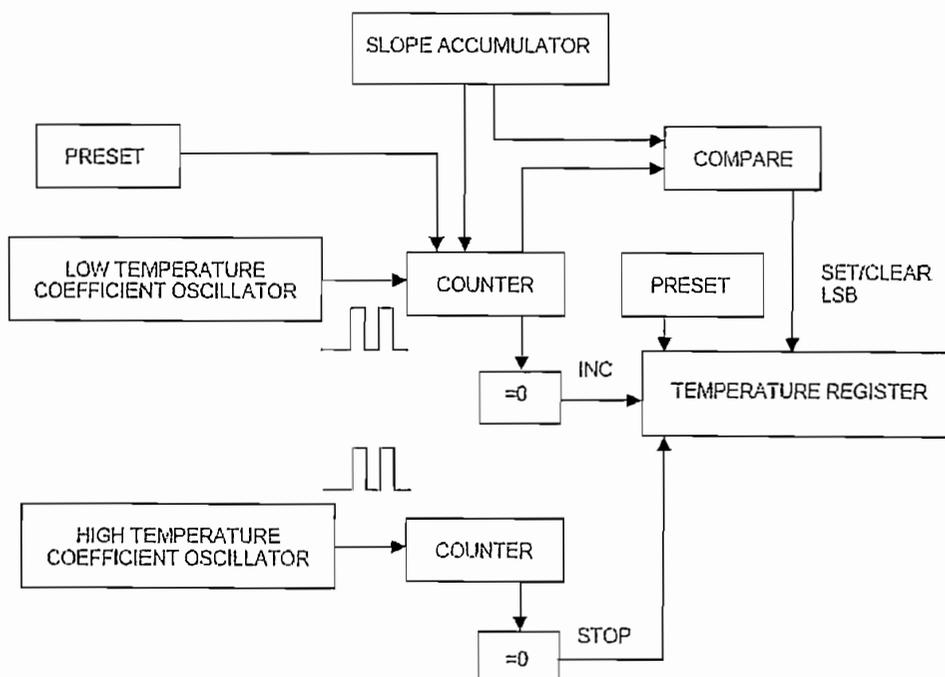


Figura 2.11

En base a lo explicado en los incisos anteriores la salida de datos del DS1821 provee una lectura de temperatura en grados centígrados, y se encuentra almacenada en un formato de 8 bits en el registro de temperatura como puede observarse en la figura 2.12

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

Figura 2.12

EL usuario puede acceder a esta información cuando el bit T/R (ver 2.3.1.5.1) del registro de configuración y de estado se encuentra en 0 lógico. El bit 7 de la figura 2.12 indica si la temperatura es positiva o negativa, para números positivos S=0 y para números negativos S=1. A continuación en la figura 2.13 se presenta un ejemplo de la información que se obtiene en la salida digital y la temperatura que representa, nótese que para los casos en que la temperatura es negativa, su valor se encuentra expresado en dos complementos.

Temperatura	Salida Digital (Binario)	Salida Digital (Hexadecimal)
+125°C	0111 1101	7Dh
+85°C	0101 0101	55h
+25°C	0001 1001	19h
0°C	0000 0000	00h
-1°C	1111 1111	FFh
-25°C	1110 0111	E7h
-55°C	1100 1001	C9h

Figura 2.13

Para el caso de la aplicación, la torre de destilación de alcohol, por lo general estará trabajando en el rango de 25°C hasta los 110°C. Si por ejemplo llegara el número 19H al microcontrolador, bastaría una conversión hexadecimal a decimal y luego transformar cada uno de los números que componen ese número a códigos ASCII para poder visualizarlos en el LCD.

El DS1821 puede ser configurado para que realice mediciones de temperaturas continuas o realizar una sola medición. Todo esto depende del bit 1SHOT (ver 2.3.1.5.1), que se encuentra en el registro de configuración y de estado. Si

1SHOT=0 realiza conversiones continuas, pero si 1SHOT=1 sólo realiza una conversión.

En el modo de conversiones continuas se requiere de un comando para poder iniciar este proceso, y este comando es el EEH (Inicio de conversión T). De igual forma si se desea parar el proceso de conversiones continuas se utiliza el comando 22H (Fin de conversión T). Cuando el DS1821 se encuentra en modo 1SHOT el comando de inicio de conversión sirve para que se inicialice el proceso. Una vez que haya terminado la conversión retorna al estado de espera de algún comando. En este modo se puede chequear el estado de conversión analizando el bit DONE (ver 2.3.1.5.1) del registro de configuración y de estado. Si DONE=0 la conversión no termina, si DONE=1 la conversión se ha completado. Este bit no provee información cuando el dispositivo se encuentra en el modo de conversiones continuas, debido a ello siempre se encontrará en el estado 0 lógico.

### **2.3.1.5 REGISTRO DE CONFIGURACION Y DE ESTADO**

El registro de configuración y de estado del circuito DS1821, provee información al usuario acerca de:

1. Estado de la conversión.
2. Actividad en la EEPROM.
3. Actividad del termostato.

También permite programar al usuario varias opciones del DS1821 como son:

1. Selección de termostato o termómetro.
2. Polaridad de salida del termostato.
3. Modo de conversión.

El registro de configuración y de estado consta de 8 bits; de los cuales el bit 6 no se utiliza, del bit 0 al bit 4 son guardados en la memoria EEPROM, en cambio los bits 7 y 5 solo tienen sentido cuando se esta realizando alguna operación en la

EEPROM. En la figura 2.14 se puede observar en detalle como esta estructurado el registro de configuración y de estado.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DONE	1	NVB	THF*	TLF*	T/R*	POL*	1SHOT*

\*Se almacenan en la memoria EEPROM

Figura 2.14

### 2.3.1.5.1 *DESCRIPCION DE LOS BITS DEL REGISTRO DE CONFIGURACION Y DE ESTADO*

**DONE (Temperatura Conversión Done):** Este es un bit de sólo lectura, indica si se ha terminado la conversión de una temperatura a una palabra de 8 bits. Este bit funciona sólo en el modo 1SHOT

DONE=0      Conversión de temperatura en progreso.

DONE=1      La conversión de temperatura se ha completado.

**NVB (Non-volatile Memory Busy):** Este bit de sólo lectura, indica si se esta efectuando alguna operación en la memoria EEPROM.

NVB=0              La memoria EEPROM no esta ocupada.

NVB=1              Una operación de escritura en la EEPROM se esta ejecutando.

**THF (Temperature High Flag):** Este bit de lectura y escritura, indica si en algún instante de tiempo; la temperatura medida excedió el valor guardado en el registro  $T_H$ , pasando al estado de 1 lógico. Cabe indicar que este bit se mantendrá en 1 lógico, hasta que el usuario lo reescriba a 0 lógico.

THF=0              La temperatura medida no ha excedido el valor del registro  $T_H$ .

THF=1              La temperatura medida excedió el valor del registro  $T_H$ .

**TLF (Temperature Low Flag):** Este bit de lectura y escritura, indica si en algún instante de tiempo; la lectura que se midió fue menor al valor guardado en el registro  $T_L$ , pasando al estado 1 lógico. Cabe indicar que este bit se mantendrá en 1 lógico, hasta que el usuario lo reescriba a 0 lógico.

- TLF=0      La temperatura medida no ha excedido el valor del registro  $T_L$ .
- TLF=1      La temperatura medida excedió el valor del registro  $T_L$ .

**T/R (Power-up Operating Mode):** Este bit de lectura y escritura, permite escoger el modo de operación del DS1821 al momento del encendido. Esto es, para su funcionamiento como termostato o termómetro.

- T/R=0      Modo seleccionado protocolo 1-WIRE.
- T/R=1      Modo seleccionado termostato.

**POL (Thermostat Output (DQ) Polarity):** Este bit de lectura y escritura, permite establecer la polaridad con la que se va activar la salida del termostato.

- POL=0      La salida del termostato al activarse se pone en bajo.
- POL=1      La salida del termostato al activarse se pone en alto.

**1SHOT (Conversion Mode):** Este bit de lectura y escritura, permite seleccionar entre conversiones continuas o una sola conversión.

- 1SHOT=0    Conversiones continuas.
- 1SHOT=1    Una sola conversión.

### 2.3.1.6      **PROTOCOLO 1-WIRE**

El protocolo 1-WIRE, utiliza un solo hilo para comunicar un maestro (microcontrolador) con un esclavo (DS1821).

El DS1821 no es direccionable, es decir que no se pueden conectar varios DS1821 a una misma línea de datos.

Es importante tomar en cuenta que todas las transmisiones y comandos empiezan con la transmisión del bit menos significativo.

A continuación se describe el Protocolo 1-WIRE desde el punto de vista de sus tres aspectos importantes de su funcionamiento, esto es:

1. Hardware.
2. Secuencia de transacción.
3. Tipos de señales del protocolo 1-WIRE.

### 2.3.1.6.1 *HARDWARE*

El bus 1-WIRE por definición tiene una sola línea de datos. Eso implica que ambos dispositivos, tanto el maestro como el esclavo, necesitan que el pin que tienen conectado al bus sea de tres estados. De esa forma se puede recibir o transmitir datos por la misma línea.

La entrada al puerto 1-WIRE del DS1821 es un MOSFET con drenaje abierto, con un circuito interno equivalente al de la figura 2.15.

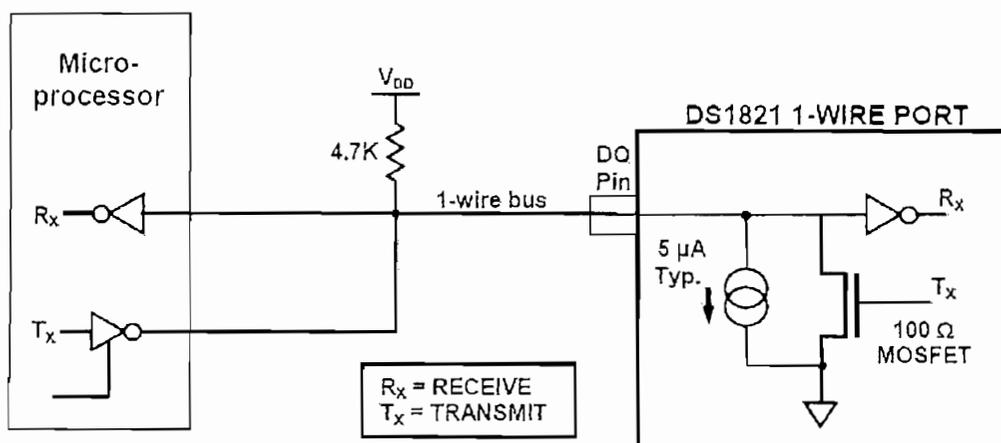


Figura 2.15

### **2.3.1.6.2      *SECUENCIA DE TRANSACCIÓN***

La secuencia que se debe establecer para acceder al puerto 1-WIRE del DS1821 es la siguiente:

1. Inicialización.
2. Enviar los comandos de funciones del DS1821.
3. Transmisión/Recepción de datos.

#### **2.3.1.6.2.1      *INICIALIZACION***

Toda transferencia de información sobre el bus 1-WIRE empieza con una secuencia de inicialización. La secuencia de inicialización consiste en un pulso de reset que transmite el maestro seguido de un pulso de presencia que transmite el DS1821. El pulso de presencia permite informar al maestro que el DS1821 esta en el bus y listo para operar.

#### **2.3.1.6.2.2      *COMANDOS DE FUNCIONES DEL DS1821***

Los siguientes comandos permiten al maestro comunicarse y configurar el DS1821.

**LEER TEMPERATURA [AAH]:** Permite realizar una lectura del registro de temperatura.

**INICIO DE CONVERSION DE TEMPERATURA [EEH]:** Inicia las conversiones de temperatura. Si el dispositivo esta en el modo 1SHOT (1SHOT=1), solamente una conversión se realizará. Si el modo de conversiones continuas esta activado (1SHOT=0), continuará efectuando conversiones indefinidamente hasta que reciba un comando de detener conversiones de temperatura.

**DETENER CONVERSIONES DE TEMPERATURA [22H]:** Detiene las conversiones de temperatura cuando el dispositivo se encuentra en el modo de

conversiones continuas (1SHOT=0). Este comando no realiza ninguna operación cuando el dispositivo se encuentra en el modo 1SHOT (1SHOT=1).

**ESCRIBIR EN EL REGISTRO TH [01H]:** Aquí es donde el usuario ingresará el valor de la temperatura superior a la cual se disparará el termostato.

**ESCRIBIR EN EL REGISTRO TL [02H]:** Este comando permite ingresar al usuario el nivel de temperatura más bajo al cual se disparará el termostato.

**LEER REGISTRO TH [A1H]:** Permite leer el contenido del registro TH.

**LEER REGISTRO TL [A2H]:** Permite leer el contenido del registro TL.

**ESCRIBIR EN EL REGISTRO DE CONFIGURACION Y DE ESTADO [0CH]:** Mediante este comando el usuario puede configurar e inspeccionar el estado del DS1821.

**LEER EL REGISTRO DE CONFIGURACION Y DE ESTADO [ACH]:** Permite al usuario acceder al contenido del registro de configuración y de estado.

### **2.3.1.6.3      *TIPOS DE SEÑALES DEL PROTOCOLO 1-WIRE***

El DS1821 usa un estricto protocolo de comunicación denominado 1-WIRE que garantiza la integridad de los datos. Muchas de las señales usadas en este protocolo están bien definidas, por ejemplo:

1. Pulso de reset.
2. Pulso de presencia.
3. Escritura de un "0"
4. Escritura de un "1"
5. Lectura de un "0"
6. Lectura de un "1"

Todas las anteriores señales, con excepción del pulso de presencia, son iniciadas por el dispositivo maestro.

### 2.3.1.6.3.1 PROCESO DE INICIALIZACIÓN, PULSOS DE RESET Y DE PRESENCIA

Como se señaló todas las comunicaciones con el DS1821 empiezan con una secuencia de inicialización que consiste en un pulso de reset que transmite el dispositivo maestro, seguido por un pulso de presencia que transmite el dispositivo esclavo, como se puede apreciar en la figura 2.16.

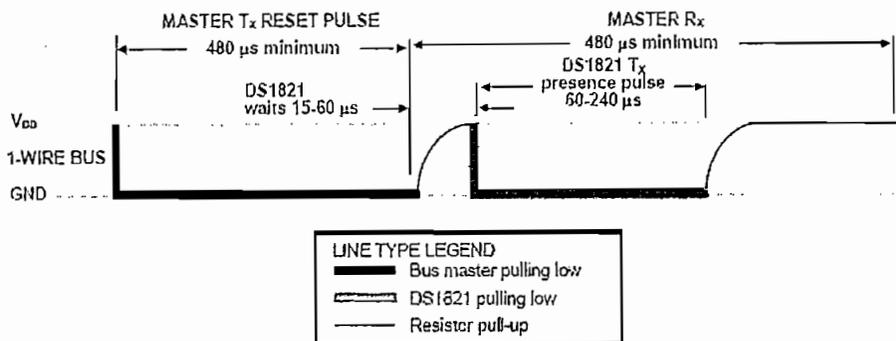


Figura 2.16

Durante la secuencia de inicialización el dispositivo maestro transmite un pulso de reset. Esto es colocar en el bus un 0 lógico durante un mínimo tiempo preestablecido. Entonces el dispositivo maestro libera el bus y espera recibir el pulso de presencia del DS1821.

Cuando el bus es liberado, la resistencia de PULL-UP de 4.7K $\Omega$ , es la encargada de colocar el bus en el estado de 1 lógico. Cuando el DS1821 detecta este flanco positivo, espera entre 15 y 60us para transmitir su pulso de presencia, que consiste en colocar en el bus un 0 lógico, el cual puede durar entre 60 y 240us.

Todo este proceso, tanto de la detección del flanco positivo, como del envío del pulso de presencia por parte del DS1821, le tiene que tomar como mínimo 480us.

### **2.3.1.6.3.2 SLOTS DE TIEMPO DE LECTURA Y ESCRITURA**

El dispositivo maestro escribe los datos hacia el DS1821, durante los SLOTS de escritura y lee datos desde el DS1821, durante los SLOTS de lectura. Sólo un bit de datos es transmitido por el bus durante un SLOT de tiempo.

#### **2.3.1.6.3.2.1 SLOTS DE TIEMPO DE ESCRITURA**

Hay dos tipos de SLOTS de tiempo de escritura:

1. Escritura de 1 lógico
2. Escritura de 0 lógico

Todos los SLOTS de tiempo deben durar un mínimo de 60us con un mínimo de 1us de separación entre SLOTS. Ambos tipos de SLOTS de escritura, empiezan con un pulso de inicio que genera el dispositivo maestro. Este pulso debe ser mayor a 1us y menor a 15us. Como se indica en la figura 2.17.

Para generar un SLOT de escritura de un 1 lógico, el dispositivo maestro debe liberar el bus dentro de 15us, después de haber generado el pulso de inicio. Cuando el bus es liberado, la resistencia de PULL-UP es la encargada de llevar al bus al estado de 1 lógico.

Para generar un SLOT de escritura de un 0 lógico, el dispositivo maestro debe mantener el bus en un estado de 0 lógico por un mínimo de 60us, después de haber enviado el pulso de inicio.

#### **2.3.1.6.3.2.2 SLOTS DE TIEMPO DE LECTURA**

En esta condición de trabajo DS1821 solo puede transmitir datos, si el maestro genera pulsos de inicio y como ya se indicó el pulso de inicio debe ser mayor a 1us y menor a 15us. Por lo tanto si el dispositivo maestro envía un comando de lectura, debe generar inmediatamente SLOTS de lectura. Todos los SLOTS de

lectura deben durar como mínimo 60us, con un mínimo de 1us de tiempo de recuperación entre SLOTS.

Después de que el dispositivo maestro genera el pulso de inicio, el DS1821 empieza a transmitir un 0 o un 1. El DS1821 transmite un 1 dejando libre el bus; es decir es la resistencia del circuito PULL-UP la encargada de poner un 1 lógico en el bus. Por el contrario para transmitir un cero, se pone el bus a un estado 0 lógico. La salida del DS1821 es valida 15us después de haberse generado el flanco descendente del pulso de inicio. En la figura 2.17 se puede observar los SLOTS de lectura para un 1 y 0 lógico.

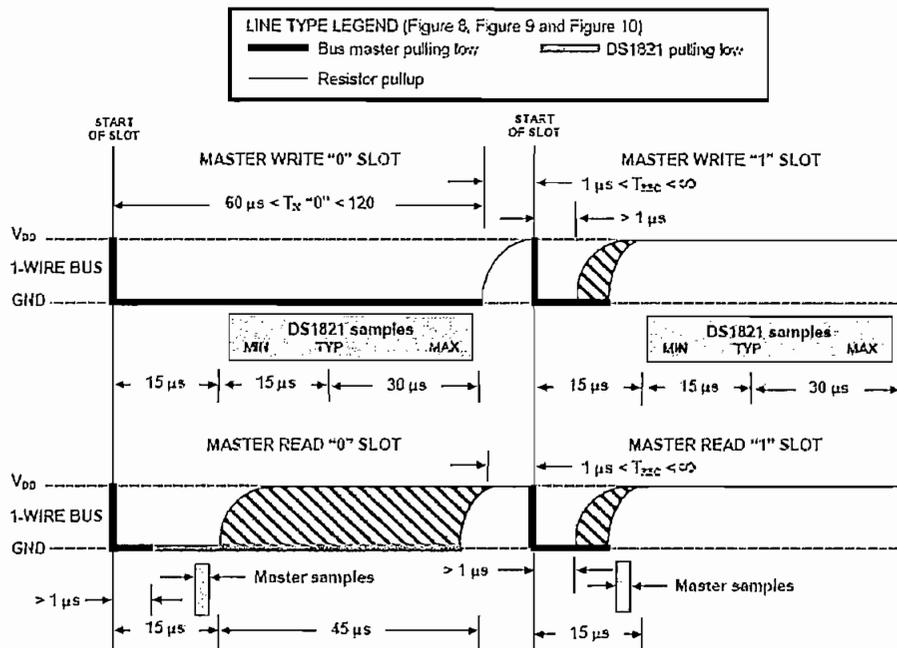


Figura 2.17

### 2.3.2 SENSORES DE NIVEL

En el presente proyecto se necesita determinar el nivel de líquido en cuatro partes específicas de la torre de destilación, que son las siguientes:

1. En el reboiler.
2. En el acumulador.

3. En el tanque de alimentación; y
4. En el tanque de reserva.

En el mercado existen una gran variedad de sensores de nivel. La mayoría de ellos trabajan bajo el mismo principio. Consisten de un material flotante dentro del cual se encuentra uno o varios imanes. Al momento que este material flotante se encuentra sumergido en un suficiente nivel del líquido, se eleva, activando un interruptor magnético que por lo general es normalmente abierto, conociéndose de esa manera que el nivel de líquido ha llegado a un nivel dado.

Tanto en el tanque de reserva como en el de alimentación sólo hay que detectar la presencia de un nivel. En el tanque de reserva hay que detectar cuando se ha terminado el líquido, en cambio al tanque de alimentación hay que mantenerlo a un nivel constante. Los sensores que se ocuparon en el tanque de alimentación y de reserva son los mostrados en las figuras 2.18 y 2.19 respectivamente.

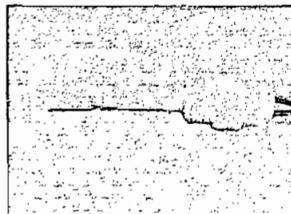


Figura 2.18



Figura 2.19

En el acumulador y en el reboiler hay que determinar más de dos niveles. En el caso del reboiler se necesita establecer 8 niveles pero en el mercado no existe, o al menos no hay disponibles en el Ecuador dispositivos que permitan establecer 8 niveles; por lo que se decidió fabricar uno de la siguiente forma.

El flotador mostrado en la figura 2.20 se lo utilizó como parte principal para la construcción de este dispositivo.

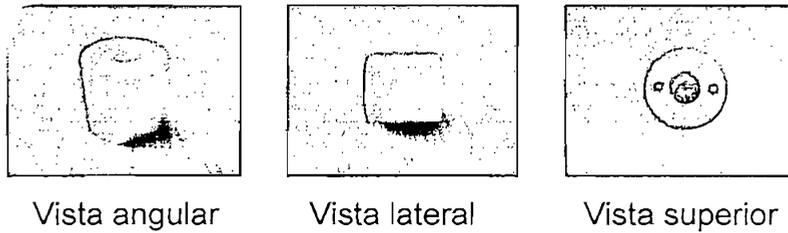


Figura 2.20

Como se puede apreciar en la vista superior de la figura 2.20, los dos puntos negros que se observan en el costado del cilindro, son los imanes que están integrados con el material flotante.

El flotador formaba parte de un sensor de un solo nivel. La característica cilíndrica hueca de este sensor hizo pensar en se podía colocar varios interruptores magnéticos dentro de un tubo de tal manera que el flotador pudiera desplazarse a través de toda la longitud del tubo he ir activando cada uno de los sensores dependiendo del nivel de líquido que se hallará en el reboiler.

El reboiler internamente tiene una altura de 27cm así que se optó por colocar equidistantemente 8 interruptores magnéticos dentro de un tubo de 26.6cm de longitud, como se indica en las figuras 2.21, 2.22 y 2.23.

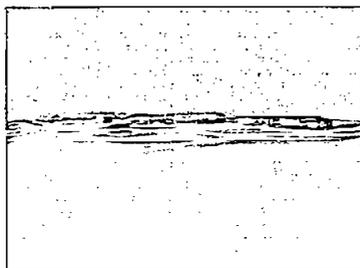


Figura 2.21

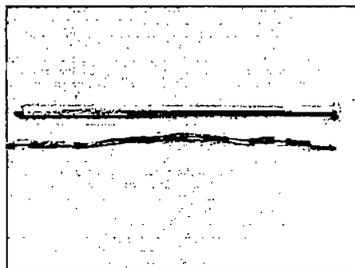


Figura 2.22

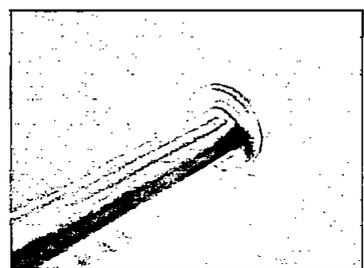


Figura 2.23

El diámetro externo del tubo es de 9mm y el diámetro interno del flotador es de 10mm con ello se garantiza que el flotador se desplace libremente a través de toda la longitud del tubo. Además como se puede ver en la figura 2.23 en la parte superior del tubo se soldó una arandela, como un tope para evitar que el flotador se salga en caso de inundación del reboiler.

Para la construcción del dispositivo de medición de nivel para el acumulador se uso uno de los mismos flotadores pero ya no se determinan 8 niveles como en el reboiler, sino solamente 3 y el tubo donde se colocaron los interruptores magnéticos es de vidrio y no de acero inoxidable como en el reboiler.

### 2.3.3 INTERRUPTORES ÓPTICOS

Para poder detectar el cierre de la válvula del termosifón se colocó un disco opaco en el motor paso a paso (el motor paso a paso está conectado al eje de la válvula para apertura o cierre de la misma), con un corte de 1.5mm en sentido radial esto es para permitir el paso de la luz del interruptor óptico.

Se utilizó dos interruptores ópticos para detectar el cierre del motor. El uno colocado físicamente para cuando la ranura del disco este exactamente en la posición del cierre de la válvula y el otro colocado físicamente 14 pasos después del primer sensor, en el sentido del cierre de la válvula. Si se supone que la válvula esta abierta 3 vueltas y se quiere cerrar la válvula, lo que se hace es enviar las secuencias digitales al motor paso a paso para que gire en el sentido de cierre de la válvula, una vez que se detecta el primer sensor que es el de cierre de la válvula, se le hace girar 14 veces más. Si al llegar al paso 14 se detecta el sensor óptico 2 implica que la válvula no se ha cerrado completamente, así que se vuelve enviar las secuencias digitales para que el motor gire en el sentido del cierre.

Cuando llegue a la última vuelta la fricción del cierre de la válvula no permitirá que el motor gire y alcance a activar el sensor dos; de esta manera se detecta que la válvula se ha cerrado por completo. Después el motor retrocede un poco y se sincroniza con el sensor 1 poniendo los contadores de la subrutina del motor a 0. De ahí en adelante no se permitirá un cierre mayor de la válvula sino el determinado por los contadores, los cuales al llegar a 0 no permiten que la válvula se cierre más.

El principio de funcionamiento del interruptor óptico es sencillo. Se tiene un emisor el cual es el encargado de emitir luz a un fototransistor, el mismo que cuando

recibe luz se satura y cuando no recibe luz entra a la región de corte. Como se puede observar en la parte izquierda de la figura 2.24 el interruptor óptico tiene forma de U de esa forma cualquier material opaco que corte el haz de luz que emite el emisor, hará que el transistor entre a la región de corte. En la parte derecha de la figura 2.24 se observa el disco de color negro que tiene un surco en sentido radial de 1.5mm.

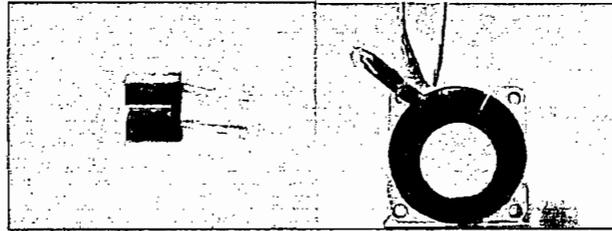


Figura 2.24

### 2.3.4 TECLADO

El teclado es un arreglo de pulsantes en forma matricial, para de esa forma ahorrar líneas de control para su manejo. El teclado utilizado en el presente proyecto es de 16 teclas formado por una matriz de cuatro columnas por cuatro filas. En la figura 2.25 se presenta un esquema de cómo está constituido el teclado y a que pines del microcontrolador esta conectado.

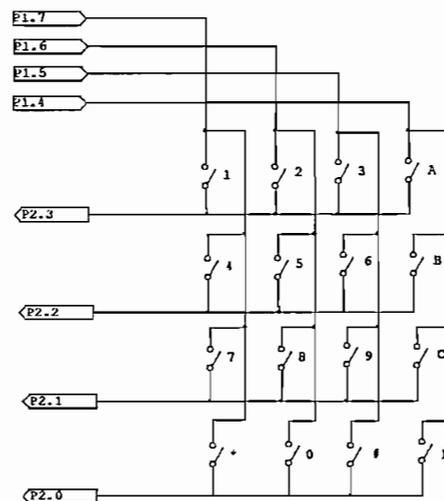


Figura 2.25

### 2.3.5 INTERRUPTORES MAGNÉTICOS

Los interruptores magnéticos al igual que los interruptores manuales presentan oscilaciones en los contactos al momento que entran en contacto los dos metales que realizan la unión. Debido a esto es que se pueden presentar varias señales falsas las cuales son conocidas como rebote de contacto y que pueden ser procesadas por el microcontrolador como verdaderas, generando errores. Para eliminar esto se utilizó en la mayoría de los casos retardos. Lo que se hace cuando se detecta la señal debida a un interruptor es esperar un tiempo mayor al que duran las oscilaciones, para después volver a leer el estado de la señal de entrada y de esa forma comprobar si realmente se ha activado el interruptor. Pero en el caso de los sensores de nivel del tanque de alimentación y el tanque de reserva se optó por eliminar estos rebotes por medio de un circuito y a su vez provocar retardo de 2 segundos para de esa manera hacer al sistema menos sensible a variaciones de nivel.

El circuito para eliminar los rebotes es presentado en la figura 2.26, en el que se puede ver que la señal que llega a la compuerta NAND tipo SCHMITT TRIGGER, es a través de la resistencia R1 y el capacitor C1 que son los que generan el tiempo de retardo cuando sea cerrado el interruptor S1, para que la señal sea considerada un 1L. Si no es así el capacitor se descarga a través del diodo D1 y la resistencia R2.

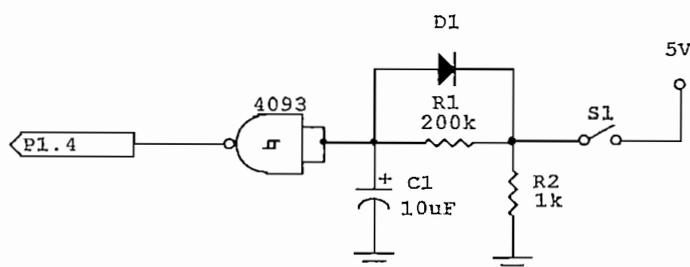


Figura 2.26

Para el cálculo de la resistencia R1, se tomó C1 de 10uF. Como se necesita de un tiempo igual a 2 segundos el valor de la resistencia R1 sería:

$$R1 = \frac{t}{C1}$$

$$R1 = \frac{2s}{10\mu F}$$

$$R1 = 200K\Omega$$

La resistencia R2 debe ser mucho menor que la R1 y para este caso se asumió una resistencia de  $1K\Omega$ .

## 2.4 LOS ACTUADORES

Una vez ubicados en el sistema los sensores requeridos, interesa saber cuales van a ser los dispositivos para actuar sobre el sistema, para de esa forma hacer que el sistema vaya del estado actual al estado deseado.

### 2.4.1 ACTUADOR PARA EL DISPOSITIVOS DE CALENTAMIENTO

Como actuador para el dispositivo de calentamiento se usa un TRIAC. El TRIAC controla la cantidad de corriente que circula a través de la carga en serie conectada al mismo, mediante la activación de la compuerta a determinado ángulo de disparo. El TRIAC es muy usado en la industria porque puede manejar cargas que consumen mucha potencia. Para el presente proyecto se utiliza un dispositivo que tiene la posibilidad de manejar alrededor de 10A.

Los pulsos de disparo hacia la compuerta son enviados por un microcontrolador, el mismo que, para saber cuando tiene que enviar los pulsos necesita de un circuito que le permita saber en que momento se producen los cruces por cero de la señal alterna. Una vez detectado los cruces por cero de la señal alterna, el microcontrolador temporiza el tiempo exacto para enviar el pulso de disparo, de acuerdo a los requerimientos del control que se desea realizar.

En el siguiente subcapítulo se describe detalladamente el circuito detector de cruce por cero.

### 2.4.1.1 DETECTOR DE CRUCE POR CERO

La naturaleza alterna de la señal senoidal, hace que el TRIAC entre en el modo de desactivación en cada cruce por cero. Eso se debe a que la corriente disminuye a un valor menor que la corriente de mantenimiento (Hold Current:  $I_H$ ). Entonces para un adecuado control se necesita conocer cuando ocurren los cruces por cero de la señal, lo cual se consigue con el circuito que se muestra en la figura 2.27.

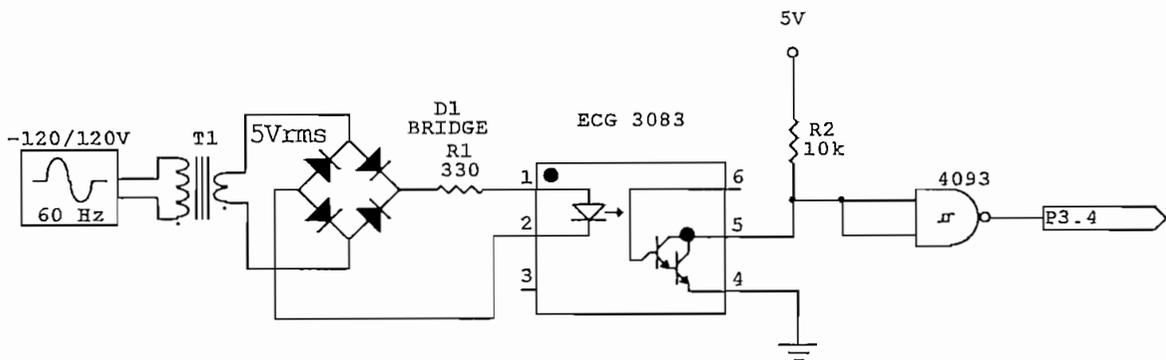


Figura 2.27

El transformador de entrada que se puede observar en la figura 2.27 es un transformador reductor, que funciona con 120V de corriente alterna en el primario y entrega  $5V_{RMS}$  en el secundario; dicho transformador se ha utilizado para que cumpla con las siguientes funciones:

1. Provee un aislamiento magnético.
2. Reduce el voltaje, para manejar valores de potencia bajos.
3. Sincronización con la red.

El transformador por ser de reducción en el secundario, entrega un valor de  $5V_{RMS}$ , lo cual implica que el voltaje máximo en el secundario será:

$$V_{MAX} = \sqrt{2} \times V_{RMS}$$

$$V_{MAX} = \sqrt{2} \times 5V$$

$$V_{MAX} = 7.07V$$

Como puede verse en la figura 2.27 la señal alterna del secundario ingresa a un puente rectificador de onda completa para eliminar los semiciclos negativos de la señal alterna para después enviar esta señal rectificada al diodo emisor de luz del optoacoplador ECG3083.

Para el cálculo de la resistencia limitadora de limitación de corriente hacia el LED del optoacoplador se necesita el valor DC de la señal rectificada; dicho valor de voltaje DC será igual a:

$$V_{DC} = \frac{2(V_{MAX} - 2V\gamma)}{\pi}$$

$$V_{DC} = \frac{2(7.07V - 2 \times 0.7V)}{\pi}$$

$$V_{DC} = 3.6V$$

El voltaje que cae en el LED del optoacoplador para una luminosidad aceptable es de 1.2V como puede verse en las características del optoacoplador ECG3083 en un manual; además, se puede encontrar que el LED de este optoacoplador puede soportar una corriente directa máxima de 60mA. Si asumimos que el LED de este optoacoplador trabajará con una corriente de 7mA se tiene:

$$R_{LIMITADORA} = \frac{V_{DC} - V_{LED}}{7mA}$$

$$R_{LIMITADORA} = \frac{3.6V - 1.2V}{7mA}$$

$$R_{LIMITADORA} = 342\Omega$$

Aproximando el valor de la resistencia limitadora a valores estándar se tiene:

$$R_{LIMITADORA} = 330\Omega$$

La señal que se obtiene a la salida del puente rectificador es una señal, que estará variando entre 0 y un voltaje pico máximo (considerando la caída de voltaje en los diodos) de 5.67V, como puede verse en la figura 2.28. El LED del optoacoplador

mantendrá al fototransistor del mismo en el estado de SATURACION, siempre que la corriente que se le suministre al LED, se encuentra de los valores requeridos, como para emitir la luz necesaria para mantener al fototransistor en dicha región de trabajo.

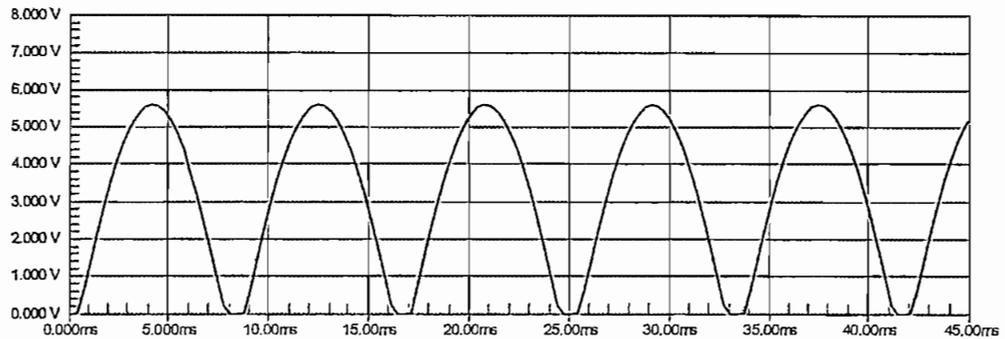


Figura 2.28

Si la corriente que llega al LED del optoacoplador es muy baja, que de hecho va a llegar a ser cero en los cruces por cero, el LED no va a poder emitir luz y el transistor entrará en la región de CORTE; de esta manera se ha conseguido el circuito que detecta cuando ocurren los cruces por cero de la señal alterna.

El optoacoplador usado en el circuito de la referencia para la detección del cruce por cero es el ECG3083; se utilizó este optoacoplador ya que el fototransistor asociado, es un DARLINGTON, y la característica de estos transistores es que son muy sensibles a incidencia luminosa de entrada, dado por su ganancia de corriente muy alta. La configuración con la que se trabajó para el fototransistor DARLINGTON es emisor común, la cual tiene la característica de invertir la señal de entrada. La forma de onda que se tiene a la salida del colector del fototransistor del optoacoplador es la que se ilustra en la figura 2.29.

Para mejorar la señal que se obtiene a la salida del fototransistor del ECG3083, se ha colocado una compuerta NAND con entrada SCHMITT TRIGGER (CMOS 4093), la misma que además de volver a invertir la señal, permite obtener una señal de onda cuadrada; es decir, un tren de pulsos periódicos como puede observarse en la figura 2.30, señal que puede ingresar a cualquier pin del microcontrolador.

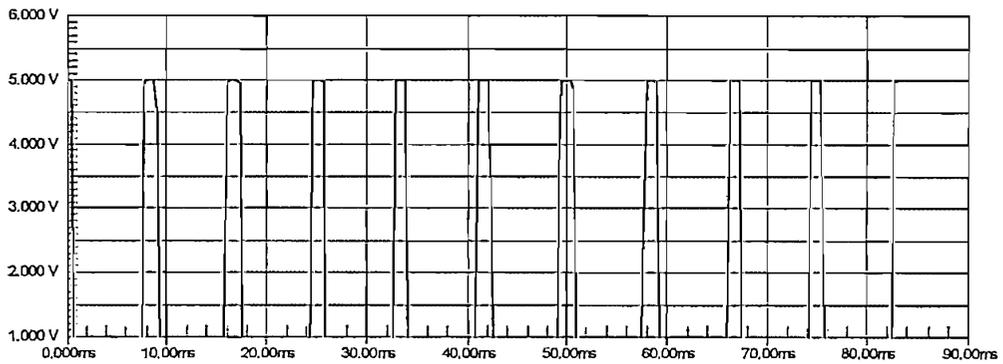


Figura 2.29

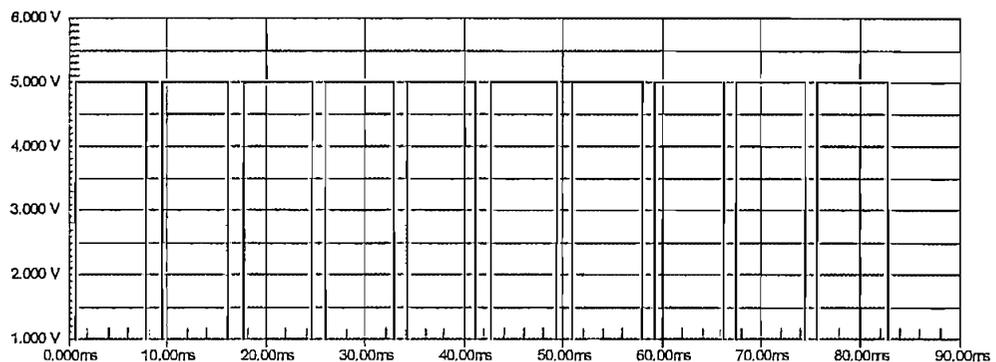


Figura 2.30

Pero para la aplicación se necesita conocer de manera exacta que tiempo pasa la señal de la figura 2.30, en el estado de 1 lógico y 0 lógico.

Debido a ello se implemento el circuito de la figura 2.27, el mismo que trabaja en conjunto con un programa que se lo puede encontrar en los anexos, para obtener una tabla de valores, con la que se determinó el tiempo que pasa la señal de la figura 2.30 en el estado de 1 lógico.

A efecto de la medición requerida, se tomaron 100 muestras, cuyos valores se encuentran en la tabla 4.1 del capítulo 4 "Pruebas y Mediciones del Equipo Construido", además el circuito y el programa para generar la tabla se encuentran en el anexo 5.

De todos los valores se utiliza el menor de todos ellos para realizar los cálculos, pues si se desea disparar el dispositivo a los 180 grados; tomando el valor más grande; podría ser que en ciertos semiciclos el dispositivo no se dispare a los 180 grados del semiciclo presente, sino a los 5 grados del siguiente semiciclo, produciéndose un efecto no deseado.

De todos los valores tomados el menor fue el de 0671H que transformado a número decimal es igual a 1649. Ahora este valor hay que multiplicarlo por el tiempo que toma ejecutar una vez el lazo generado en el programa de control y esto es:

$$t_{1\_LÓGICO} = 1649 \times 4.34 \mu s$$

$$t_{1\_LÓGICO} = 7.16 ms$$

Como era de esperarse el valor encontrado es muy cercano al inverso de la frecuencia de la red eléctrica dividido para dos esto es:

$$t_{MEDIO\_CICLO\_RED} = \frac{1}{\frac{f_{RED\_ELECTRICA}}{2}}$$

$$t_{MEDIO\_CICLO\_RED} = \frac{1}{\frac{60 Hz}{2}}$$

$$t_{MEDIO\_CICLO\_RED} = 8.33 ms$$

La diferencia entre el tiempo que dura medio ciclo de la señal sinusoidal de la red eléctrica, y el encontrado por medio del detector de cruce por cero de la figura 2.27, da el tiempo que la señal de la figura 2.30 se encuentra en el estado 0 lógico.

$$t_{0\_LÓGICO} = t_{MEDIO\_CICLO\_RED} - t_{1\_LÓGICO}$$

$$t_{0\_LÓGICO} = 8.33 ms - 7.16 ms$$

$$t_{0\_LÓGICO} = 1.17 ms$$

Para el desarrollo del programa que controla el ángulo de disparo del tiristor, como se dijo anteriormente se va a tomar el menor de los valores de la tabla 4.1.

Asegurando un margen de tolerancia se fija este valor en 7ms. Ahora este valor de 7ms se lo divide entre 180 grados. El resultado de esta división es el que se puede multiplicar por un factor entre 0 y 180 que el usuario ingresará por teclado, o como en el caso de este proyecto se ingresa un número entre 0 y 100 que representa un porcentaje. En el receptor se realiza la equivalencia para cada número entero comprendido entre 0 y 100 a un factor que va de 0 a 180.

Por tanto, el tiempo para un grado será el siguiente:

$$t_{1\_GRADO} = \frac{7ms}{180}$$

$$t_{1\_GRADO} = 38.9\mu s$$

Pero como además del lazo existen otras instrucciones en el programa, se decidió realizar un ajuste y se fijó este valor en 37us.

$$t_{1\_GRADO} = 37\mu s$$

Entonces cuando el usuario ingrese un porcentaje cualquiera como por ejemplo 50 grados, lo que se hace por programa es multiplicar este factor por el tiempo establecido para un grado que es 37us; entonces, espera un flanco positivo del detector de cruce por cero y dispara el TRIAC a los 1.85ms.

#### 2.4.1.2 CIRCUITO DE DISPARO DEL TRIAC

El TRIAC es un dispositivo semiconductor de cuatro capas, también conocido como tiristor bidireccional, debido a la propiedad que tiene, de conducir la corriente en ambos sentidos. Posee tres terminales los cuales son:

1. Terminal principal 1.
2. Terminal principal 2.
3. Terminal de compuerta.

El símbolo del TRIAC es el mostrado en la figura 2.31, como se puede observar, el terminal de la compuerta va en el lado del terminal principal 1.

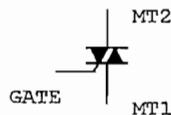


Figura 2.31

Cuando el terminal MT2 es más positivo que el terminal MT1, y se envía un pulso positivo en la compuerta se dispara el TRIAC. Pero si el terminal MT1 es más positivo que el terminal MT2, entonces el TRIAC se dispara con un pulso negativo en la compuerta.

La forma de disparo del TRIAC, anteriormente descrita, es la más segura. Ya que existen otras formas de disparo que son poco seguras, pues producen ruptura del dispositivo o disparos no deseados.

El circuito que se pasa a detallar a continuación sirve para disparar el TRIAC de una manera segura mediante un optoacoplador, el cual es activado por uno de los pines del microcontrolador. Ver la figura 2.32.

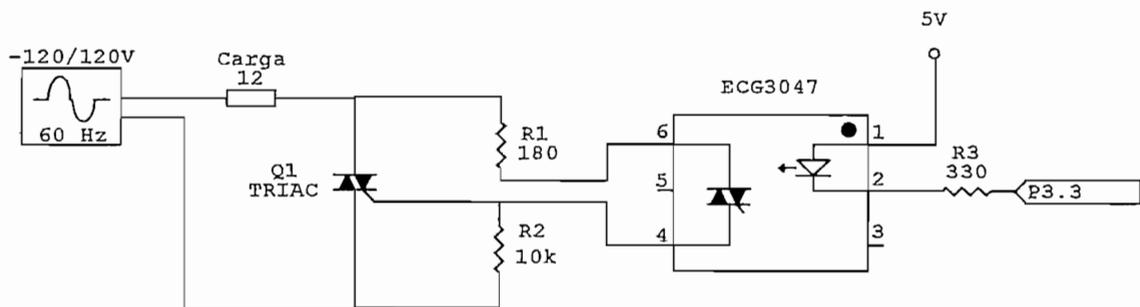


Figura 2.32

Para calcular el valor de la resistencia R3 cuya finalidad es limitar la corriente que pasa por el LED del optoacoplador se tiene que tomar como referencia el valor de la fuente de voltaje es de 5V. En el manual respectivo se puede encontrar que el voltaje que cae en el LED del optoacoplador es de 1.2V y que puede soportar una corriente directa de hasta 50mA.

Para un trabajo seguro del circuito, se hace trabajar al LED del optoacoplador con una corriente de 12mA, por lo tanto el valor de la resistencia limitadora R3 será igual a:

$$R_{LIMITADORA} = \frac{V_{CC} - V_{\gamma}}{12mA}$$

$$R_{LIMITADORA} = \frac{5V - 1.2V}{12mA}$$

$$R_{LIMITADORA} = 317\Omega$$

Aproximando el valor de la resistencia a valores comerciales estándar, se tiene:

$$R_{LIMITADORA} = 330\Omega$$

Para la elección del TRIAC que va a controlar la carga, los parámetros más importantes que se debe tomar en cuenta son: la corriente que va a manejar el dispositivo y el voltaje RMS que maneja. El cálculo de la corriente que va a pasar por el TRIAC es el siguiente:

$$I_{RMS\_TRIAC} = \frac{V_{RMS\_RED}}{R_{LOAD}}$$

$$I_{RMS\_TRIAC} = \frac{120V}{12\Omega}$$

$$I_{RMS\_TRIAC} = 10A$$

Tomando en cuenta el valor de la corriente, seleccionamos el TRIAC ECG56020 el cual soporta una corriente de hasta 25A RMS y un voltaje pico de 400V, los datos completos de este tiristor también se ajuntan en los anexos.

Ahora, se calculará los valores de las resistencias que forman el divisor de tensión que activan la compuerta del TRIAC. El voltaje máximo en la compuerta del TRIAC ECG56020 es de 2.5V. Asumiendo un voltaje de disparo de 2V y una R2 de la figura 2.30 de 10kΩ, se puede calcular el valor de R1, también se ha despreciado la caída de tensión en la carga ya que solamente es de 12 ohmios.

$$R1 = \frac{V_{RMS\_R1} \times R2}{V_{RMS\_RED} - V_{RMS\_R1}}$$

$$R1 = \frac{2V \times 10K\Omega}{120V - 2V}$$

$$R1 = 169\Omega$$

Escogiendo un valor comercial para R1 se tiene:

$$R1 = 180\Omega$$

#### 2.4.2 MOTOR PASO A PASO

Para controlar la apertura y cierre de la válvula del termosifón se optó por usar un motor paso a paso, el cual esta conectado al vástago de la válvula mediante una unión de teflón. Esto se hizo con el objetivo de aislar térmicamente la válvula del motor.

Son motores de los que se puede tener un control muy preciso del grado de rotación de su rotor, aplicando una combinación de niveles lógicos en sus bobinas se logra que recorra cierto número de grados a la vez. Para el caso particular de esta aplicación se está usando un motor unipolar de imán permanente que recorre 1.8 grados por paso.

Este tipo de motores son muy usados en la robótica, impresoras, sistemas donde se requiera movimientos exactos. En la figura 2.33 se muestra el motor paso a paso utilizado en este proyecto.

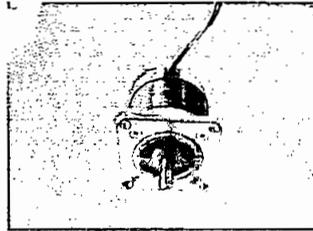


Figura 2.33

#### 2.4.2.1 PRINCIPIO DE FUNCIONAMIENTO

Como la mayoría de motores, consta de un rotor y un estator. En el rotor existen varios imanes permanentes alrededor del mismo. En cambio en el estator existen varias bobinas. Se puede diferenciar claramente dos clases de motores paso a paso de imán permanente, los cuales son:

1. Bipolares.
2. Unipolares.

En la figura 2.34 se presenta el estator de un motor paso a paso. En cambio en la figura 2.35 se puede visualizar el rotor de un motor paso a paso.

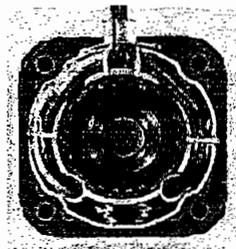


Figura 2.34

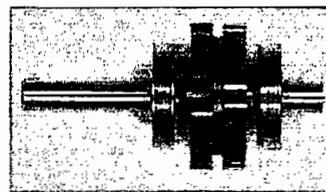


Figura 2.35

Los motores bipolares se caracterizan por tener dos bobinas, los mismos que realizan los giros a través de una secuencia de pulsos, e inversión de la corriente en las bobinas. El control de este tipo de motores se lo hace a través de lo que se denomina un puente H, se necesita un puente H para cada una de las dos bobinas. La forma en como se logra realizar los giros en este tipo de motores es mucho más difícil que la forma como se controlan los giros de los motores unipolares.

Los motores unipolares a diferencia de los bipolares tienen 4 bobinas, al energizar una de ellas se crea un campo magnético que hace que el polo opuesto del imán permanente que se encuentra en el estator, se oriente hacia la bobina energizada. Al ir energizando individualmente cada una de las bobinas, en el orden físico alrededor del estator logramos que el motor gire. La anterior secuencia de activación de bobinas se denomina WAVE DRIVE. Un ejemplo de esta secuencia se presenta en la tabla 2.1, en ella sólo una de las bobinas se energiza.

Paso	Bobina A	Bobina B	Bobina C	Bobina D
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF </td <td>OFF</td> <td>ON</td>	OFF	ON

Tabla 2.1

No es muy recomendada la secuencia de la tabla 2.1 ya que el torque de esta secuencia es mucho menor que el torque obtenido en la denominada secuencia normal. En la figura 2.36 se puede visualizar gráficamente la secuencia WAVE DRIVE.

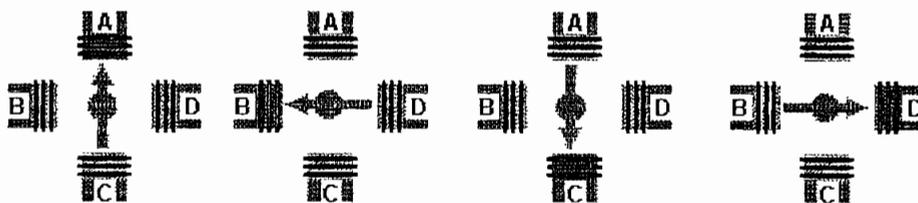


Figura 2.36

La secuencia más recomendable es la llamada secuencia normal, que consiste en activar dos bobinas físicamente adjuntas. Con ello se logra que el rotor gire a la mitad de las dos bobinas energizadas obteniéndose un mayor torque. La secuencia normal se presenta en la tabla 2.2.

Paso	Bobina A	Bobina B	Bobina C	Bobina D
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON

Tabla 2.2

En la figura 2.37 se puede ver gráficamente como es la secuencia normal.

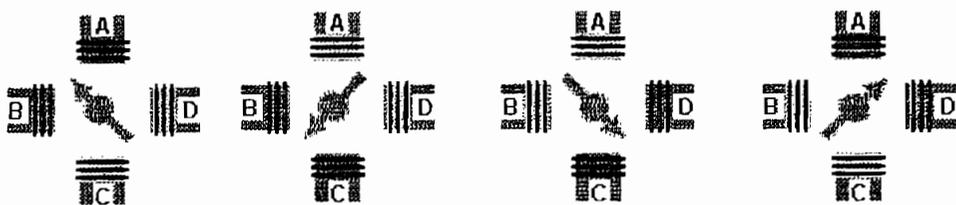


Figura 2.37

### 2.4.3 ELECTROVÁLVULA

El principio de funcionamiento de una electroválvula es muy similar a la de un relay. Lo que se hace es energizar una bobina, la cual genera un campo magnético, la fuerza de este campo magnético vence la fuerza generada por un resorte y desplaza el vástago, en este caso produciendo la apertura de la válvula. Cuando se deja de energizar la bobina, se deja de generar el campo magnético, y la acción del resorte, en sentido inverso, hace que se vuelva a cerrar la válvula.

La electroválvula que se ocupó en este proyecto funciona con 120V de corriente alterna y trabaja conjuntamente con una bomba de 12V para controlar el nivel de líquido que existe en el acumulador. Ambos dispositivos son mostrados en las figuras 2.38 y 2.39 respectivamente.

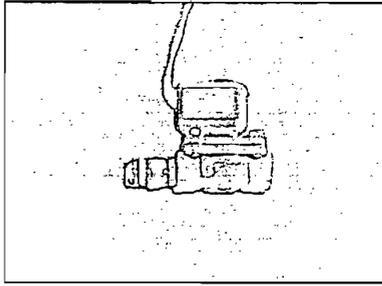


Figura 2.38

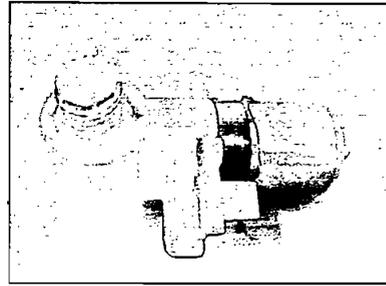


Figura 2.39

Cuando la cantidad de producto en el acumulador excede cierto nivel, se activan simultáneamente la electroválvula y la bomba para poder desalojar una cantidad de producto y volver a dejar a nivel adecuado el acumulador.

#### 2.4.4 BOMBAS

En el presente proyecto se ocuparon dos bombas una utilizada para la realimentación del líquido obtenido en el acumulador hacia la torre, y la otra ubicada en el tanque de reserva para descargar el producto al tanque de alimentación.

Las bombas utilizadas en este proyecto para los propósitos antes señalados, funcionan con 110V de corriente alterna. El principio de funcionamiento de estas bombas es en base a un motor, el cual al activarse, succiona el líquido en el que está sumergida. La bomba que se utilizó tanto en el tanque de reserva se muestra en la figura 2.40.



Figura 2.40

### 2.4.5 LCD

El LCD o display de cristal líquido permiten presentar cualquier caracter del tipo alfanumérico. Cada celda de un LCD consta de una matriz de 7 puntos de alto por 5 puntos de ancho en donde se puede visualizar un caracter. En el presente proyecto el LCD utilizado es de 2 líneas por 16 caracteres por cada línea. El LCD la mayoría de veces consta de 14 pines de los cuales 8 son de entrada y salida de datos, 3 sirven para manejar las comunicaciones en el bus, 2 son para la polarización y el último sirve para manejar el contraste. Hay veces que vienen dos pines adicionales para la luz de fondo.

En la tabla 2.5 se presenta la configuración estándar de los pines de un LCD así como la simbología de cada pin, que es lo que hace cada pin, y si el pin funciona como entrada o como salida.

ASIGNACIÓN DE PINES EN UN LCD			
Pin	Simbología	I/O	Función
1	GND	I	Pin de polarización 0 voltios.
2	VCC	I	Pin de polarización 5 voltios.
3	CONT	I	Pin para regular el contraste su valor va de 0 a 5V.
4	RS	I	0L Entrada de una instrucción. 1L Entrada de un dato.
5	R/W	I	0L Escribir en el módulo LCD. 1L Leer del módulo LCD.
6	E	I	0L Deshabilitación del módulo LCD. 1L Habilitación del módulo LCD.
7	DB0	I/O	Línea 1 del bus de datos. (LSB)
8	DB1	I/O	Línea 2 del bus de datos.
9	DB2	I/O	Línea 3 del bus de datos.
10	DB3	I/O	Línea 4 del bus de datos.
11	DB4	I/O	Línea 5 del bus de datos.
12	DB5	I/O	Línea 6 del bus de datos.

13	DB6	I/O	Línea 7 del bus de datos.
14	DB7	I/O	Línea 8 del bus de datos. (MSB)

Tabla 2.3

### 2.4.5.1 DESCRIPCIÓN DE LOS PINES DE UN LCD

#### 2.4.5.1.1 *PINES GND Y VCC*

Estos dos pines sirven para la polarización del LCD. El voltaje típico de polarización es de 5V. Cabe decir que se necesita un determinado tiempo que esta en el orden de los milisegundos para que el LCD alcance el voltaje de polarización.

#### 2.4.5.1.2 *PIN CONT*

Este pin sirve para manejar el contraste del LCD esto depende del voltaje que se envíe al mismo. Por lo general lo que se hace es conectar los dos terminales no variables de un potenciómetro a los pines de polarización del LCD, para de esa forma el voltaje en el terminal variable del potenciómetro pueda variar entre 0 y 5 voltios, y esta señal es la que se envía al pin CONT.

#### 2.4.5.1.3 *PIN RS*

El pin RS determina si es una instrucción (RS=0L) o un dato (RS=1L) lo que se esta recibiendo a través de los pines del bus de datos.

#### 2.4.5.1.4 *PIN R/W*

El pin R/W determina si se escribirá un dato en el LCD (R/W = 0L) o si se leerá del módulo LCD (R/W), en caso de la lectura del LCD el cursor indica que carácter es el que se va a enviar desde el LCD.

#### 2.4.5.1.5 *PIN E*

Hay veces que se conecta más de un dispositivo al p rtico de un microcontrolador, debido a ello, cada dispositivo conectado al mismo bus tiene que tener un pin que permita poner los pines que est n conectados al bus en alta impedancia. Esto se hace para poder determinar a cual de los dispositivos se desea enviar la informaci n o cual dispositivo se desea que env e un dato. Esto es lo que hace el pin E. Cuando  $E=0L$ , el m dulo se desconecta del bus, pero cuando  $E=1L$  los pines del LCD se conectan al bus.

#### 2.4.5.1.6 *BUS DE DATOS*

El bus de datos de un m dulo LCD es bidireccional. Se puede enviar informaci n al LCD y tambi n podemos recibir informaci n del LCD. Todo depende de c mo se manejan las l neas RS, R/W y E. Una ventaja importante que tienen la mayor a de los LCDs es que se puede trabajar solamente con las 4 l neas m s significativas del bus para enviar datos de 4 en 4 bits, de esta forma se ahorran pines de control del microcontrolador. Todo depende de la inicializaci n del LCD.

Ahora se describir  como act an en conjunto las l neas E, RS, R/W para:

- Escribir una instrucci n en el LCD.
- Escribir un dato en el LCD.
- Leer un dato del LCD.

### 2.4.5.2 **DIAGRAMAS DE TIEMPO DE ESCRITURA Y LECTURA**

#### 2.4.5.2.1 *ESCRITURA DE UNA INSTRucci N*

Para la escritura de una instrucci n hay que poner el bit  $RS=0L$  el bit  $R/W=0L$  y el bit E debe generar un pulso que va de  $0L$  a  $1L$  y luego a  $0L$  que dure por lo menos 450ns. Este diagrama de tiempo es presentado en la figura 2.41.

### 2.4.5.2.2 *ESCRITURA DE UN DATO*

Para la escritura de un dato hay que poner el bit RS=1L el bit R/W=0L y el bit E debe generar un pulso que va de 0L a 1L y luego a 0L que dure por lo menos 450ns. Este diagrama de tiempo es presentado en la figura 2.42.

### 2.4.5.2.3 *LECTURA DE UN DATO*

Para la escritura de una instrucción hay que poner el bit RS=0L el bit R/W=1L y el bit E debe generar un pulso que va de 0L a 1L y luego a 0L que dure por lo menos 450ns. Este diagrama de tiempo es presentado en la figura 2.43.

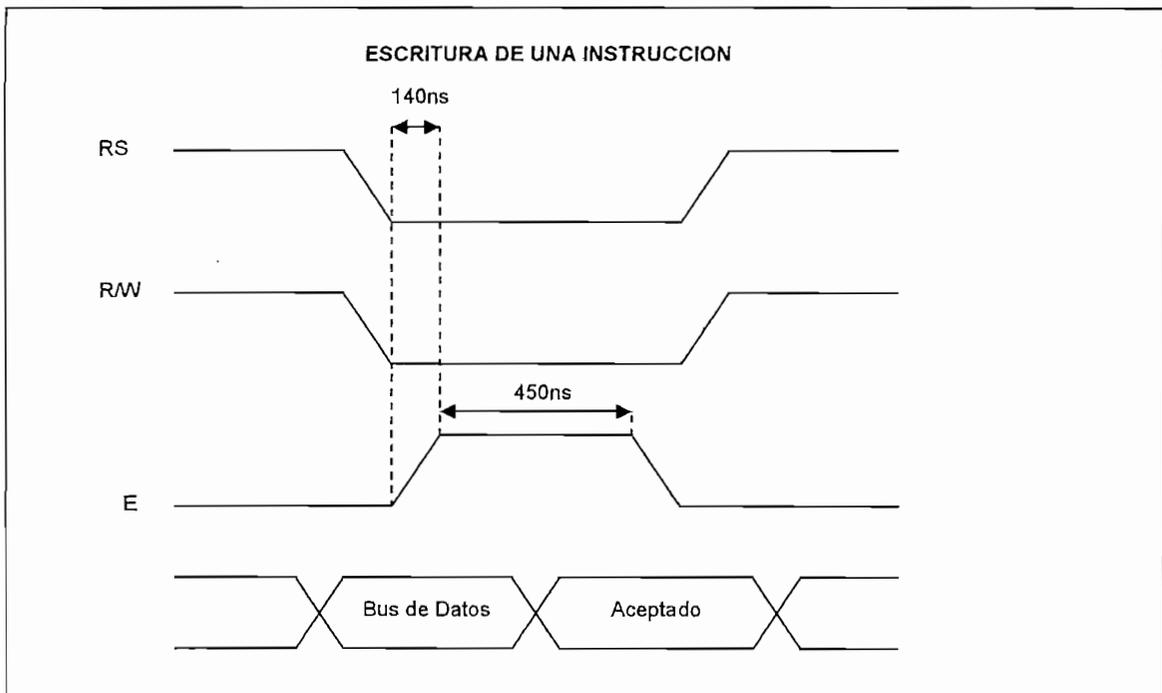


Figura 2.41

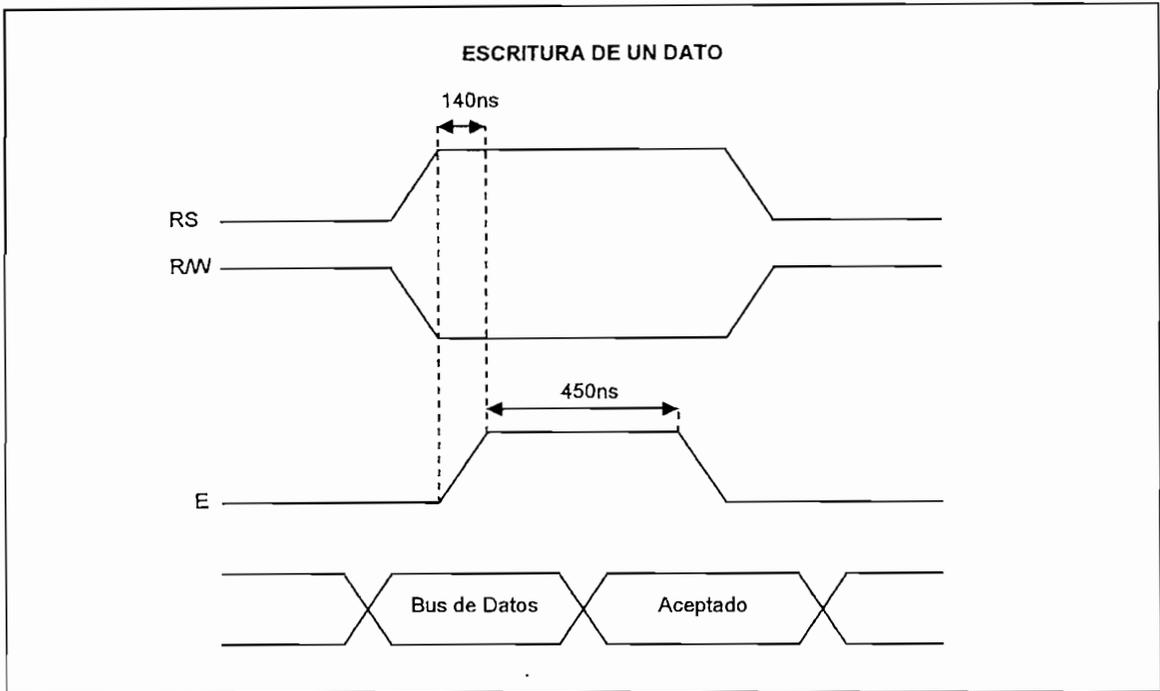


Figura 2.42

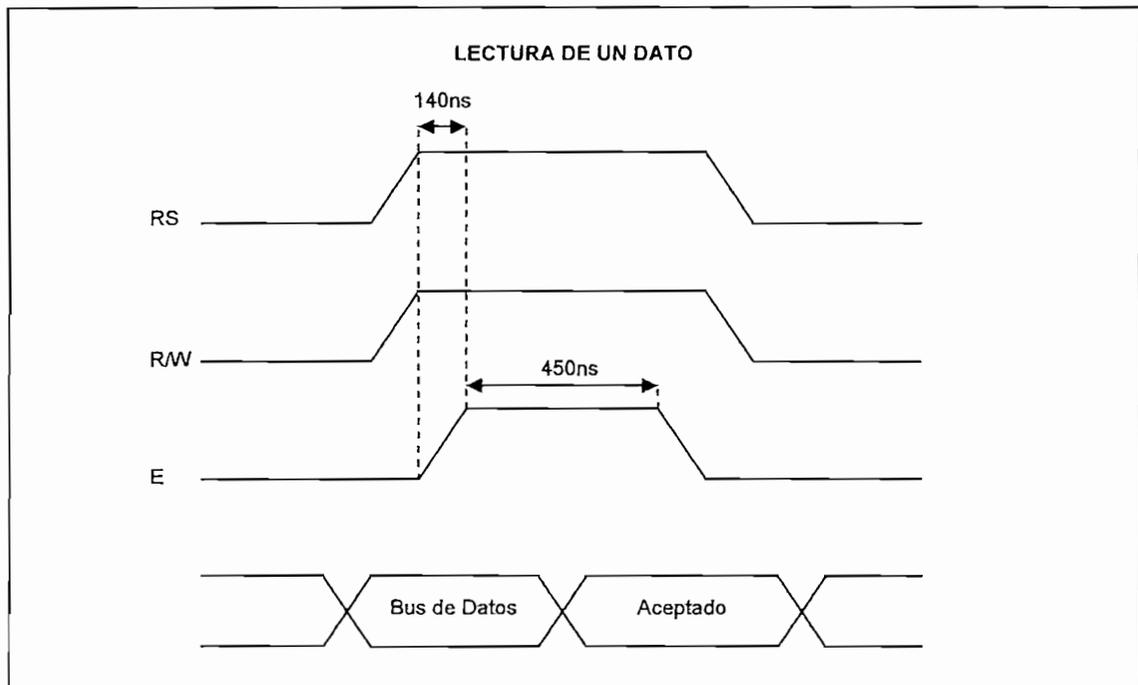


Figura 2.43

### 2.4.5.3 INICIALIZACIÓN DEL MODULO LCD

El módulo LCD necesita ser inicializado, es decir cargarle los valores con los que va a trabajar, por ejemplo: Si al ingresar el carácter este se desplaza a la izquierda o hacia la derecha, si el cursor debe estar titilando o no, si se va a trabajar con un bus de 4 u 8 líneas etc.

Para un ahorro de líneas de control se utiliza la configuración de un bus de 4 líneas de datos. El diagrama de flujo para inicializar el LCD en el modo de un bus de 4 líneas de datos es mostrado en la figura 2.44.

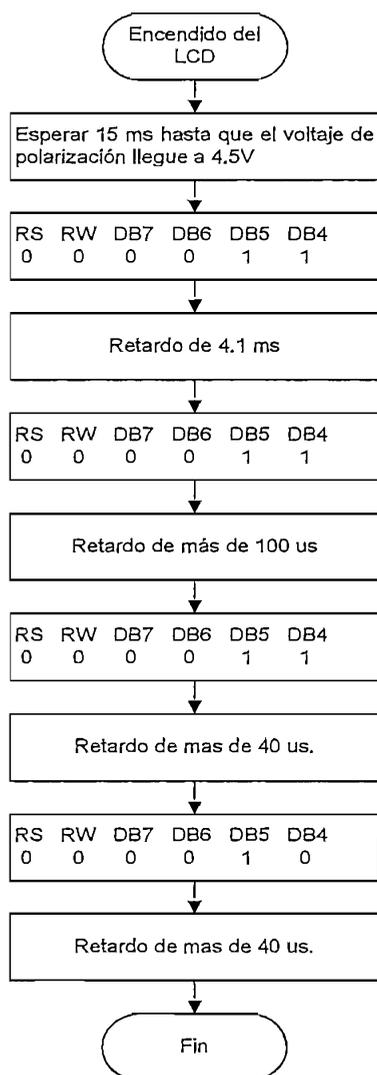


Figura 2.44

#### 2.4.5.4 INSTRUCCIONES DEL LCD

Como se dijo anteriormente además de la inicialización hay que enviar algunas instrucciones al LCD para de esa forma personalizar la presentación de los caracteres en el mismo. Estas instrucciones son diversas, unas sirven para apagar o encender la pantalla, otros manejan el cursor, otros desplazan el mensaje etc. En la tabla 2.4 se muestran algunas de las instrucciones disponibles para los LCDs.

INSTRUCCIONES BÁSICAS DEL LCD											
Descripción	CÓDIGO										Tiempo de ejecución
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Borrar pantalla.	0	0	0	0	0	0	0	0	0	1	1.64 ms
Cursor inicio.	0	0	0	0	0	0	0	0	1	x	1.64 ms
Modo de entrada de caracteres.	0	0	0	0	0	0	0	1	I/D	S	40 us
Apagado y encendido de la pantalla.	0	0	0	0	0	0	1	D	C	B	40 us
Desplazamiento del cursor y del texto.	0	0	0	0	0	1	S/C	R/L	x	x	40 us
Funcionamiento del LCD.	0	0	0	0	1	DL	N	F	x	x	40 us
Escribir en el LCD.	1	0	Escribir el dato.								120 us
Leer del LCD.	1	1	Leer el dato.								40 us

Tabla 2.4

## SIMBOLOGÍA

I/D = 0L      Decremento del cursor en una posición.

I/D = 1L      Incremento del cursor en una posición.

S = 0L        No se desplaza el texto.

S = 1L        Si se desplaza el texto.

D = 0L        Pantalla apagada.

D = 1L        Pantalla encendida.

C = 0L        Cursor apagado.

C = 1L        Cursor encendido.

B = 0L        Intermitencia del cursor encendida.

B = 1L        Intermitencia del cursor apagada.

S/C = 0L     Mover el cursor.

S/C = 1L     Mover el texto.

R/L = 0L     Mover todo el texto a la izquierda.

R/L = 1L     Mover todo el texto a la derecha.

DL = 0L     Bus de datos de 4 bits.

DL = 1L     Bus de datos de 8 bits.

N = 0L        Activación de una línea del LCD.

N = 1L        Activación de dos líneas del LCD.

F = 0L        Matriz del carácter de 5 x 7 dots.

F = 1L        Matriz del carácter de 5 x 10 dots.

### 2.4.6 LED

Para la visualización del encendido o apagado de algunos dispositivos del proyecto se ha optado por utilizar LEDs que son directamente controlados por las salidas del microcontrolador como se puede visualizar en la figura 2.45. Para economizar la corriente que atraviesa por los LEDs en polarización directa se ha optado por limitar la misma a 6 mA por cada LED. Con ello el cálculo para la resistencia limitadora, teniendo en cuenta que la caída de voltaje en el LED es de 1.2V, será:

$$R_{LIMITADORA} = \frac{V_{CC} - V_D - V_{OL(PIN)}}{6mA}$$

$$R_{LIMITADORA} = \frac{5V - 1.2V - 0.2V}{6mA}$$

$$R_{LIMITADORA} = 600\Omega$$

Aproximando el valor de la resistencia limitadora a valores estándar se tiene:

$$R_{LIMITADORA} = 560\Omega$$

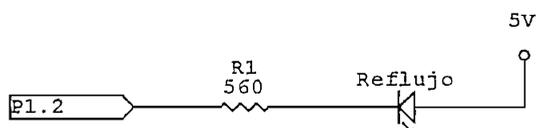


Figura 2.45

### 2.4.7 RELAYS

Para el manejo de dispositivos que funcionan a 120V como son el ventilador y las bombas se utilizaron relays que funcionan con 5V DC la corriente que utilizan estos relays es de 100mA, para el manejo adecuado de los mismos se utilizó el transistor ECG 159. Es un transistor PNP que soporta como máximo 1A de corriente de colector. Con estos datos se necesita calcular la corriente de base para que el transistor entre el estado de saturación. Existen gráficos como el provisto en el anexo 6 que indican las curvas de la corriente de colector con

respecto a la corriente de base y el voltaje colector emisor. Así pues, para una corriente de colector de 100mA en el estado de saturación del transistor, se tiene un voltaje colector emisor de 0.4V, lo cual se consigue con una corriente de base de 4mA. Si se tiene un voltaje de entrada de 0.2V haciendo suma de voltajes de red se tiene:

$$R_{BASE} = \frac{V_B + V_{BE} - V_{CC}}{I_{BASE}}$$

$$R_{BASE} = \frac{0.2V + 0.7V - 5V}{-4mA}$$

$$R_{BASE} = \frac{-4.1}{-4mA}$$

$$R_{BASE} = 1025\Omega$$

Aproximando el valor de la resistencia a valores estándar se tiene:

$$R_{BASE} = 1K\Omega$$

Es importante tener en cuenta que el trabajo del optoacoplador es el de permitir la circulación de corriente a la base del transistor ECG159. La salida del optoacoplador 4N25 soporta como máximo una corriente de 50mA, así que bastará con conectar una resistencia de 10K $\Omega$  al colector del fototransistor, y conectar el colector del mismo a la resistencia de base del ECG159 como se muestra en la figura 2.46

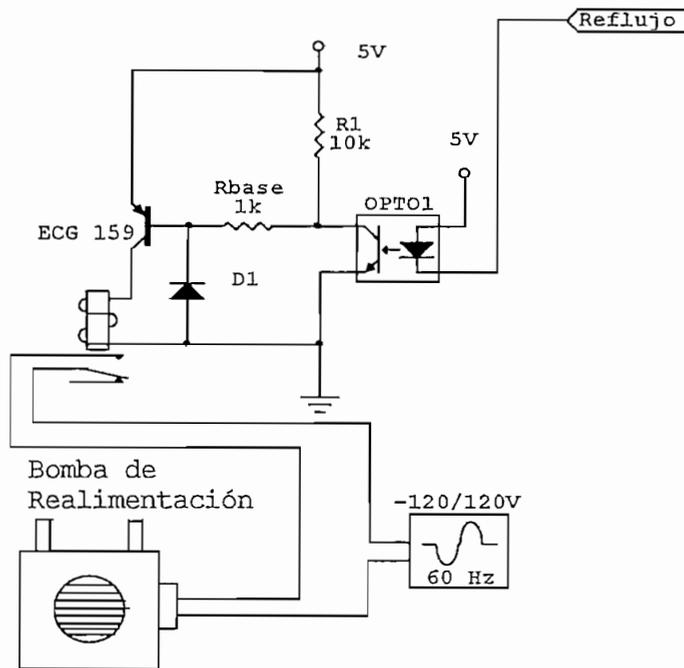


Figura 2.46

Para controlar las bobinas se construyó un circuito similar excepto que se cambio el transistor que controla la bobina ya que el consumo es de 0.8A. El transistor escogido fue el ECG 153 que soporta una corriente máxima de colector de hasta 7A. Este circuito se muestra en la figura 2.47

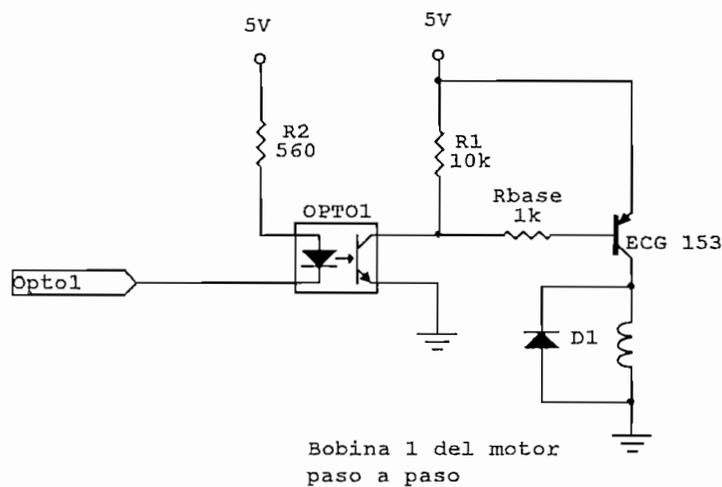


Figura 2.47.

## 2.5 LAS COMUNICACIONES

Las comunicaciones son una parte, de fundamental importancia en cualquier sistema de control. Cabe indicar que el sistema desarrollado no tiene un solo microcontrolador sino varios microcontroladores encargados de las diferentes tareas, es decir es un sistema descentralizado y se necesita un sistema de comunicaciones que permita la comunicación entre ellos. A continuación en la figura 2.48 se presenta un diagrama de cómo están distribuidos los diferentes microprocesadores, sus direcciones y su condición de trabajo como maestro ó esclavo.

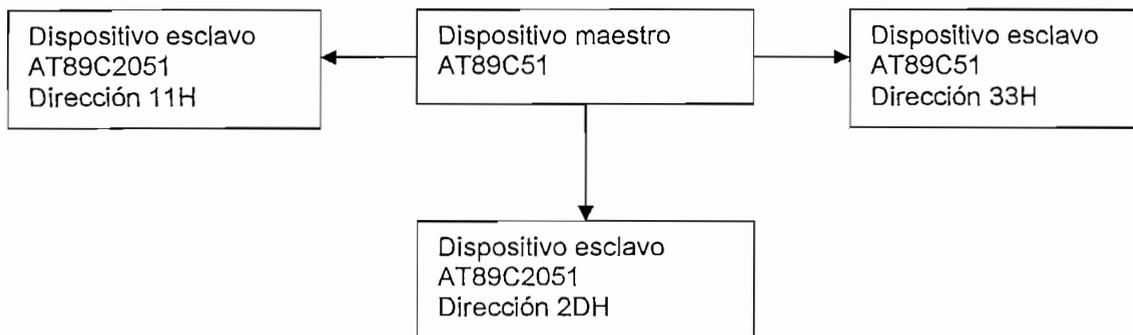


Figura 2.48

Un microcontrolador AT89C51 hace la función de dispositivo maestro y es el encargado de recibir los datos que ingresa el usuario por teclado y de la visualización en el LCD de información tal como: el menú, la entrada de datos y las lecturas del sistema; además, es este el que se puede comunicar con el resto de microcontroladores denominados esclavos que para el caso son los circuitos AT89C2051 y también un AT89C51.

Cuando el maestro desea comunicarse con un esclavo, envía la dirección del esclavo con el que desea comunicarse, ha todos los esclavos. El esclavo que identifica su dirección establece una comunicación con el dispositivo maestro, y sólo finaliza hasta que el maestro decida liberar la comunicación, o hasta que se hayan transmitido cierto número fijo de datos.

La velocidad con la que trabajan todos los microcontroladores utilizados en este proyecto es de 19200bps, que es la máxima permitida por el estándar RS 232.

Los microcontroladores AT89C51 y el AT89C2051 disponen de fábrica de un protocolo de comunicaciones para trabajar en un ambiente multiprocesadores.

Antes de pasar a explicar en detalle como funciona este protocolo, se describirá la interfaz serial de la que disponen estos microcontroladores así como los diferentes modos de comunicación de los que se dispone.

### **2.5.1 INTERFAZ SERIAL**

El p rtico serial es FULL DUPLEX, lo cual significa que puede transmitir y recibir datos simult neamente. Tambi n tiene la caracter stica de RECEIVE - BUFFERED, lo que determina que pueda recibir un segundo byte antes de que el byte previamente recibido haya sido le do desde el registro de recepci n. (Sin embargo, si el primer byte no ha sido le do cuando la recepci n del segundo byte ha sido completada, uno de los bytes se perder ). Tanto para acceder al buffer de transmisi n como al de recepci n del p rtico serial, se utiliza el registro de funci n especial SBUF. Al escribir en el SBUF se carga el registro de transmisi n y al leer el SBUF se accede al registro de recepci n. F sicamente el buffer de recepci n y transmisi n est n separados, pero se puede acceder a ellos por medio del mismo nombre el que es SBUF.

El p rtico serial puede trabajar en los siguientes 4 modos:

#### **2.5.1.1 MODO 0**

Los datos seriales entran y salen a trav s del pin RXD. TXD permite una salida de reloj para la sincronizaci n. Ocho datos son transmitidos o recibidos, empezando por el bit menos significativo. La tasa de barrido queda fijada a un doceavo de la frecuencia de oscilaci n del cristal.

### 2.5.1.2 MODO 1

Diez bits son transmitidos (a través de TXD), o recibidos (a través de RXD): un bit de inicio (0), 8 bits de datos (empezando por el menos significativo), y un bit de parada (1). En recepción el bit de parada es puesto en RB8 del registro especial SCON. La tasa de barrido es variable.

### 2.5.1.3 MODO 2

Once bits son transmitidos (a través de TXD) o recibidos (a través de RXD): un bit de inicio (0), 8 bits de datos (empezando por el menos significativo), un noveno bit de datos programable, y un bit de parada (1). En el transmisor el noveno bit de datos (TB8 en el SCON) le puede ser asignado el valor de 1 o 0; por ejemplo, la paridad (P, en el PSW) puede ser puesto dentro de TB8. En el receptor el noveno bit de datos es puesto dentro de RB8 en el registro especial SCON, mientras que el bit de parada es ignorado. La tasa de barrido puede ser programada a  $1/32$  o  $1/64$  de la frecuencia de oscilación.

### 2.5.1.4 MODO 3

Once bits son transmitidos (a través de TXD) o recibidos (a través de RXD): un bit de inicio (0), 8 bits de datos (empezando por el menos significativo), un noveno bit de datos programable, y un bit de parada (1). En efecto, el modo 3 hace lo mismo que el modo2, con la diferencia aquí que la tasa de barrido es variable.

En los cuatro modos, la transmisión inicia con cualquier instrucción que utilice el registro SBUF como registro de destino. La recepción es iniciada en el modo 0 con la condición que RI=0 y REN=1.

La recepción es iniciada en los otros modos con la llegada del bit de inicio y que el bit REN=1.

En el modo 3 para la transmisión serial se utiliza el TIMER 1 en modo 2, es decir en el modo de autorecarga. En este modo del temporizador el valor de recarga se

coloca en el registro TH1. Entonces apenas termine la temporización TL1, se vuelve a cargar con el valor de TH1.

Para calcular el valor con el que hay que cargar el registro TH1, para obtener una determinada velocidad de transmisión, se utiliza la siguiente fórmula:

$$TH1 = 256 - \frac{2^{SMOD} \times f_{oscilacion}}{384 \times V_{TX}}$$

Donde:

SMOD: Bit programable del registro PCON.

f<sub>OSCILACIÓN</sub>: Frecuencia del cristal.

V<sub>TX</sub>: Velocidad de transmisión.

De ahí la importancia de utilizar cristales con frecuencias de oscilación como por ejemplo 11.0592MHz o 7.3728MHz, ya que estos valores al aplicarlos en la fórmula dan como resultado valores exactos para cargarlos en el registro TH1.

Para el presente proyecto se trabaja con una velocidad de transmisión de 19200bps con un cristal de 11.0592MHz, entonces el valor a cargar en el registro TH1 será.

$$TH1 = 256 - \frac{2^{SMOD} \times f_{oscilacion}}{384 \times V_{TX}}$$

$$TH1 = 256 - \frac{2^1 \times 11059200}{384 \times 19200}$$

$$TH1 = 256 - 3$$

$$TH1 = 253$$

## 2.5.2 COMUNICACIONES EN UN AMBIENTE MULTIPROCESADOR

Los modos 2 y 3 de trabajo tienen una característica especial y es que soportan comunicación multiprocesadores. En estos modos, nueve bits de datos son recibidos, seguidos de un bit de parada. El noveno bit de datos llega y se coloca

en el bit RB8 del registro SCON, entonces llega el bit de parada. El pórtilo puede ser programado de tal manera que cuando el bit de parada es recibido, se produzca una interrupción debida al puerto serial solo si el bit RB8=1. Esta característica esta disponible poniendo SM2=1 en el SCON.

El siguiente ejemplo muestra como utilizar esta interrupción debida al puerto serial para la comunicación multiprocesadores. Cuando el dispositivo maestro tenga que transmitir un bloque de datos a uno de muchos esclavos, primero envía un byte de dirección que identifica al esclavo con el que se desea realizar la transmisión. Un byte de dirección se diferencia de un byte de datos en que el noveno bit es uno en un byte de dirección y cero en un byte de datos. Con SM2=1, ningún esclavo es interrumpido por un byte de datos. Un byte de dirección, sin embargo, interrumpe a todos los esclavos, así que cada esclavo puede examinar el byte recibido y ver si esta siendo direccionado. El esclavo direccionado borra su SM2 y se prepara para recibir los bytes de datos que siguen. Los esclavos que no han sido direccionados colocan a 1 sus bits SM2 e ignoran los bytes de datos.

SM2 no tiene efecto en el MODO 0, pero puede ser usado para chequear la validez del bit de parada en el modo 1. En el modo 1 el receptor, si SM2=1, la interrupción de recepción no es activada a menos que un bit de parada valido sea recibido.

### **2.5.3 CONTROL DE ERRORES**

A pesar de que el medio de transmisión es un medio guiado, y que no se excede el largo del cable definido por el interfaz RS232, que son 15m, no se descarta la posibilidad de que se pueda producir un error en la transmisión de los datos. Así por ejemplo si se querría establecer una comunicación con un dispositivo esclavo cuya dirección es 05h y la dirección de otro dispositivo esclavo es la 04h. Si se produce un error en el primer bit ya no se va a direccionar al dispositivo que tiene la dirección 05h sino al dispositivo que tiene la dirección 04h. Se dispone de 256 códigos diferentes los cuales van desde 00h hasta FFh. En el presente proyecto en la tarjeta principal se tiene disponibilidad para conectar hasta 8 dispositivos

esclavos. De los 256 códigos se escogerán 16 de tal manera que si se producen 1 o 2 errores aislados en la transmisión de uno de los 16 códigos escogidos, nunca se convierta en uno de los 15 restantes.

En el control de errores existen dos grandes grupos de códigos que permiten resolver este problema los cuales son:

1. Códigos de corrección de errores, y
2. Códigos de detección de errores.

Los códigos de corrección de errores se los utiliza cuando el medio de transmisión es no confiable, como por ejemplo el aire. En estos casos se incluye la suficiente redundancia en la información de tal forma que el receptor intuya que es lo que le quiso enviar el transmisor, en este tipo de casos no es conveniente una retransmisión de los datos, porque esta retransmisión también podría llegar con errores.

Los códigos de detección de errores fueron creados para aquellos medios de transmisión en donde los errores ocurren en un bajo porcentaje. Y en donde es mucho más eficiente detectar el error y enviar una retransmisión, que estar enviando un excesivo número de bits adicionales que no son de información, para corregir el error que se produjo.

En telecomunicaciones los códigos de detección de errores más comúnmente utilizados son los CRC o códigos de redundancia cíclica.

### **2.5.3.1 CODIGOS DE REDUNDANCIA CÍCLICA**

En este tipo de códigos se tratan a los unos y ceros como si fueran los coeficientes de un polinomio. Así por ejemplo si se tiene la cadena de bits 10001101 correspondería al polinomio  $x^7+x^3+x^2+1$ . Cabe destacar que todas las operaciones que se realizan en este método son realizadas en módulo 2; es decir, no hay acarreo para la suma ni préstamos para la resta, toda operación de suma

y resta es equivalente a una OR exclusiva. La división se la realiza de la misma forma que en binario, pero al momento de restar se utiliza una OR exclusiva.

El método de los códigos de redundancia cíclica, consiste en adicionar una cadena de bits denominada suma de verificación, a la cadena de datos original. Tanto el transmisor como el receptor acuerdan de antemano, un polinomio al que se le denomina polinomio generador, el cual es de menor grado que el polinomio de datos a enviar. En el transmisor se utiliza el polinomio generador para calcular la suma de verificación. En cambio en el receptor la trama que llega se la divide módulo 2 para el polinomio generador. Si el residuo es cero, implica que la trama no llegó con error. Si el residuo es diferente de cero implicará que la trama tiene errores. Una característica importante del polinomio generador es que siempre los coeficientes del primero y último término deben ser uno.

El estudio de los polinomios generadores está muy avanzada, y se han llegado a crear algunas reglas que determinan que errores se pueden detectar dependiendo de las características del polinomio generador. A continuación se presenta algunas de esas reglas:

1. Si el polinomio generador tiene 2 o más términos puede detectar errores individuales.
2. Si el polinomio generador no divide todos los  $(x^K+1)$  términos, donde K puede valer 1 hasta la longitud máxima de los datos, se puede detectar 2 errores individuales.
3. Si el polinomio generador tiene como factor  $x+1$  se puede detectar un número impar de errores.
4. Se puede detectar errores en ráfaga de igual o menor valor del grado del polinomio generador.

Para el presente proyecto las tramas de datos que se utiliza son de 8 bits, así que para determinar cuáles van a ser los códigos de las direcciones se va a calcular la suma de verificación de los números hexadecimales del 0 a la F.

Si se multiplica 3 veces el factor  $x+1$  el resultado será  $x^3+x^2+x+1$ . Este va a ser el polinomio generador que se utiliza en este proyecto.

A simple vista podemos observar que el polinomio generador escogido tiene más de dos términos, eso implica que detectará cualquier error individual. Por error individual se entiende que si se produce un error en cualquier bit de la trama enviada no se transformará en una palabra código válida.

También el polinomio generador escogido tiene como factor  $x+1$  eso implica que el polinomio generador detectará un número impar de errores. Lo que faltaría por analizar es que si este polinomio puede detectar errores simultáneos en dos bits cualesquiera de la trama enviada. Para determinar esto lo que se hace es dividir los  $(x^k+1)$  términos para el polinomio generador, en donde  $k$  varía desde 1 hasta el número máximo de la trama, es decir 8. El polinomio no tiene que dividir exactamente a cualquiera de los  $(x^k+1)$  términos, para que pueda detectar los errores.

Los resultados siguientes se los obtuvo por medio del uso de algunas funciones de la calculadora programable HP49G, que trabajan en módulo 2 y que se los presenta como ejemplos que ayudan a ilustrar lo explicado.

$$\begin{array}{r|l} x^8+1 & x^3+x^2+x+1 \\ 0 & x^5-x^4+x-1 \end{array} \quad \text{Ecuación 2.1}$$

$$\begin{array}{r|l} x^7+1 & x^3+x^2+x+1 \\ -(x^2+x) & x^4-x^3+1 \end{array} \quad \text{Ecuación 2.2}$$

$$\begin{array}{r|l} x^6+1 & x^3+x^2+x+1 \\ x^2+1 & x^3-x^2 \end{array} \quad \text{Ecuación 2.3}$$

$$\begin{array}{r|l} x^5+1 & x^3+x^2+x+1 \\ x+1 & x^2-x \end{array} \quad \text{Ecuación 2.4}$$

$$\begin{array}{r|l} x^4+1 & x^3+x^2+x+1 \\ 0 & x-1 \end{array} \quad \text{Ecuación 2.5}$$

$$\begin{array}{r|l} x^3+1 & x^3+x^2+x+1 \\ -(x^2+1) & 1 \end{array} \quad \text{Ecuación 2.6}$$

$$\frac{x^2+1}{x^2+1} \Big| \frac{x^3+x^2+x+1}{0} \quad \text{Ecuación 2.7}$$

$$\frac{x+1}{x+1} \Big| \frac{x^3+x^2+x+1}{0} \quad \text{Ecuación 2.8}$$

En la ecuación 2.1 y en la ecuación 2.5 el resultado es 0 eso significa que si se produce un error en el primero y quinto bits, no se detectará el error ya que al invertir ese par de bits, se transformará en una palabra código válida.

Ahora se calculará la suma de verificación desde el 0 hasta la F hexadecimal. Como el polinomio generador es de grado 3 se adiciona 3 ceros al final de la trama pero el cuarto bit que queda libre puede valer cero o uno. Para la codificación de todos los datos de 0 a F el cuarto bit se lo considera un uno lógico.

#### Suma de verificación para F:

11111000	1111	
1111	10001	
00001000		La trama a enviarse para la F será:     11111111
00001111		
00000111		

#### Suma de verificación para E:

11101000	1111	
1111	10010	
0001100		La trama a enviarse para la E será:     11101110
0001111		
00000110		

**Suma de verificación para D:**

11011000		1111	
1111		10111	
001010			
001111			
0001010			La trama a enviarse para la D será: 11011101
0001111			
00001010			
00001111			
00000101			

**Suma de verificación para C:**

11001000		1111	
1111		10100	
001110			La trama a enviarse para la C será: 11001100
001111			
00000100			

**Suma de verificación para B:**

10111000		1111	
1111		11101	
01001			
01111			
001100			La trama a enviarse para la B será: 10111011
001111			
00001100			
00001111			
00000011			

**Suma de verificación para A:**

10101000		1111		
1111		10111		
01011				
01111				
001000			La trama a enviarse para la A será:	10101010
001111				
0001110				
0001111				
00000010				

**Suma de verificación para el 9:**

11011000		1111		
1111		10111		
001010				
001111				
0001010			La trama a enviarse para el 9 será	11011101
0001111				
00001010				
00001111				
00000101				

**Suma de verificación para el 8:**

10001000		1111		
1111		11000		
01111			La trama a enviarse para el 8 será:	11011101
01111				
00000000				

**Suma de verificación para el 7:**

01111000		1111		
01111		1000	La trama a enviarse para el 7 será:	01111000
00000000				

**Suma de verificación para el 6:**

01101000		1111		
01111		1011		
0001000				
0001111			La trama a enviarse para el 6 será:	01101001
00001110				
00001111				
00000001				

**Suma de verificación para el 5:**

01011000		1111		
01111		1110		
001000				
001111			La trama a enviarse para el 5 será:	01011010
0001110				
0001111				
00000010				

**Suma de verificación para el 4:**

01001000		1111		
01111		1101		
001100				
001111			La trama a enviarse para el 4 será:	01001011
00001100				
00001111				
00000011				

**Suma de verificación para el 3:**

11011000		1111		
1111		10111		
001010				
001111				
0001010			La trama a enviarse para el 3 será:	00111100
0001111				
00001010				
00001111				
00000101				

**Suma de verificación para el 2:**

00101000		1111		
001111		111		
0001010				
0001111			La trama a enviarse para el 2 será:	00101101
00001010				
00001111				
00000101				

**Suma de verificación para el 1**

00011000		1111		
0001111		10	La trama a enviarse para el 1 será:	00011110
00000110				

**Suma de verificación para el 0**

00001000		1111		
00001111		1	La trama a enviarse para el 0 será:	00001111
00000111				

Lo que se quiere hacer es crear un conjunto de códigos que van a servir como comandos o direcciones de tal forma que si se produce algún error en estos códigos se asegure que, no se ejecute algún comando o no se direcciona cierto dispositivo.

Es preferible en vez de desarrollar un algoritmo que calcule la suma de verificación, hacer una tabla con los códigos que se han obtenido con el método anterior, de esa forma si el receptor recibe un código diferente al de la tabla sabrá que hubo un error, o simplemente se omitirá el comando. Recopilando todos los datos anteriores se obtuvo la tabla 2.5.

<b>Numero a Codificarse</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Código Hexadecimal</b>
0	0	0	0	0	1	1	1	1	0FH
1	0	0	0	1	1	1	1	0	1EH
2	0	0	1	0	1	1	0	1	2DH
3	0	0	1	1	1	1	0	0	3CH
4	0	1	0	0	1	0	1	1	4BH
5	0	1	0	1	1	0	1	0	5AH
6	0	1	1	0	1	0	0	1	69H
7	0	1	1	1	1	0	0	0	78H
8	1	0	0	0	1	0	0	0	88H
9	1	0	0	1	1	0	0	1	99H
A	1	0	1	0	1	0	1	0	AAH
B	1	0	1	1	1	0	1	1	BBH
C	1	1	0	0	1	1	0	0	CCH
D	1	1	0	1	1	1	0	1	DDH
E	1	1	1	0	1	1	1	0	EEH
F	1	1	1	1	1	1	1	1	FFH

Tabla 2.5

Del análisis hecho con las ecuaciones desde la 2.1 hasta la 2.8 se sabe que si se alteran los bits primero y quinto de cualquiera de los códigos de la tabla 2.5 se

transformará en otra palabra código válida. Por ejemplo la codificación del número 0 según la tabla 2.5 es 0Fh, si se invierte los bits 1 y 5 se obtiene la palabra código 1Eh que corresponde a la codificación del número 1, y por lo tanto no se detectará el error.

Pero, como se indicó, el bit 4 es seleccionable es decir puede valer 0 o 1, si se toma los números impares del 0 a la F y se los vuelve a codificar, pero ahora con el cuarto bit con valor 0 se tendrá lo siguiente.

### Nueva suma de verificación para el 1

00010000		1111	
00011111		11	
00001110			La nueva trama para el 1 será: 00010001
00001111			
00000001			

### Nueva suma de verificación para el 3

00110000		1111	
00111111		101	
00001100			La nueva trama para el 3 será: 00110011
00001111			
00000011			

**Nueva suma de verificación para el 5**

01010000		1111	
01111		1111	
001010			
001111			
0001010			La nueva trama para el 5 será: 01010101
0001111			
00001010			
00001111			
00000101			

**Nueva suma de verificación para el 7**

01110000		1111	
01111		1001	
00001000			La nueva trama para el 7 será: 01110111
00001111			
00000111			

**Nueva suma de verificación para el 9**

10010000		1111	
0001111		101	
00001110			La nueva trama para el 9 será: 10010110
00001111			
00000001			

### Nueva suma de verificación para B

10110000		1111		
1111		1110		
01000				
01111			La nueva trama para B será:	10110100
001110				
001111				
00000100				

### Nueva suma de verificación para D

11010000		1111		
1111		1011		
001000				
001111			La nueva trama para D será:	11010010
0001110				
0001111				
00000010				

### Nueva suma de verificación para F:

11110000		1111		
1111		10000	La trama a enviarse para F será:	11110000
00000000				

De acuerdo con lo expuesto la tabla 2.5, sería reemplazada por la tabla 2.6.

Numero a Codificarse	8	7	6	5	4	3	2	1	Código Hexadecimal
0	0	0	0	0	1	1	1	1	0FH
1	0	0	0	1	0	0	0	1	11H
2	0	0	1	0	1	1	0	1	2DH
3	0	0	1	1	0	0	1	1	33H

4	0	1	0	0	1	0	1	1	4BH
5	0	1	0	1	0	1	0	1	55H
6	0	1	1	0	1	0	0	1	69H
7	0	1	1	1	0	1	1	1	77H
8	1	0	0	0	1	0	0	0	88H
9	1	0	0	1	0	1	1	0	96H
A	1	0	1	0	1	0	1	0	AAH
B	1	0	1	1	0	1	0	0	B4H
C	1	1	0	0	1	1	0	0	CCH
D	1	1	0	1	0	0	1	0	D2H
E	1	1	1	0	1	1	1	0	EEH
F	1	1	1	1	0	0	0	0	F0H

Tabla 2.6

Con los anteriores códigos se asegura que si se invierte 1, 2, 3 o un número impar de bits, nunca se transformará en una palabra de código válida.

El análisis realizado tiene como propósito determinar los códigos que se utilizarán como direcciones de los esclavos y que por tanto van a corresponder para cada esclavo, uno de los códigos de la tabla 2.6. De esa forma el usuario se asegura que está direccionando el dispositivo correcto. Una vez que se tiene la atención del dispositivo esclavo, se utiliza el resto de códigos de la tabla como comandos, para que el dispositivo esclavo realice cualquier tarea previamente especificada.

## 2.6 CONFIGURACIÓN E INTERFACES

La finalidad de este subcapítulo es presentar una breve descripción de como funcionan cada uno de los circuitos implementados en este proyecto. Cabe decir que el procesamiento de la información que llega de los sensores, así como de las señales que se envía a los actuadores, es realizado por medio de los microcontroladores ATMEL 89C51 y 89C2051.

El control de la torre de destilación que se realiza en este proyecto es descentralizado; es decir, todas las señales no llegan a un solo microcontrolador; sino que, a cada microcontrolador se le ha asignado determinada tarea de control. Por ejemplo a uno de los microcontroladores AT89C2051 se le ha asignado la tarea de controlar la temperatura dentro del dispositivo de calentamiento para de esa forma obtener determinadas concentraciones de fondos así como de productos, por medio de un control de la corriente. A un microcontrolador AT89C51 se le destinó para que cumpla la tarea de controlar el nivel de líquido dentro del calderín, por medio de la apertura ó cierre de la válvula de fondos, y al último microcontrolador AT89C2051 se le destinó para que cumpla la función de controlar el nivel dentro del acumulador por medio de la activación de la bomba de cosecha; también el control de nivel del tanque de alimentación así como el de sensar la existencia de líquido en el tanque de reserva y por último es el encargado de encender o apagar el ventilador. En cambio un microcontrolador AT89C51 es el encargado de interactuar con el usuario pues maneja los dispositivos de entrada de datos así como visualización de los mismos y también es el encargado de administrar las comunicaciones. Lo indicado se puede apreciar en la figura 2.49; es decir, cómo están distribuidas las tareas de control.

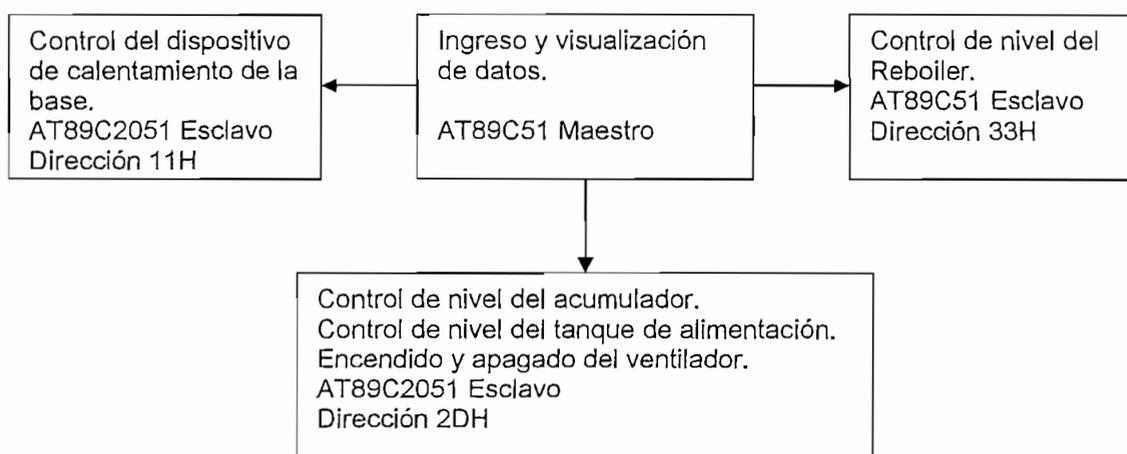


Figura 2.49

### 2.6.1 CIRCUITO DEL PANEL PRINCIPAL

Como se señaló el panel principal es el medio que permite al operador interactuar con el sistema de control, para cuyo propósito se dispone de un teclado y un LCD. El teclado está conectado al pórtilo 2 del microcontrolador AT89C51, también se ha conectado una resistencia de PULL-UP en los pines de este pórtilo, debido a que cuatro de ellos funcionan como entradas.

El LCD está conectado a los pines del pórtilo 1. Se utilizan seis de ellos para su manejo desde P1.2 hasta P1.7. El pin P1.1 sirve para activar un transistor ECG 3906 que trabaja en corte y saturación, el mismo que activa una bocina la cual emite una señal sonora cada vez que se presiona una tecla.

Para recibir las señales de los pines de transmisión de los dispositivos esclavos se utilizó un arreglo de compuertas AND y NAND, el mismo que se realizó con los circuitos integrados ECG 4082 y ECG 4093. Es importante aclarar que cualquier entrada de una compuerta CMOS que no se este utilizando, debe ser conectada a un nivel lógico, porque estas compuertas funcionan con corrientes muy bajas y cualquier ruido eléctrico podría provocar una falsa entrada. A la salida de este arreglo de compuertas NAND de 7 entradas y una salida, se conectó dos compuertas NAND tipo SCHMITT TRIGGER en serie para conseguir una mejor señal al ser mejorados sus flancos. Así mismo se colocó dos compuertas NAND tipo SCHMITT TRIGGER en serie a la salida del pin de transmisión para de esa forma contar con una interfaz que disponga de suficiente capacidad de corriente para poder manejar los dispositivos esclavos. A una de las siete entradas de la compuerta NAND se ha conectado el circuito integrado MAX 232 para en el futuro poder establecer una comunicación con el computador por medio de la interfaz RS232.

También el sistema esta provisto de un POWER ON RESET el mismo que se lo realizó por medio de un capacitor de 10 uF y una resistencia de 8.2K $\Omega$ . Lo mencionado se lo puede visualizar en el circuito de la figura 2.54, que se encuentra al final de este capítulo.

## 2.6.2 CIRCUITO PARA EL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE

Este circuito está basado en el microcontrolador ATMEL 89C2051 y se lo presenta en la figura 2.55. Consta de un circuito detector de cruce por cero cuya salida se la ingresa al pin P3.4 y el pin que activa el optoacoplador que dispara el TRIAC es el P3.3 circuitos que se explicaron en los subcapítulos 2.4.1.1 y 2.4.1.2. Esta tarjeta tiene capacidad para conectar hasta 5 sensores de temperatura DS1821 los cuales se conectan a los pines P1.7, P1.6, P1.5, P1.4 y P3.2. Además, se tiene 5 LEDs; el primero de ellos es el denominado ESTABLE conectado al pin P1.0, el cual es de color verde y se enciende cuando se han alcanzado los SET POINTS que el usuario ingreso. El segundo denominado ACTUADOR conectado al pin P1.1, cuyo color es rojo, conmuta su estado cada vez que se ha tomado una acción de control. El tercer LED denominado MUESTREO conectado al pin P1.2, de color amarillo, conmuta su estado cada vez que se toma una muestra de temperatura. El cuarto LED denominado FALLA DS1821 conectado al pin P3.5, de color rojo, se enciende si se detectó una falla de comunicación con cualquiera de los dispositivos DS1821. El último LED denominado FALLA de NIVEL conectado al pin P1.3, de color rojo, se enciende cuando el nivel dentro del reboiler es menor a 5.

Asimismo, existe un pin que funciona como entrada la cual esta conectada a la salida de uno de los pines del control de nivel, el cual esta en un nivel 0 lógico cuando el nivel dentro del reboiler es mayor o igual a 5 y se pone en un estado 1 lógico cuando el nivel dentro del reboiler es menor a 5.

Para el circuito de recepción se ha colocado dos NAND 4093 tipo SCHMITT TRIGGER para la recepción de datos. Y para la transmisión sólo se ha colocado una resistencia de PULL UP de  $10K\Omega$  a la salida del pin de transmisión.

### **2.6.3 CIRCUITO DE POTENCIA PARA EL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE**

El circuito de potencia mostrado en la figura 2.56, es el que se encarga de manejar la cantidad de corriente que circula a través de la resistencia de carga. Es un circuito sencillo; consta de un TRIAC de potencia y un divisor de tensión que se conecta a las salidas de un optoacoplador que es un OPTOTRIAC, las entradas del optoacoplador se encuentran conectadas, el ánodo del diodo emisor de luz a la fuente a través de una resistencia de limitación de corriente y el cátodo al pin P3.3 del microcontrolador.

### **2.6.4 CIRCUITO PARA EL CONTROL DEL NIVEL DENTRO DEL REBOILER**

El circuito de control de nivel dentro del reboiler está basado en un microcontrolador AT89C51. El conjunto de los 8 sensores magnéticos colocados dentro del reboiler, sirve para determinar uno de los 8 niveles dentro del mismo. Cada uno de estos sensores está conectado a uno de los pines del pórtico 0, y para una visualización de cada nivel se a conectado un display de 7 segmentos a los pines del pórtico 1, por medio del cual se conoce en que nivel del reboiler se encuentra el líquido.

Con los pines P2.0 hasta el pin P2.3 se controlan los diodos emisores de luz de 4 optoacopladores que a su vez actúan sobre otros transistores que manejan mayor corriente para controlar las bobinas del motor paso a paso, en la figura 2.53 estos pines están etiquetados como OPTO mas un índice. Por los pines P2.4 y P2.5 ingresan las señales de las salidas de los optoacopladores que sirven para determinar si la válvula de fondos está completamente cerrada, y cada uno de ellos tiene asociado un LED que se enciende cada vez que el optoacoplador entra en el estado de saturación. El LED etiquetado como CERRADO, se enciende cuando el optoacoplador etiquetado como CERRADO entra en el estado de saturación; el LED etiquetado como CERRADO indica que la válvula está cerrada ó se enciende cuando el motor paso a paso ha dado una vuelta completa. El optoacoplador y LED etiquetados como CHECK se activan 14 pasos después de

que la ranura del disco acoplado al eje del motor paso a paso, permite el paso de la luz del primer optoacoplador etiquetado como CERRADO en el sentido de cierre de la válvula. Además existe un LED etiquetado como ERROR en la figura 2.53, conectado al pin P3.2 que se enciende cuando el cero de los contadores dentro del programa para controlar el motor paso a paso, no coincide con el estado de saturación del transistor del optoacoplador etiquetado como CERRADO.

La activación de la señal de ENABLE que es la que se envía al control de temperatura del dispositivo de calentamiento para hacerle saber a dicho control si el nivel esta bajo 5, se la realiza a través de un optoacoplador que esta conectado al pin P3.3. El circuito descrito corresponde al que se presenta en la figura 2.57 al final del capítulo.

#### **2.6.5 CIRCUITO DE POTENCIA DEL CONTROL DE NIVEL DENTRO DEL REBOILER**

En el circuito de potencia del control del nivel del líquido dentro del reboiler es manejado por medio de cuatro optoacopladores, los mismos que activan a otros cuatro transistores cuya finalidad es proveer mayor corriente, estos últimos a su vez son los encargados de manejar las bobinas del motor de pasos. Además se a colocado un diodo en paralelo a cada bobina en polarización inversa, para que se pueda descargar la energía de la bobina a través del diodo cuando ésta ya no esté energizada y de esa forma evitar la corriente inversa a través de la juntura del transistor. Este circuito de potencia es presentado en la figura 2.58.

#### **2.6.6 CIRCUITO DE CONTROL PARA EL ACUMULADOR, VENTILADOR Y TANQUES DE ALIMENTACIÓN Y DE RESERVA**

Este circuito está basado en un microcontrolador AT89C2051. Hay tres niveles que se establecen en el acumulador a través de 3 switches magnéticos que están conectados a los pines P1.7, P1.6 y P1.5. Se dispone de 4 optoacopladores encargados de controlar el encendido y apagado de los siguientes dispositivos: la bomba de reflujo, la bomba de cosecha, la bomba de alimentación y el ventilador.

Y estos pines son: P3.2, P3.3, P3.4 y P3.5. Además la activación de cada optoacoplador esta asociada con el encendido de un LED, los mismos que están conectados a los pines P1.2, P1.1, P1.0 y P3.7. Se tienen también dos sensores de nivel uno de los cuales esta instalado en el tanque de alimentación y el otro en el tanque de reserva, cada uno de ellos tiene asociado un circuito de retardo de 2 segundos y están conectados a los pines P1.4 y P1.3.

Para la recepción de la comunicación serial se ha colocado dos compuertas NAND 4093 tipo SCHMITT TRIGGER conectadas en serie para de esa forma mejorar los flancos de la señal receptada. También este circuito tiene un circuito de POWER ON RESET como se puede visualizaren la figura 2.59.

### **2.6.7 CIRCUITO DE POTENCIA PARA EL CONTROL DEL ACUMULADOR, VENTILADOR Y TANQUES DE RESERVA Y ALIMENTACIÓN**

En este circuito se encuentran los optoacopladores que activan a transistores ECG 3906 los cuales a su vez activan relays de 5V que son los encargados de activar las bombas. Las bombas de alimentación y de reflujo funcionan a 120VAC y la bomba de cosecha funciona a 12 VDC, conjuntamente con la válvula de cosecha se activa una electroválvula que funciona con 120VAC. A cada bobina de cada RELAY se ha conectado un diodo en paralelo en polarización inversa para que la energía almacenada en la bobina se descargue a través del diodo cuando esta ya no esté energizada. Este circuito de potencia se lo puede visualizar en la figura 2.60.

## **2.7 POSIBILIDADES DE CONTROL**

Ahora se pasará a describir las distintas posibilidades de control sugeridas en los libros para una torre de destilación de alcohol, y se analizará las ventajas o desventajas de cada una de ellas. A continuación se pasará a describir la posibilidad de control de la figura 2.50.

En el esquema de control de la figura 2.50 se lo describe de la siguiente manera:

- El nivel dentro del acumulador se controla con la válvula de productos.
- El nivel dentro del reboiler se controla con la válvula de fondos.
- El caudal de reflujo se lo controla por medio de la medición de temperatura en alguna parte de la torre
- La corriente que circula a través del dispositivo de calentamiento es constante.
- La corriente que circula a través del ventilador es constante

Al mantener la corriente constante en el dispositivo de calentamiento no se tiene un buen control de la composición de los fondos; además si la concentración de la mezcla dentro del reboiler es alta en alcohol, teniendo una alta corriente circulando a través de la del dispositivo de calentamiento producirá que exista un alto burbujeo impidiendo realizar el control de nivel dentro del reboiler. Y si la corriente que circula a través de la resistencia es muy baja evitará que los gases de ebullición de la mezcla alcancen el tope de la torre.

El control del reflujo por medio de la medición de temperatura en la parte alta de la torre permitiría tener una buena concentración de fondos. Pero un excesivo caudal de reflujo provocaría que no se pueda tener la concentración de fondos adecuada como para abrir la válvula de fondos para hacer el control de nivel.

Por otro lado se tiene un buen control de nivel del acumulador por medio de la válvula de productos. Pero al haber un flujo variable de la válvula de reflujo tendría que haber un suficiente flujo de entrada al acumulador para que pueda funcionar la válvula de productos.

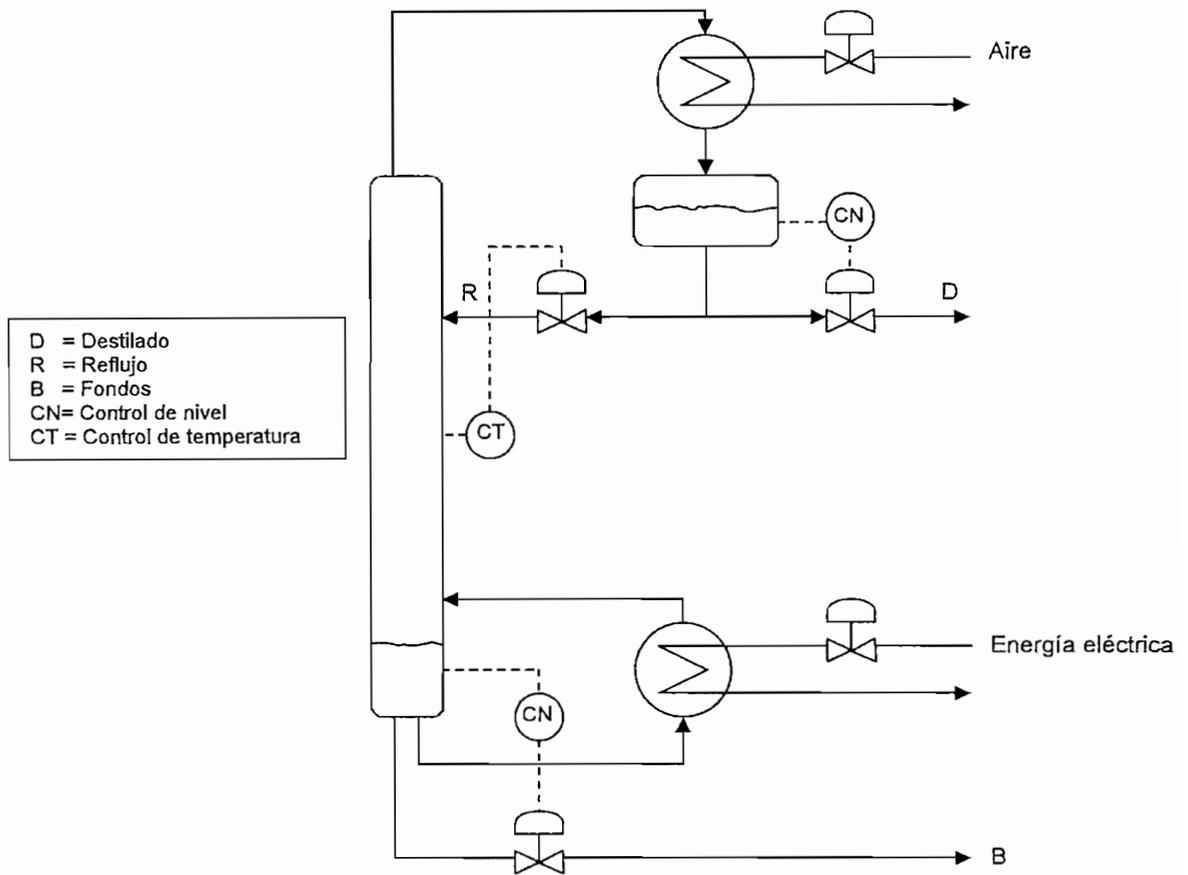


Figura 2.50

El esquema de control de la figura 2.51 se lo puede describir de la siguiente manera:

- El nivel dentro del acumulador se controla con la válvula de reflujo.
- El nivel dentro del reboiler se controla con la válvula de fondos.
- La cosecha se controla por medio de la medición de temperatura en alguna parte de la torre
- La corriente que circula a través del dispositivo de calentamiento es constante.
- La corriente que circula a través del ventilador es constante

El esquema de control de la figura 2.51 tiene la misma debilidad que el de la figura 2.50 y esto es que la corriente que circula por el dispositivo de calentamiento es constante y eso no permite un buen control sobre la composición de fondos.

Además se podría producir un burbujeo que impediría realizar el control de nivel dentro del reboiler.

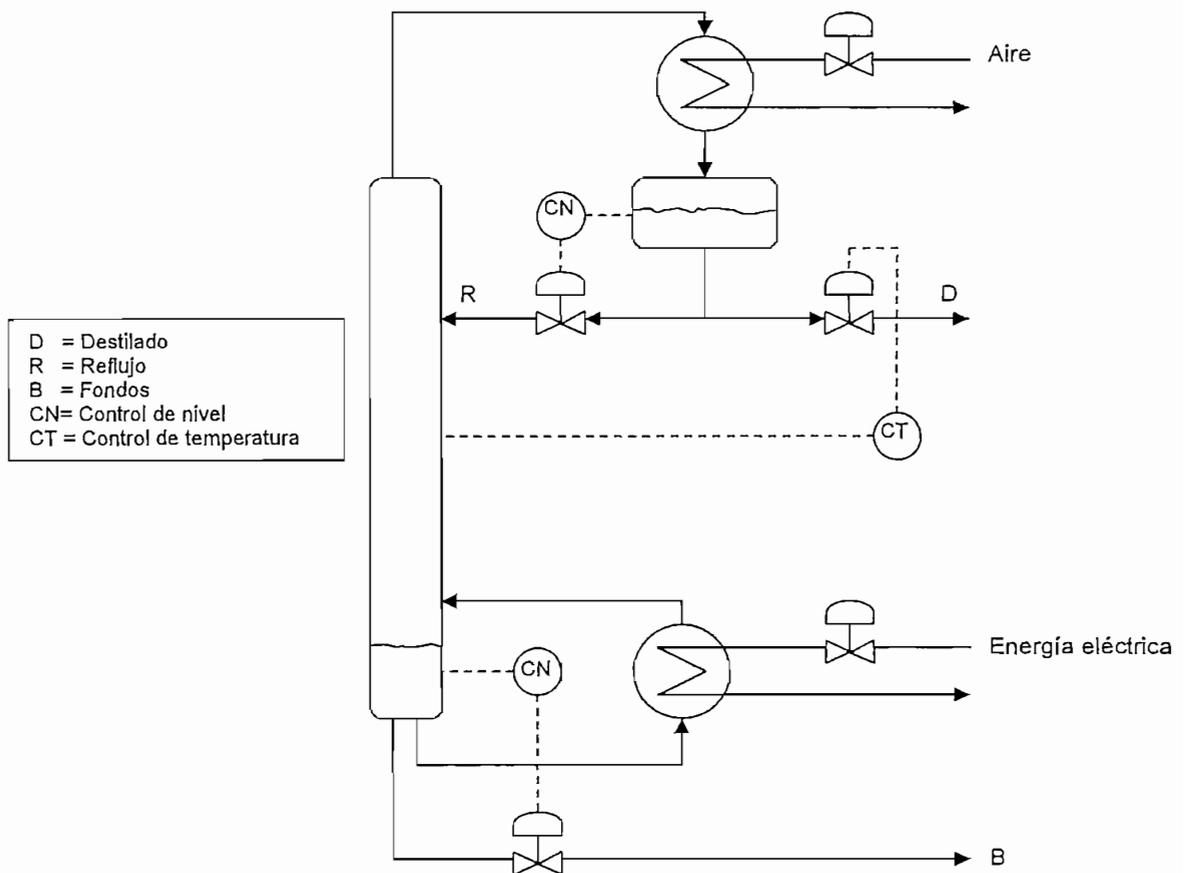


Figura 2.51

El esquema de control de la figura 2.52 se lo describe como sigue:

- El nivel dentro del acumulador se controla con la válvula de productos.
- El nivel dentro del reboiler se controla con la cantidad de corriente que se envía al dispositivo de calentamiento.
- El caudal de reflujo es constante.
- La válvula de fondos se controla a través de la medida de temperatura en alguna parte de la torre.
- La corriente que circula a través del ventilador es constante.

Al tener controlada la válvula de fondos con alguna medición de temperatura en la torre, se aseguraría tener una buena concentración de fondos. Por otro lado el controlar el nivel dentro del reboiler por medio de la corriente que circula a través del dispositivo de calentamiento, podría ser que haga burbujear la mezcla dentro del reboiler.

También lo positivo de este esquema de control es que mantiene constante la realimentación y eso da cierta estabilidad a la torre de destilación.

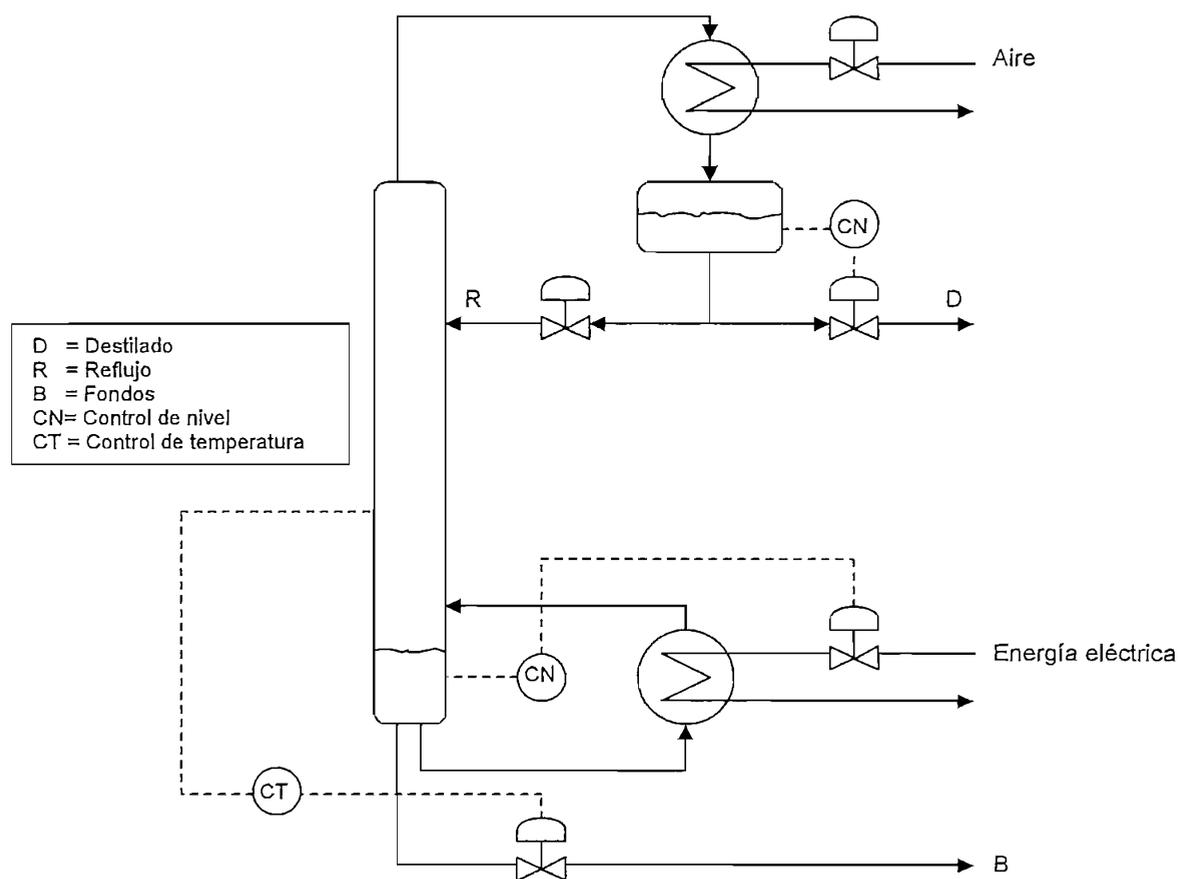


Figura 2.52

El esquema de control de la figura 2.53 se describe como sigue:

- El nivel dentro del acumulador se controla con la válvula de productos.
- El nivel dentro del reboiler se controla con la válvula de fondos.
- El caudal de reflujo es constante.

- La corriente que circula a través del dispositivo de calentamiento se controla a través de la medida de temperatura en alguna parte de la torre.
- La corriente que circula a través del ventilador es constante.

El esquema de la figura 2.53 se utilizó como base de este proyecto. El tener el reflujo constante da estabilidad a la torre y el control de la corriente que circula a través del dispositivo de calentamiento se lo realizó a través de dos mediciones de temperatura en dos partes de la torre que son: la cima y dentro del reboiler. Con ello se garantiza tener las composiciones que se requieren tanto en productos como en fondos. Además con la válvula de productos se puede obtener un muy buen control del nivel dentro del acumulador.

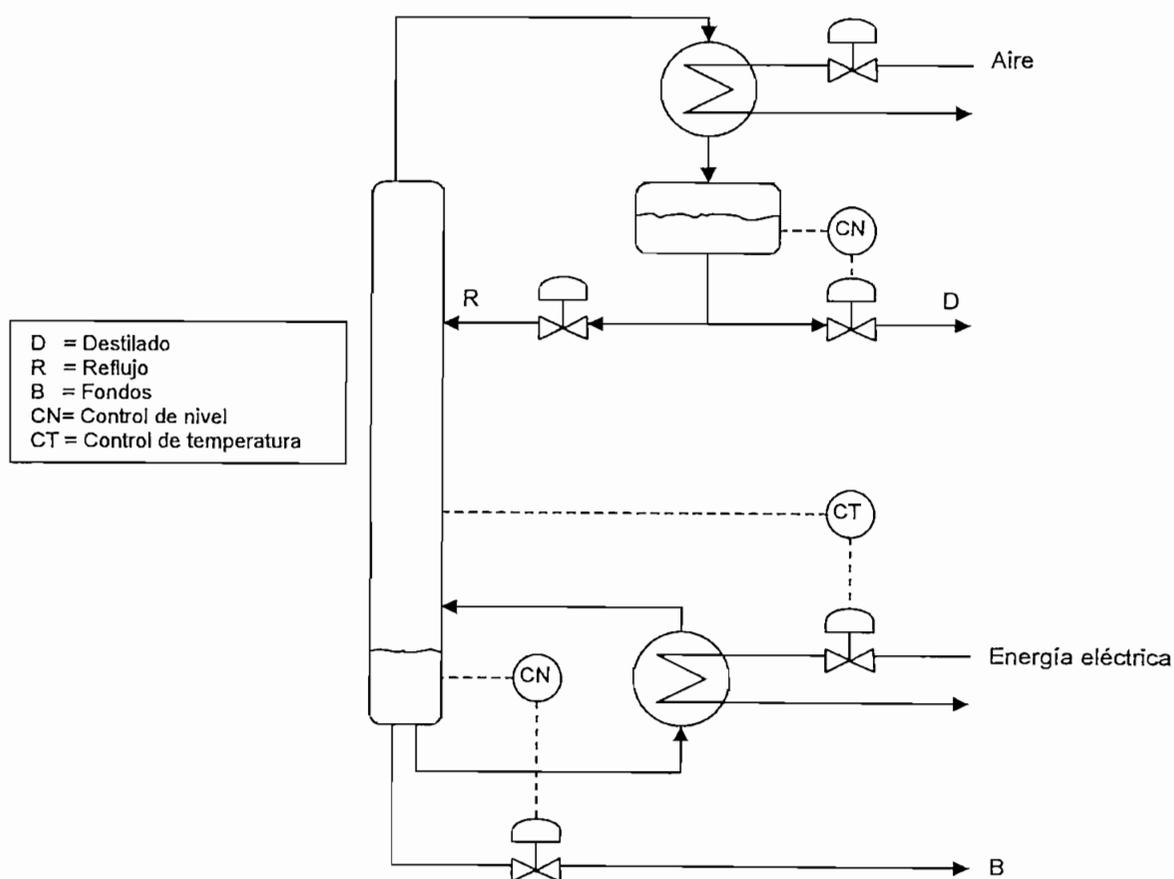


Figura 2.53

A continuación se pasará a describir la funcionalidad de los circuitos de control.

### 2.7.1 OPCIONES DEL PANEL PRINCIPAL

La función del panel principal es la de que el usuario pueda ingresar los SET POINTs, encender o apagar cada uno de los dispositivos en el modo MANUAL y visualizar la temperatura de tres puntos diferentes de la torre. Esto se hace por medio de la visualización de menús en el LCD. El primer menú es el del control automático en donde el usuario puede ingresar los SET POINTs y ver cualquiera de las tres temperaturas. El menú del control automático se lo puede ver en la tabla 2.57.

MENU EN EL MODO AUTOMATICO		
OPCION	ETIQUETA	DESCRIPCIÓN
01	P inicio Reb1	Con esta opción el usuario ingresa el porcentaje de potencia con el que funcionará la resistencia al momento que detecte ebullición o cuando se llegue a la temperatura de disparo que se lee con el DS1821 ubicado en la parte más alta de la torre.
02	SP T Fondos	Esta opción sirve para que el usuario ingrese el SET POINT de temperatura a la cual vamos a obtener cierto grado de concentración en los productos. La lectura de temperatura se la obtiene del DS1821 que se encuentra dentro del reboiler.
03	SP T Producto	Esta opción sirve para que el usuario ingrese el SET POINT de temperatura a la cual vamos a obtener cierto grado de concentración en los fondos. La lectura de temperatura se la obtiene del DS1821 ubicado en la parte más alta de la torre.
04	Tempe Disparo	Con esta opción se ingresa la temperatura de disparo. Es decir cuando se alcanza esta temperatura en la parte más alta de la torre,

		automáticamente se hace que el dispositivo de calentamiento funcione al porcentaje de potencia ingresado en la opción 1.
05	Ver T Reboil1	Esta opción sirve para mostrar en el LCD la temperatura que existe dentro del reboiler.
06	Ver T Dallas2	Esta opción sirve para mostrar en el LCD la temperatura que existe un poco más arriba de la mitad de la torre.
07	Ver T Dallas3	Esta opción sirve para mostrar en el LCD la temperatura que existe en la parte alta de la torre.
08	Delay Acumulado	Aquí se ingresa el tiempo que el usuario desea que permanezca prendida la bomba de reflujo una vez que se halla detectado el nivel 2 en el acumulador. Al valor ingresado se le multiplica por 50ms y ese es el tiempo que la bomba de cosecha permanecerá encendida.
09	Min_Act_Motor	Al valor ingresado aquí se le multiplica por 10 y ese será el número de grados mínimo que se abrirá la válvula de fondos.
10	Max_Act_Motor	Al valor ingresado aquí se le multiplica por 10 y ese será el número de grados máximo que se abrirá la válvula de fondos.

Tabla 2.7

Solo se pueden visualizar dos opciones a la vez en el LCD y para poder desplazarse a través del menú se utilizan las teclas "\*" y "#". Para poder pasar del menú automático al menú manual se necesita presionar la tecla "D" seguida de 4 dígitos. Y para pasar del menú control manual al menú control automático solo basta con presionar la tecla "D" una vez.

Las teclas "A", "B" y "C" son teclas de función: la letra "A" sirve para aceptar un dato, la letra "B" sirve para borrar un dato ingresado, y la letra "C" sirve para cancelar una entrada o retornar al menú principal en ciertos casos.

A continuación se presenta las opciones que ofrece el menú manual en la tabla 2.8.

<b>MENU EN EL MODO MANUAL</b>		
<b>OPCION</b>	<b>ETIQUETA</b>	<b>DESCRIPCIÓN</b>
01	%P Reboiler	Con esta opción el usuario ingresa el porcentaje de potencia con el que funcionará la resistencia al momento que detecte ebullición o cuando se llegue a la temperatura de disparo que se lee con el DS1821 ubicado en la parte más alta de la torre.
02	Ver T Reboil1	Esta opción sirve para mostrar en el LCD la temperatura que existe dentro del reboiler.
03	Ver T Dallas2	Esta opción sirve para mostrar en el LCD la temperatura que existe un poco más arriba de la mitad de la torre.
04	Ver T Dallas3	Esta opción sirve para mostrar en el LCD la temperatura que existe en la parte alta de la torre.
05	I/O Reflujo	Esta opción sirve para encender o apagar el reflujo.
06	I/O Cosecha	Esta opción sirve para encender o apagar la cosecha.
07	I/O Alimentar	Esta opción sirve para encender o apagar la alimentación.
08	I/O Ventilar	Esta opción sirve para encender o apagar el ventilador.
09	Abrir 10 grad	Esta opción sirve para abrir la válvula de fondos 10 grados.

10	Cerra 10 grad	Esta opción sirve para cerrar la válvula de fondos 10 grados
11	Abrir válvula	Esta opción sirve para abrir por completo la válvula de fondos.
12	Detener	Esta opción sirve para detener el motor, cuando se están ejecutando cualquiera de los comandos "Abrir válvula" o "Cerrar válvula".
13	Cerrar válvula	Esta opción sirve para cerrar por completo la válvula de fondos.

Tabla 2.8

Además al control del panel llegan todas las mediciones de temperatura, y es el encargado de visualizar una de ellas a la vez, dependiendo de la selección del usuario. También se ejecuta un algoritmo de control que controla la apertura de válvula de fondos y el encendido del ventilador.

### **2.7.2 OPCIONES DEL CONTROL DE TEMPERATURA DEL DISPOSITIVO DE CALENTAMIENTO DE LA BASE**

Al control de temperatura del dispositivo de calentamiento de la base llegan 4 valores importantes que son: La potencia inicial, el SET POINT de temperatura de fondos, SET POINT de temperatura de productos y el SET POINT de temperatura de disparo.

El valor de potencia inicial es un valor de potencia entre 0 y 100 que el usuario ingresa para que en el momento que se detecte ebullición (variación brusca del nivel menor a 5) o si se llega al SET POINT de temperatura de disparo que se mide en el tope de la torre; inmediatamente carga y ese será el valor de potencia con el que funcionará la resistencia hasta que el valor de temperatura en el tope de la torre halla disminuido o las variaciones bruscas hallan terminado.

De ahí en adelante lo que primero busca el control es llegar al SET POINT de temperatura de productos. Una vez alcanzado el SET POINT de temperatura de

productos, va incrementado la potencia de la resistencia en forma secuencial desde la potencia de inicio hasta alcanzar el valor de SET POINT de fondos, una vez hecho esto la potencia permanece fija hasta que halla alguna perturbación que le obligue a subir o bajar la potencia secuencialmente.

### **2.7.3 OPCIONES DEL CONTROL DE NIVEL DEL REBOILER**

Lo primero que realiza este control es sensor el nivel que hay dentro del reboiler para de esa manera controlar el apagado del dispositivo de calentamiento de la base. Cuando el nivel es menor a 5 el control automáticamente envía una señal al control de temperatura del reboiler para que apague la resistencia.

De ahí en adelante el objetivo del control de nivel es mantener el nivel de líquido dentro del reboiler en el nivel 6. Cuando sobrepasa este nivel y se ha alcanzado el SET POINT de temperatura de fondos se abre la válvula de fondos hasta el valor que el usuario ingreso en la opción 10 del menú de control automático, para que de esa forma el nivel retorne otra vez a 6.

Al momento que se llega al SET POINT de temperatura de fondos o mayor, el control recibe una señal del panel principal para abrir la válvula el número de grados que el usuario ingreso en la opción 9 del menú de control automático, de igual forma si la temperatura dentro del reboiler es menor que el SET POINT de temperatura de fondos, el control del panel principal envía una señal para que se cierre por completo la válvula de fondos.

### **2.7.4 OPCIONES DE CONTROL DEL ACUMULADOR, VENTILADOR, BOMBAS DE REFLUJO Y ALIMENTACION**

Este algoritmo es el encargado de mantener a nivel el tanque de alimentación; así pues, cuando el nivel ha bajado en el tanque de alimentación, se activa la bomba que esta dentro del tanque de reserva y pone otra vez a nivel el tanque de alimentación. Existe otro sensor de nivel dentro del tanque de alimentación el que

se encarga de sensar cuando no existe líquido en el tanque de reserva; por lo tanto impide la activación de la bomba dentro del tanque de reserva.

También se encarga del control de las bombas de cosecha y de reflujo midiendo tres niveles diferentes en el acumulador. Cuando esta en el nivel más bajo se apagan las bombas de cosecha y de reflujo. Cuando se llega al nivel dos se enciende la bomba de cosecha por el tiempo establecido en la opción 8 del menú de control automático, para de esa forma mantener el nivel 2 dentro del acumulador. Cada vez que se detecte el nivel 2 la bomba de cosecha extraerá el suficiente nivel de líquido como para dejar el acumulador en el nivel 2. Asimismo al llegar al nivel 2 se enciende la válvula de reflujo que permanecerá encendida a menos que el nivel del líquido llegue al nivel 1.

En caso de que se llegue al nivel 3 dentro de acumulador se encienden la válvula de cosecha hasta volver nuevamente al nivel 2.

Además recibe una señal del panel principal para cuando haya una temperatura mayor o igual a 50 grados en la mitad de la torre, encienda el ventilador. Caso contrario apaga el ventilador.

## **2.8 PERTURBACIONES**

Existen 2 perturbaciones fundamentales en la operación de una columna de destilación binaria: el caudal de alimentación y la composición.

Si el caudal de alimentación es muy grande provocará que cambie la concentración de fondos para que se enriquezca en alcohol; por lo tanto el control de temperatura incrementará la potencia de la resistencia para que se evapore más rápido el alcohol. Por otro lado si el caudal de alimentación es pequeño se alcanzará el SET POINT de temperatura de fondos mucho más rápido que si la alimentación fuera grande.

Otra de las cosas que alteran el funcionamiento de la torre es la composición de la materia prima. No es lo mismo trabajar con una mezcla de alcohol al 30% que una mezcla de alcohol al 50%. Si se trabaja con una mezcla de alcohol al 50%

ebullicionará mucho más rápido que una mezcla de alcohol al 30% y en la mayoría de veces se produce burbujeo al inicio con una mezcla al 50%.

### Control principal de la torre de destilación de alcohol

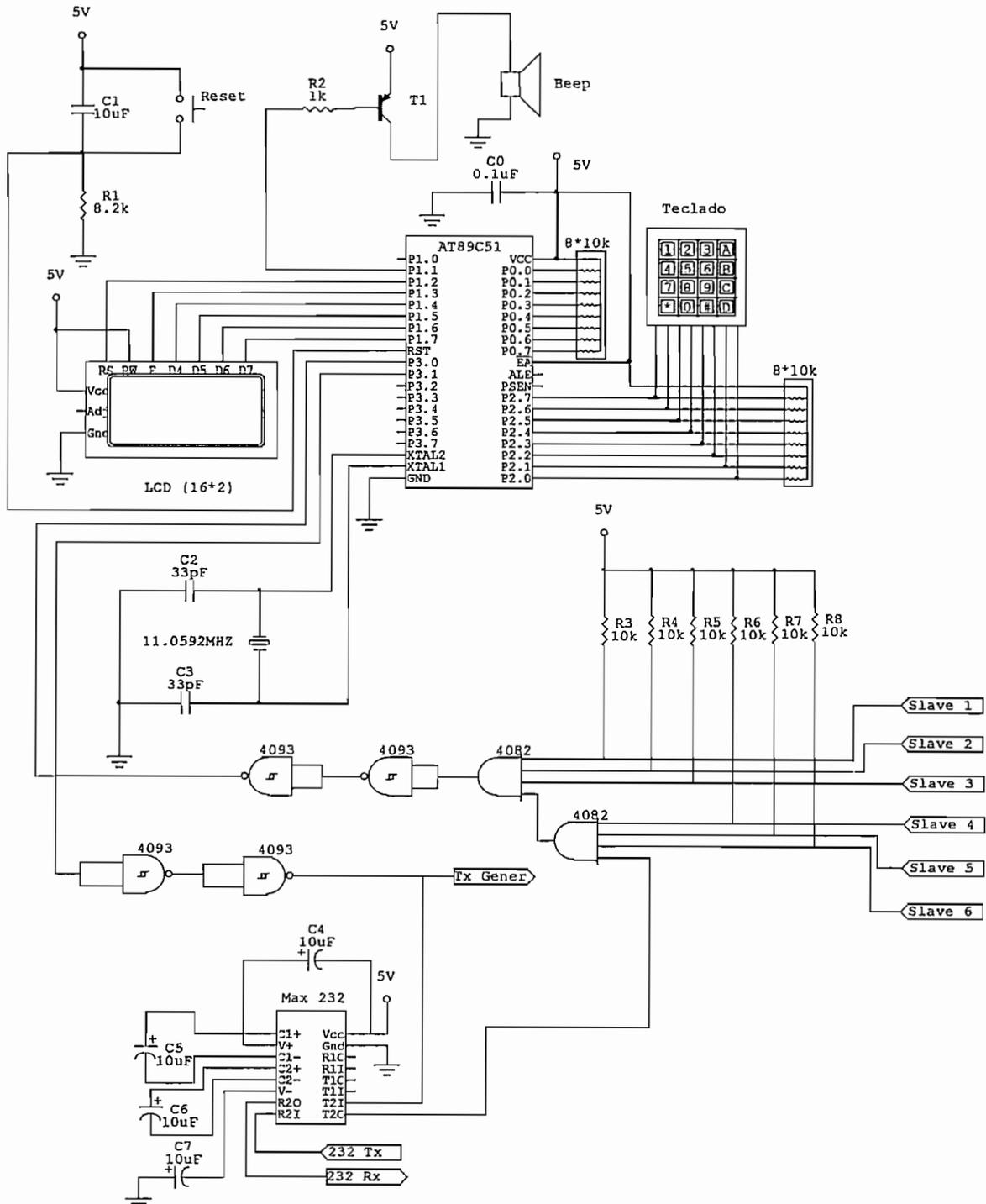


Figura 2.54



Circuito de potencia del control  
de temperatura del reboiler

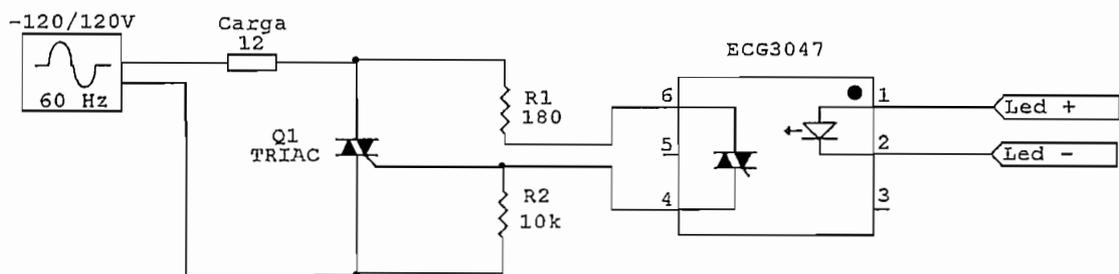


Figura 2.56

## Circuito para el control de nivel del reboiler

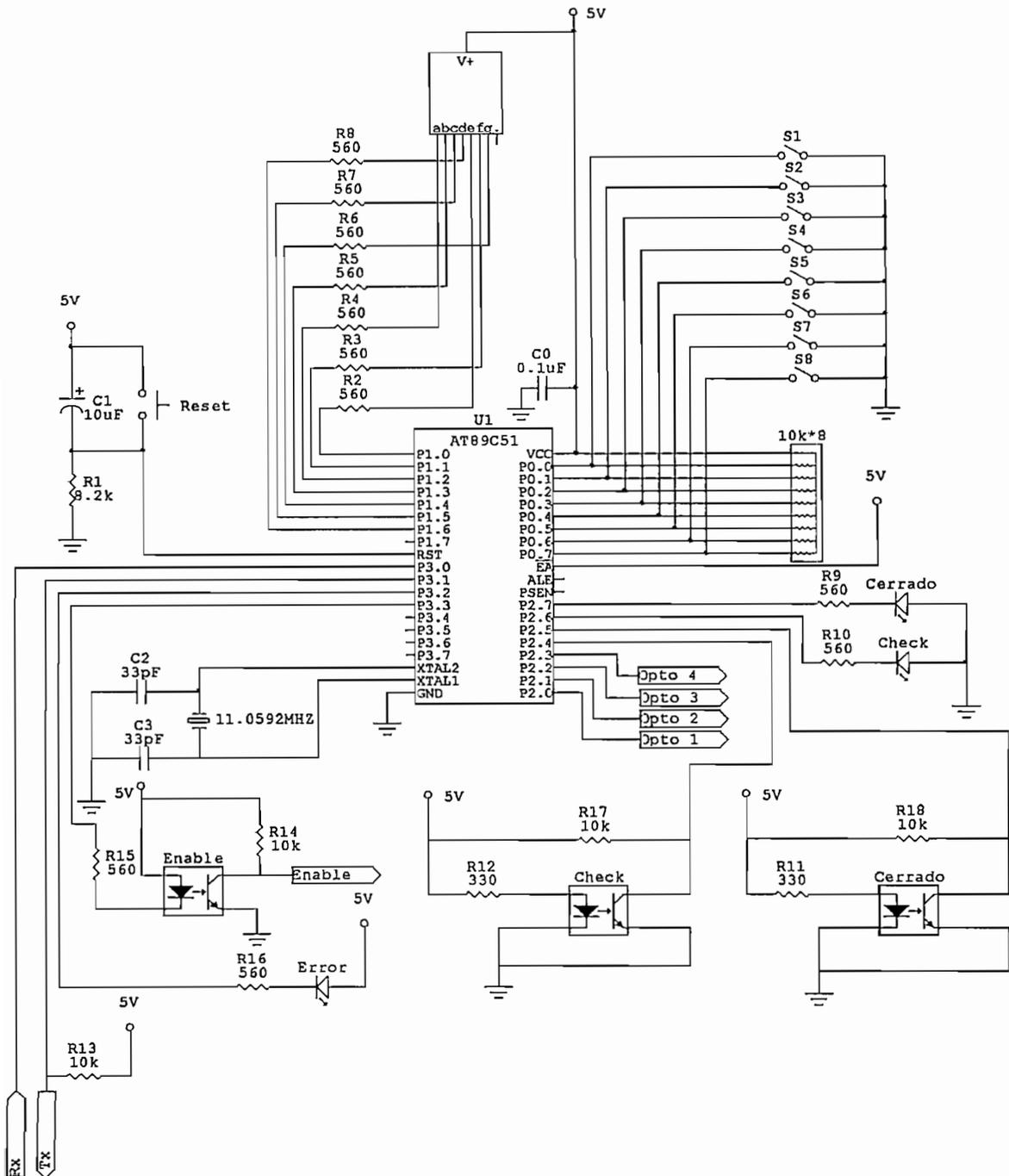


Figura 2.57

## Circuito de potencia del control de nivel del reboiler

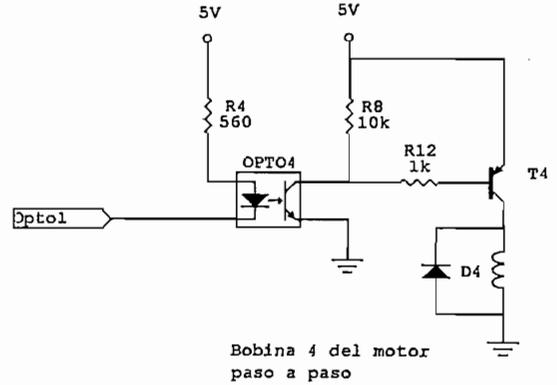
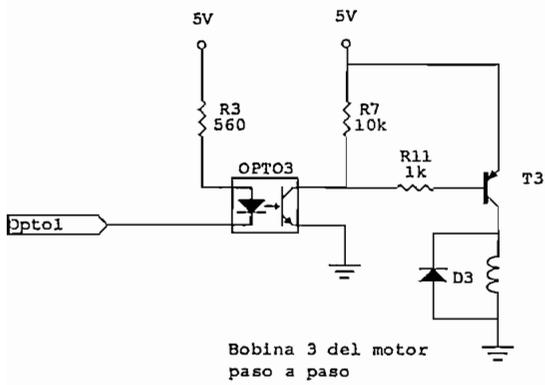
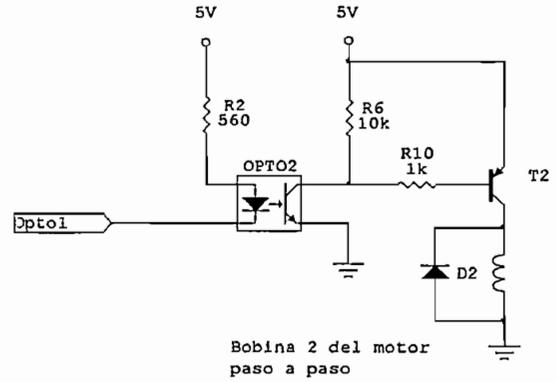
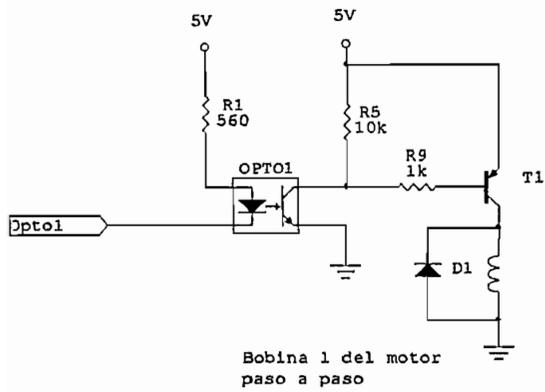


Figura 2.58

Circuito de control para el acumulador, ventilador y tanques de alimentación y de reserva

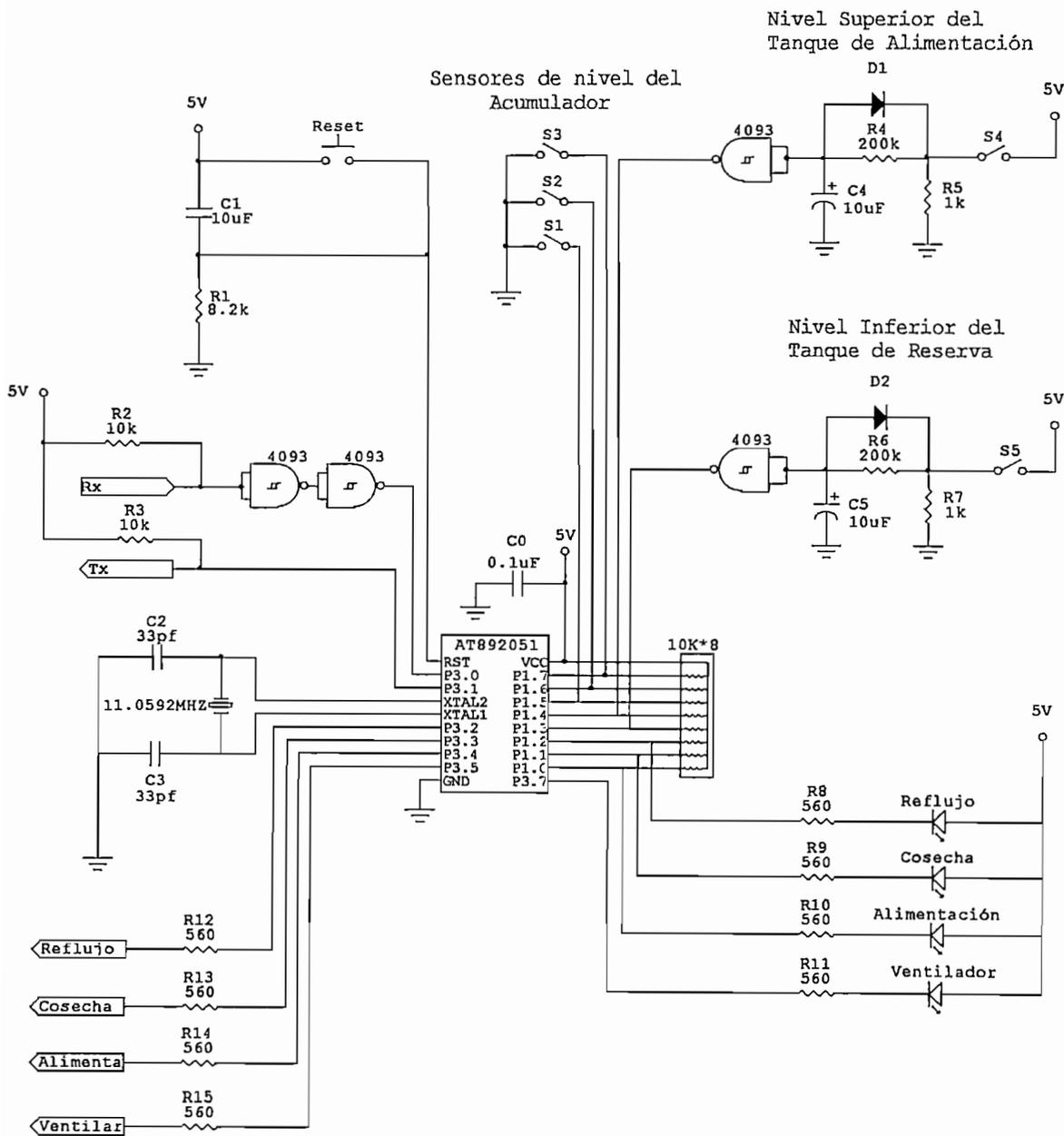


Figura 2.59

### Circuito de potencia del control del acumulador, ventilador y tanques de alimentación y de reserva

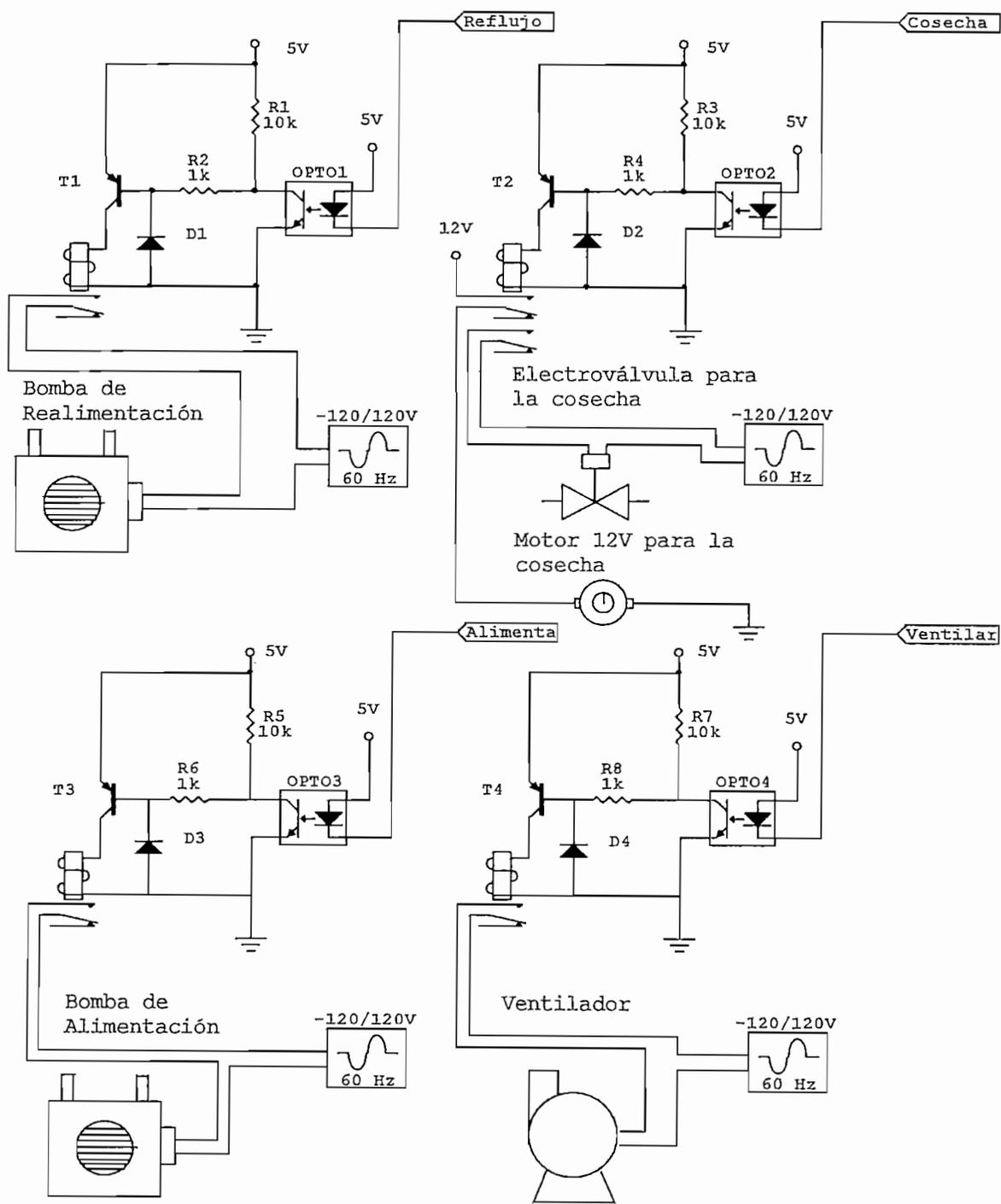


Figura 2.60

## **CAPÍTULO 3: IMPLEMENTACIÓN DEL SOFTWARE**

El presente capítulo tiene por objeto explicar y representar en diagramas de flujo, los programas y subrutinas usados por cada uno de los microcontroladores utilizados en cada uno de los circuitos de control y en el panel principal. Cabe decir, que los diagramas de flujo deben ser analizados en conjunto con el listado de etiquetas del programa que se incluyen en los anexos y las etiquetas que se han utilizado en los mismos.

### **3.1 SUBRUTINAS DEL PANEL PRINCIPAL**

El programa en lenguaje assembler que se ejecuta en el microcontrolador AT89C51 para manejar el panel principal se encuentra en el anexo 1.

#### **3.1.1 PROGRAMA PRINCIPAL**

El programa principal en su inicio se encarga de borrar la memoria RAM así como también de configurar los parámetros de la comunicación serial a 19200 bps y 9 bits de datos, habilitar la interrupción serial, inicializar algunas variables y configurar el LCD.

Una vez realizada la inicialización del programa, entra en un bucle en donde constantemente se ejecuta un algoritmo de control que actúa de acuerdo a las medidas de temperatura que le envía el control de temperatura del reboiler y además se lee las entradas del teclado, para de esa manera escoger alguna de las opciones que se ofrece en el menú que se presenta en el LCD. También aproximadamente cada 4.3 segundos se tiene una interrupción debida al pórtilo serial debido a que el control de temperatura del reboiler envía los datos de temperatura y potencia de la resistencia al panel principal.

El panel principal maneja también las comunicaciones hacia los dispositivos esclavos. Dependiendo de la opción que seleccionó el usuario, primero se envía la dirección al bus y después el dato o los datos. Existen 3 direcciones a las que se

les ha asignado etiquetas y que se refieren a los 3 dispositivos esclavos a los cuales se envía la información desde el panel principal; las cuales se detallan en la tabla 3.1.

Etiqueta	Número HEX	Comentario
DIREC_1	11H	Hace referencia a la dirección del control de temperatura del reboiler.
DIREC_2	2DH	Hace referencia a la dirección del control del acumulador, ventilador y bombas.
DIREC_3	33H	Hace referencia a la dirección del control de nivel del reboiler.

Tabla 3.1

Los diagramas de flujo que explican el programa principal del cual se desprenden las demás subrutinas excepto las subrutinas de interrupción, se presentan en las figuras 3.1.a, 3.1.b, 3.1.c, 3.1.d, 3.1e, 3.1.f.

### 3.1.2 SUBROUTINA INI\_4BITS

Esta es la subrutina encargada de inicializar el LCD en el modo de bus de datos de 4 bits. El programa fue desarrollado en base al diagrama de flujo presentado en la figura 2.44, en el cual se envía 3 veces un mismo dato a diferentes tiempos de retardo. Debido a ello se optó por enviar el mismo dato 3 veces por medio de un bucle con el máximo tiempo de retardo de todos ellos. Y después se envía el último dato que es diferente de los anteriores con el mismo tiempo de retardo. El diagrama de flujo de esta subrutina es mostrado en la figura 3.2

### 3.1.3 SUBROUTINA COMANDO

Esta subrutina sirve para enviar comandos al LCD como por ejemplo: apagar o encender la pantalla, que el cursor titile o no, que el cursor baje una línea o suba una línea etc. El manejo de las líneas que controlan el bus del LCD que son la

línea E y la línea RS y R/W se basa en el diagrama de estados mostrado en la figura 2.41. Como no se van a leer datos del LCD la línea de control del bus del LCD R/W permanece permanentemente en el estado 0L.

Para enviar un comando al LCD lo primero que se hace es cargar el acumulador con este valor y después se llama a la subrutina COMANDO. Hay que tomar en cuenta el tiempo que le toma al LCD ejecutar un comando, porque puede ser que si no se espera el tiempo necesario antes de enviar el siguiente comando, el primer comando enviado, no se ejecute.

Existen dos tiempos de ejecución de los comandos según la tabla 2.4: 40us y 1.64ms. Como sólo existen 2 comandos que necesitan de un tiempo de ejecución de 1.64ms, se creo una subrutina para cada uno de ellos. El diagrama de flujo de la subrutina COMANDO se muestra en la figura 3.3.

Para no estar recurriendo a la tabla 2.4 para determinar la combinación de bits para ejecutar determinado comando, se optó por relacionar etiquetas con las combinaciones más utilizadas para de esa forma relacionar el nombre de la etiqueta con el comando que se desea realizar. La tabla 3.2 presenta las etiquetas que representan comandos del LCD.

Etiqueta	Número HEX	Comentario
SET4_BITS	28H	Con este comando se configura la matriz del carater a 5 * 7 puntos, también que el LCD utilice 2 líneas y que el bus de datos sea de 4 bits
DISPLAY_OFF	08H	Con este comando se apaga la pantalla del LCD.
DISPLAY_ON	0CH	Con este comando se enciende la pantalla del LCD.
CURSOR_ON	0EH	Con este comando se enciende el cursor.
BLINK_ON	0FH	Con este comando el cursor empieza a titilar.

CURSOR_INC_TF	06H	Con este comando el texto permanece fijo, y se incrementa la posición del cursor en una posición a la derecha después de la escritura de un carácter.
CURSOR_LEFT	10H	Con este comando el cursor se desplaza una posición hacia la izquierda.
CURSOR_RIGHT	14H	Con este comando el cursor se desplaza una posición hacia la derecha.
LOWER_LINE	C0H	Con este comando el cursor se dirige a la línea inferior.
UPPER_LINE	80H	Con este comando el cursor se dirige a la fila superior.

Tabla 3.2

### 3.1.4 SUBROUTINA CLS

La subrutina CLS permite borrar todos los caracteres del LCD y envía el cursor al inicio de la primera fila, el tiempo de ejecución de este comando según la tabla 2.4 es de 1.64ms. El diagrama de estados de las líneas que manejan el bus; es el mismo que para todos los comandos, mostrado en la figura 2.41; la diferencia radica en que este comando tiene un tiempo de ejecución mayor. El diagrama de flujo de esta subrutina es mostrado en figura 3.14.

### 3.1.5 SUBROUTINA HOME

Esta subrutina sirve para enviar al cursor del LCD al inicio de la primera fila, el tiempo de ejecución de este comando según la tabla 2.4 es de 1.64ms. El diagrama de estados para este comando es el mismo de la figura 2.41. El diagrama de flujo de esta subrutina es mostrado en la figura 3.15.

### 3.1.6 SUBROUTINA CHARACTER

La subrutina CHARACTER sirve para enviar un dato que será visualizado en el LCD. El LCD trabaja con caracteres ASCII; así por ejemplo si se desea visualizar la letra "A" bastará con colocar el código ASCII de la letra "A" en el acumulador y después llamar a la subrutina CHARACTER. El tiempo de ejecución de esta instrucción según la tabla 2.4 es de 120us y el diagrama de estados de las señales que controlan el bus es mostrado en la figura 2.42. Cabe decir que esta subrutina sólo puede visualizar un caracter a la vez y el diagrama de flujo de la misma es presentado en la figura 3.6.

### 3.1.7 SUBROUTINA LCD

Ante la necesidad de enviar al LCD no solamente un caracter si no un conjunto de caracteres, se diseño subrutina LCD la cual envía todos los caracteres de una tabla hacia el LCD. Entonces para utilizar esta subrutina lo primero que hay que hacer es cargar el valor de la base de la tabla en el DPTR, el número de caracteres que van ha ser visualizados en una fila del LCD, se carga en el registro R3 y se procede a llamar a la subrutina LCD. El diagrama de flujo se muestra en la figura 3.7. Debe tenerse en cuenta que el número de caracteres máximo que se puede visualizar en un LCD, depende directamente del número de caracteres por fila del LCD.

### 3.1.8 SUBROUTINA TECLADO

La subrutina TECLADO lo que hace es rotar un cero en los 4 pines más significativos del pòrtico 2, y en cada rotación se lee el pòrtico 2. Si el valor que se lee es diferente al que se envió significa que se presionó una tecla, caso contrario se rota una vez más el cero hasta que se haya barrido las cuatro columnas del teclado. Si después de haber barrido todas las columnas no se ha detectado la presión de una tecla se carga el acumulador con el número 80H. Por otro lado después que se ha detectado la presión de una tecla, se ejecuta una subrutina de retardo de 100ms para eliminar los rebotes del contacto. Cuando se comprueba la

presión de una tecla se procede a llamar a la subrutina BEEPER la cual produce una señal sonora con cada presión de una tecla. El diagrama de flujo de esta subrutina es mostrado en la figura 3.8. Cabe decir que el código de la tecla presionada se guarda en el acumulador.

### 3.1.9 SUBROUTINA BEEPER

Esta subrutina es la encargada de activar un transistor, el mismo que activa un anunciador sonoro con cada presión de alguna tecla del teclado. El pulso de activación que se envía al transistor tiene una duración de 35ms. El diagrama de flujo que explica esta subrutina es el de la figura 3.9.

### 3.1.10 SUBROUTINA COD2HEX

La subrutina COD2HEX sirve para transformar el código de la tecla presionada; es decir, el dato que se lee cuando se presiona una tecla y que es guardado en el acumulador, como un número comprendido entre 00H y 0FH. Esto se lo hace mediante el uso de la tabla T\_CODIGO, que es mostrada en la tabla 4.2. Entonces la subrutina va comparando con todos los valores de la tabla, y donde coincida el número, se guarda el valor del DPL en el acumulador, siendo este valor el valor hexadecimal equivalente de la tecla presionada. El diagrama de flujo que explica el funcionamiento de esta subrutina es el mostrado en la figura 3.10. En el caso de que el dato guardado en el acumulador no coincida con ninguno de los datos de la tabla se guarda en el acumulador el número 80H.

### 3.1.11 SUBROUTINA HEX2ASCII

La subrutina HEX2ASCII sirve para transformar un número hexadecimal comprendido entre 00H y 0FH, a su correspondiente código ASCII. Esto se lo realiza con ayuda de la tabla 4.3. Entonces el valor menos significativo del DPL en donde inicia la tabla de conversión de código hexadecimal a ASCII, necesariamente tiene que ser cero; para de esa forma hacer un direccionamiento indirecto entre el DPTR y el acumulador y obtener el código ASCII guardado en la

localidad de memoria dependiendo de que localidad apunten conjuntamente el acumulador y el DPTR. El diagrama de flujo de esta subrutina es mostrada en la figura 3.11.

### **3.1.12 SUBROUTINA HEX2BCD**

Esta subrutina sirve para transformar un número hexadecimal de dos dígitos en código BCD, en donde las unidades, decenas y centenas se guardan en localidades de memoria diferente. El proceso de conversión se lo realiza a través de operaciones matemáticas en vez de utilizar tablas como los 2 casos anteriores. En este proceso se toma el dato hexadecimal y se lo divide para 0Ah, el residuo de esta primera operación se envía a localidad de memoria etiquetada como UNIDAD, después el resultado de la división se lo vuelve a cargar en el acumulador y se lo divide para 0Ah, pero esta vez el residuo se lo envía a la localidad de memoria etiquetada como DECENA y el resultado se lo envía a la localidad de memoria etiquetada como CENTENA. El diagrama de flujo de esta subrutina es el de la figura 3.12.

### **3.1.13 SUBROUTINA BCD2HEX**

Esta subrutina sirve para transformar un número BCD cuyas unidades, decenas y centenas residen en localidades de memoria diferentes. Cabe decir que esta subrutina sólo sirve para transformar números BCD comprendidos entre 0 y 255. Lo primero que se hace es tomar el dato que se encuentra en la localidad de memoria etiquetada como CENTENA y guardarlo en el acumulador para después multiplicarlo por 100. El resultado de esta operación se lo guarda en la localidad de memoria etiquetada como DATO. Después se mueve al acumulador el dato que se encuentra en la localidad de memoria etiquetada como DECENA, y se lo multiplica por 10. De ahí este resultado se lo suma al dato guardado en la localidad de memoria DATO y el resultado de esta operación se lo suma con la localidad de memoria etiquetada como UNIDAD y el resultado de toda esta operación es guardado en la localidad de memoria etiquetada como DATO que es

el resultado de la conversión. El diagrama de flujo de esta subrutina se muestra en la figura 3.13.

#### **3.1.14 SUBROUTINA INGRESE DATO**

Esta subrutina lo que hace es presentar una plantilla de entrada de datos cuya leyenda aparece en la primera fila y depende de la opción que el usuario escogió. Esta subrutina tiene un filtro que impide el ingreso de algún dato que no esté comprendido entre 000 y 100. Antes de llamar a esta subrutina se tiene que cargar en el DPTR la base de la tabla en donde se muestran los mensajes de la primera y segunda fila del LCD, la dirección del dispositivo al que será enviado el dato de ser aceptado y un comando para que el dispositivo esclavo sepa para que sirve el dato que se le está enviando. Una vez que se retorna de la subrutina el dato se guarda en determinada localidad de memoria dependiendo de la opción que se seleccionó. El diagrama de flujo es presentado en la figura 3.14. El usuario dispone de las opciones aceptar, borrar y cancelar la entrada de datos, con las teclas de función "A", "B", y "C" respectivamente.

#### **3.1.15 SUBROUTINA VER\_TEMPERATURA**

Esta subrutina sirve para visualizar en el LCD 2 valores: el porcentaje de potencia con el que está trabajando el dispositivo de calentamiento de la base y dependiendo de que opción se seleccionó; la temperatura en una de tres partes de la torre que pueden ser: dentro del reboiler, en la mitad de la torre, o en la parte más alta de la torre.

Es importante aclarar que cuando se está ejecutando esta subrutina también se está ejecutando la subrutina CONTROL. Para salir de esta opción y retornar al menú principal se necesita presionar la tecla C. Además se actualizan los datos de temperatura y de potencia cada vez que se reciben los datos del control de temperatura del reboiler. El diagrama de flujo de esta subrutina es el de la figura 3.15.

### **3.1.16 SUBROUTINA INT\_SERIAL**

Esta subrutina es la que se ejecuta cuando se produce una interrupción debida al puerto serial, esta subrutina sólo sirve para la adquisición de datos. Primero se recibe un comando después las tres temperaturas correspondientes a: la temperatura dentro del reboiler, la temperatura en la parte media de la torre y la temperatura en la parte alta de la torre. Y finalmente la potencia con la que está trabajando el dispositivo de calentamiento de la base. Al momento que se reciben en su totalidad los datos se cambia el valor de la variable BAN a OFFH, para que la subrutina VER\_TEMPERATURA, pueda actualizar los datos. El diagrama de flujo de esta subrutina se presenta en la figura 3.16.

### **3.1.17 SUBROUTINA ENVIAR**

Para utilizar esta subrutina lo primero que se hace es cargar en el acumulador el dato que va ha ser enviado y se procede a llamar a la subrutina ENVIAR. En la subrutina ENVIAR lo primero que se hace es deshabilitar todas las interrupciones para después borrar la bandera de transmisión TI. Luego se carga el dato guardado en el acumulador hacia el SBUF y se espera hasta que la bandera de transmisión pase a un estado 1L. Una vez que haya ocurrido esto, se borra la bandera de transmisión, se habilita la interrupción serial y se retorna de la subrutina. En el diagrama de flujo mostrado en la figura 3.17 se explica el funcionamiento de esta subrutina.

### **3.1.18 SUBROUTINA ENVIAR\_DATO**

Esta subrutina es utilizada por la subrutina INGRESE\_DATO. En la subrutina ENVIAR\_DATO lo primero que se hace es llamar a la subrutina BCD2HEX que transforma el dato BCD ingresado por el usuario a un dato de 2 dígitos hexadecimales. Una vez hecho esto, guarda el dato hexadecimal en la variable DATO\_INGRESADO, y procede a enviar la dirección del dispositivo al que va ha ser enviada la información hacia el puerto serial poniendo el bit TB8 en el estado de 1L. Después se coloca a 0L el bit TB8 para enviar datos. Lo primero que se

envía es un comando que se guarda en la localidad de memoria denominada COMANDOS y finalmente se envía el dato guardado en la localidad de memoria DATO\_INGRESADO. Todo esto se lo puede ver en el diagrama de flujo de la figura 3.18.

### **3.1.19 SUBROUTINA BORRAR**

Esta subrutina es utilizada por la subrutina INGRESE\_DATO cuando es presionada la tecla "B". Dependiendo del valor que tenga R7, que indirectamente indica el número de datos ingresados por el usuario, la subrutina BORRAR, borrará 1, 2, 3 o ningún dato, dejando otra vez los espacios libres en el LCD para que el usuario pueda realizar una nueva entrada de datos. El diagrama de flujo de esta subrutina es el de la figura 3.19.

### **3.1.20 SUBROUTINA CONTROL**

Esta subrutina se ejecuta permanentemente, tiene 2 funciones principales: la primera es enviar un comando de encender el ventilador, hacia el control del acumulador, ventilador y bombas, cuando la temperatura en medio de la torre sea mayor o igual a 50 grados centígrados. Caso contrario cuando la temperatura es menor a 50 grados centígrados envía al mismo control un comando para apagar el ventilador. Y la segunda función es la de enviar un comando de abrir la válvula de fondos, hacia el control de nivel del reboiler, cuando se haya alcanzado el SET POINT de temperatura de fondos. Caso contrario envía un comando al mismo control para cerrar la válvula de fondos por completo.

Existen dos localidades de memoria denominadas BANDERA\_1 y BANDERA\_2 que sirven como banderas, para evitar que se envíe el mismo comando más de dos veces al mismo control.

### 3.1.21 SUBROUTINA BAJAR\_MENU

Esta subrutina se ejecuta cuando se presiona la tecla "\*" cuando el LCD muestra el menú principal. Sirve para visualizar las dos opciones siguientes del menú principal. Entonces lo primero que hace es llamar a la subrutina CLS para borrar las dos opciones actuales que se muestran en el LCD. Después se compara si se estaban visualizando las dos últimas opciones del menú principal; si es así se carga el DPTR con las dos primeras opciones del menú principal; sino se apunta con el DPTR a las dos siguientes opciones del menú principal. El diagrama de flujo de esta subrutina es el mostrado en la figura 3.21.

### 3.1.22 SUBROUTINA SUBIR\_MENU

Esta subrutina se ejecuta cuando se presiona la tecla "#" cuando el LCD muestra el menú principal. Sirve para visualizar las dos opciones anteriores del menú principal. Lo primero que hace esta subrutina es llamar a la subrutina CLS para borrar las dos opciones actuales que se muestran en el LCD. Después se compara si se estaban visualizando las dos primeras opciones del menú principal, si es así se carga al DPTR con las dos últimas opciones del menú principal caso contrario se apunta con el DPTR a las dos opciones anteriores. El diagrama de flujo de esta subrutina es mostrado en la figura 3.22.

### 3.1.23 SUBROUTINA CLAVE

Esta subrutina se ejecuta cuando se presiona la tecla "D" cuando se visualiza el menú principal. Cuando se esta en el menú principal de control automático y se presiona esta tecla, se entra en una plantilla de entrada de datos en donde el usuario debe ingresar cuatro números. Si los cuatro números coinciden con los cuatro números previamente grabados al inicio del programa principal entonces se pasa un nuevo menú que es el de control manual. También se envía a todos los esclavos el número 0FH como dirección con lo que todos los dispositivos esclavos pasan a modo manual.

Estando en el menú de control manual y si se presiona la tecla "D" automáticamente se retorna al menú de control automático. Además se envía por separado a cada uno de los dispositivos esclavos el dato F0H para que retornen a modo automático. El diagrama de flujo de esta subrutina es el de la figura 3.23.

## PROGRAMA PRINCIPAL

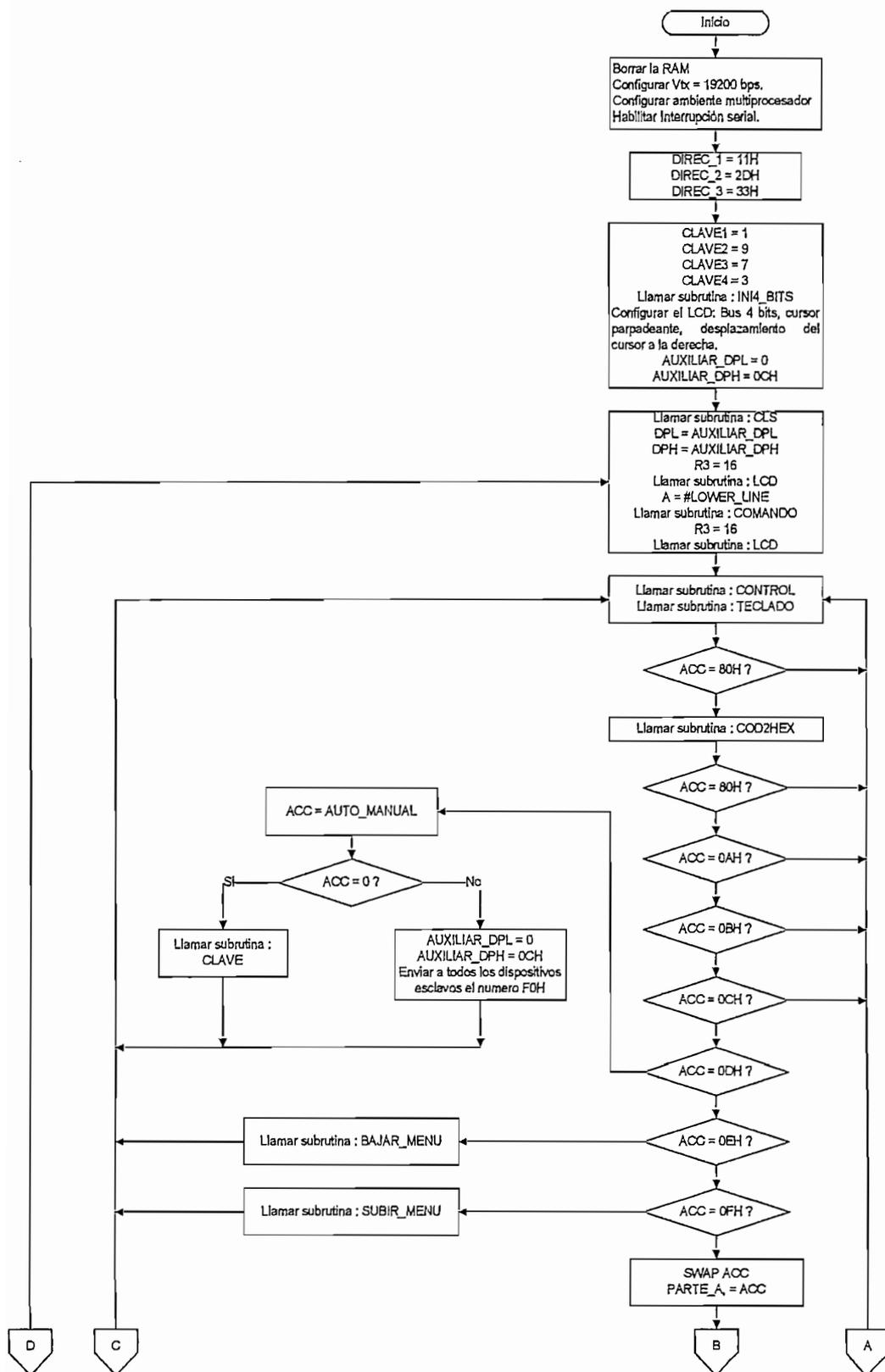


Figura 3.1.a

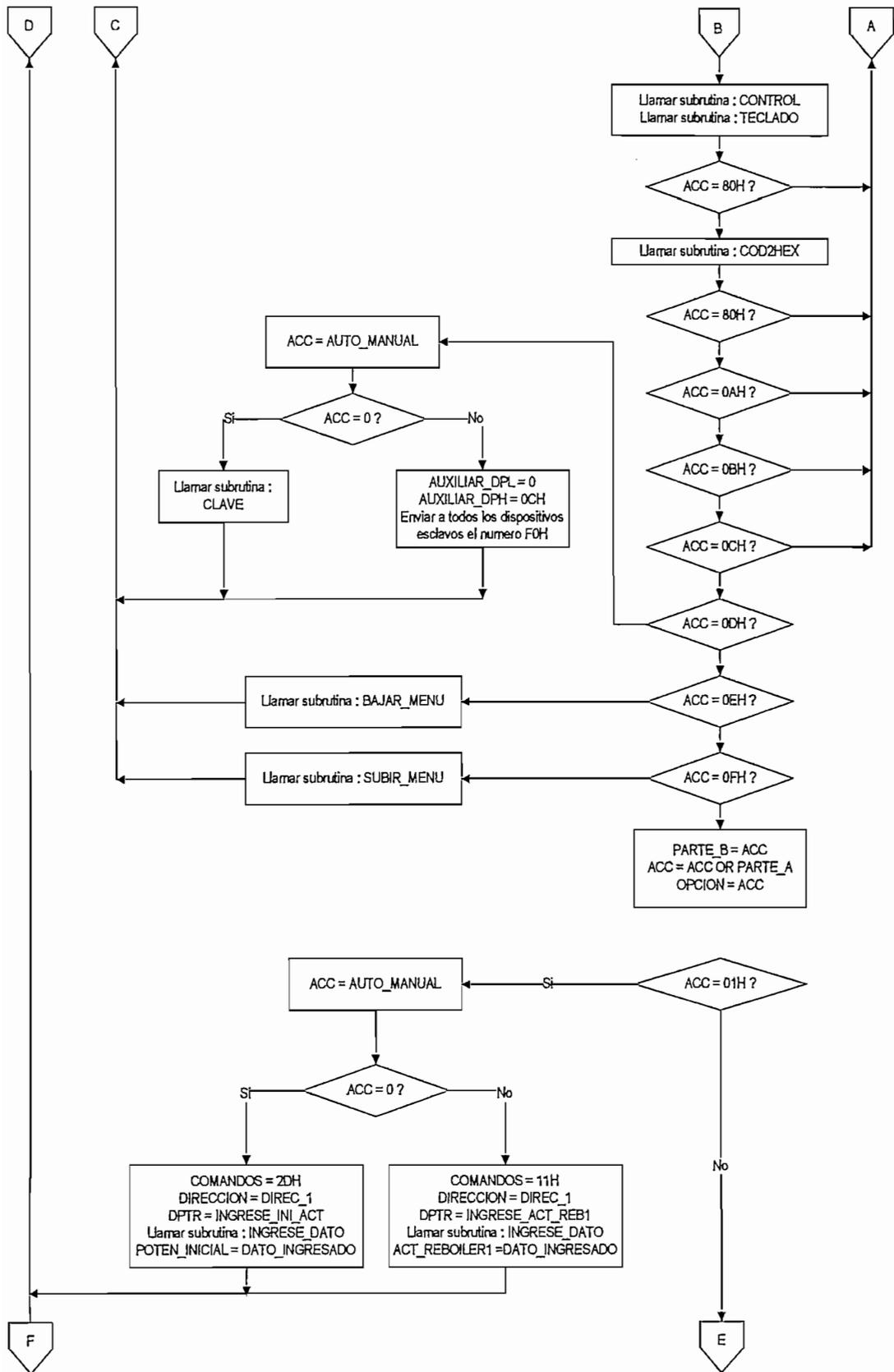


Figura 3.1.b

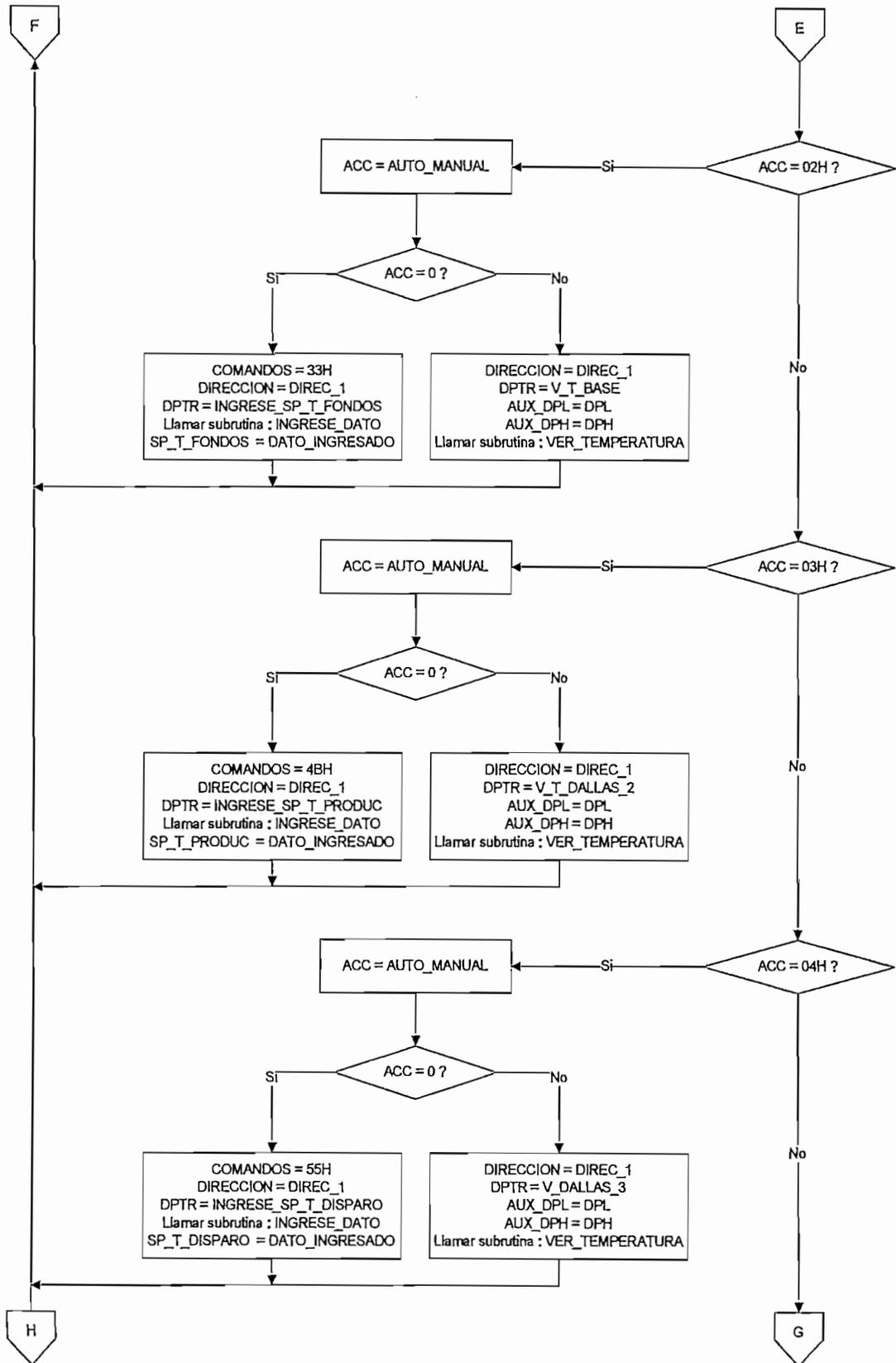


Figura 3.1.c

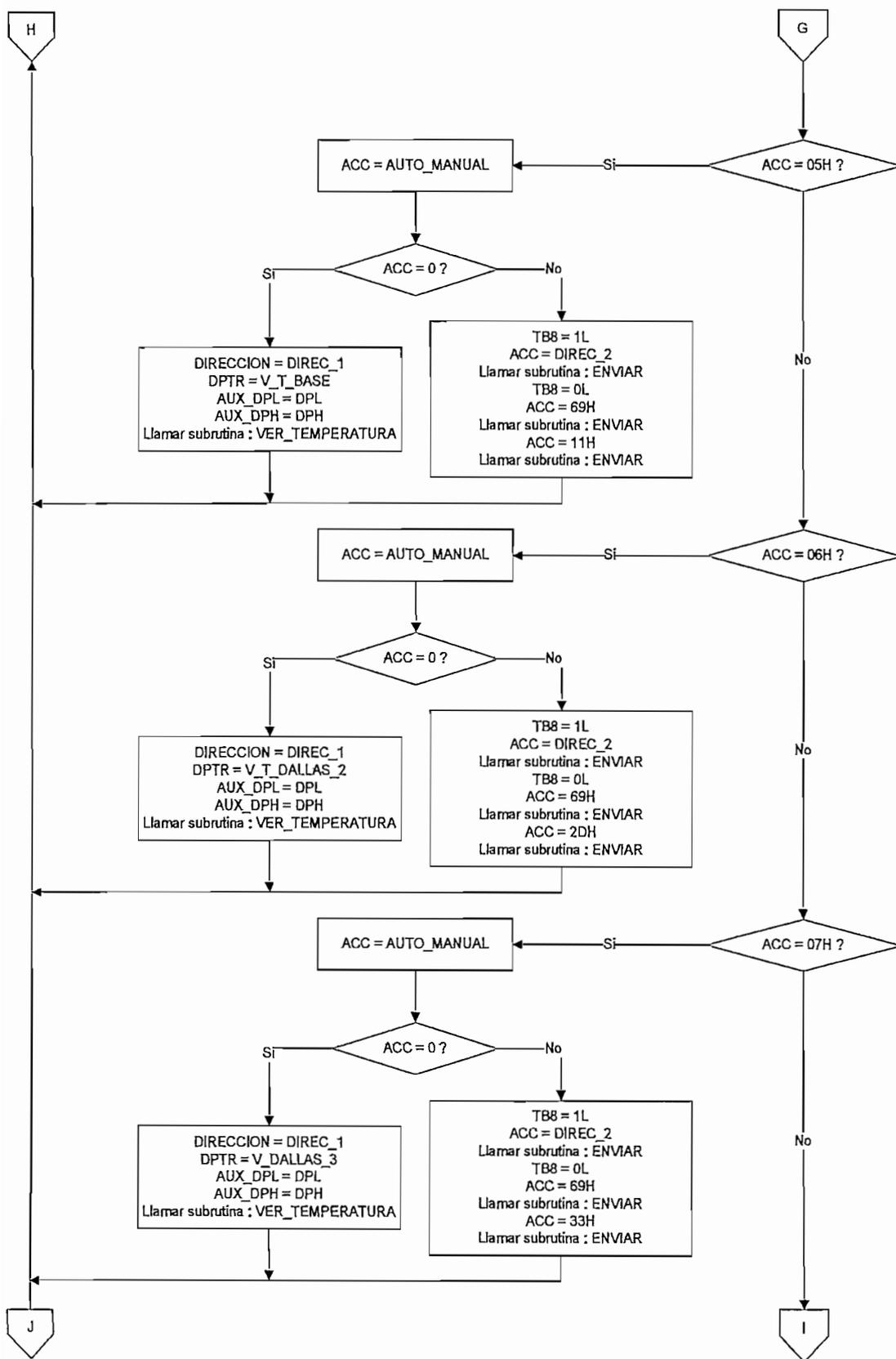


Figura 3.1.d

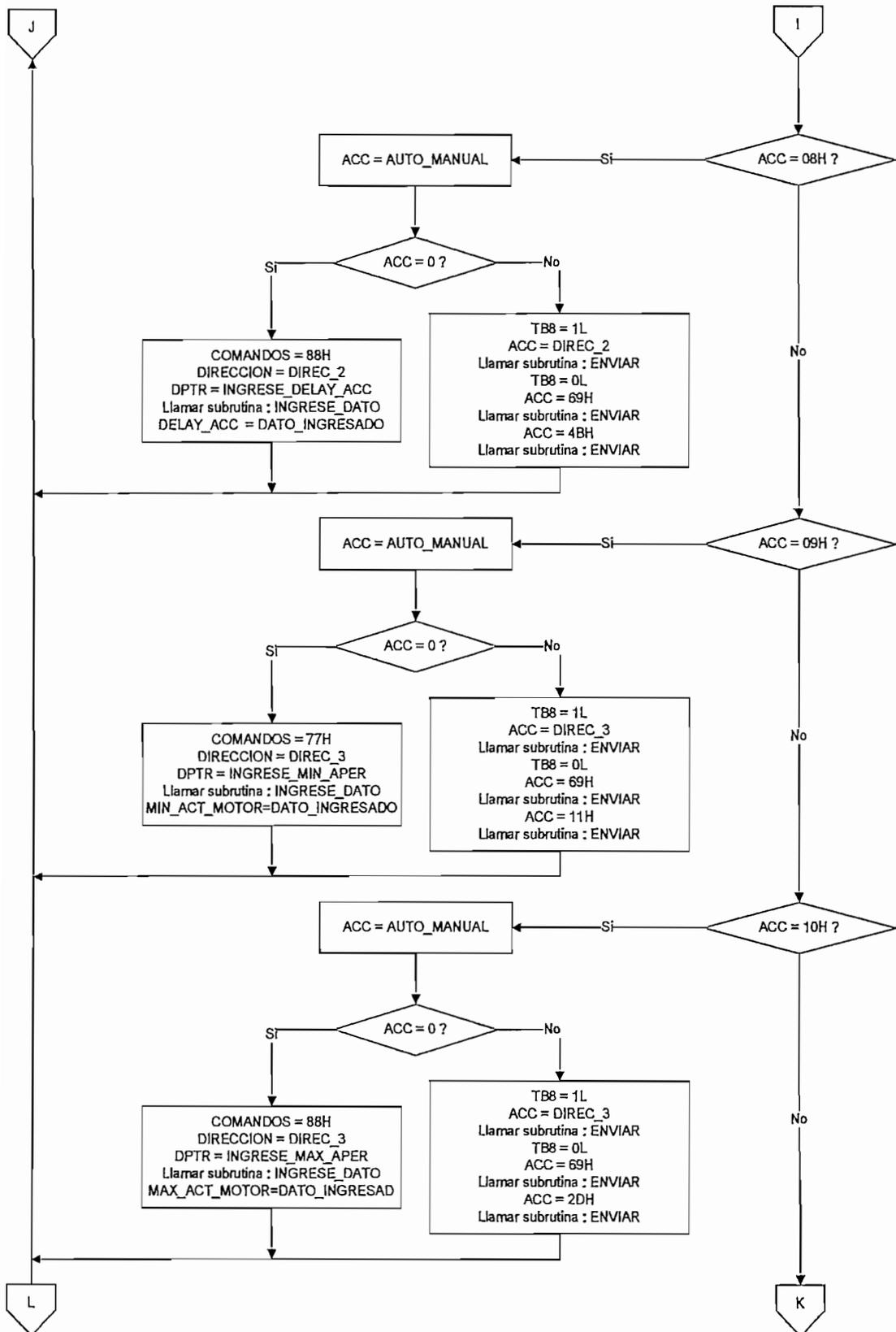


Figura 3.1.e

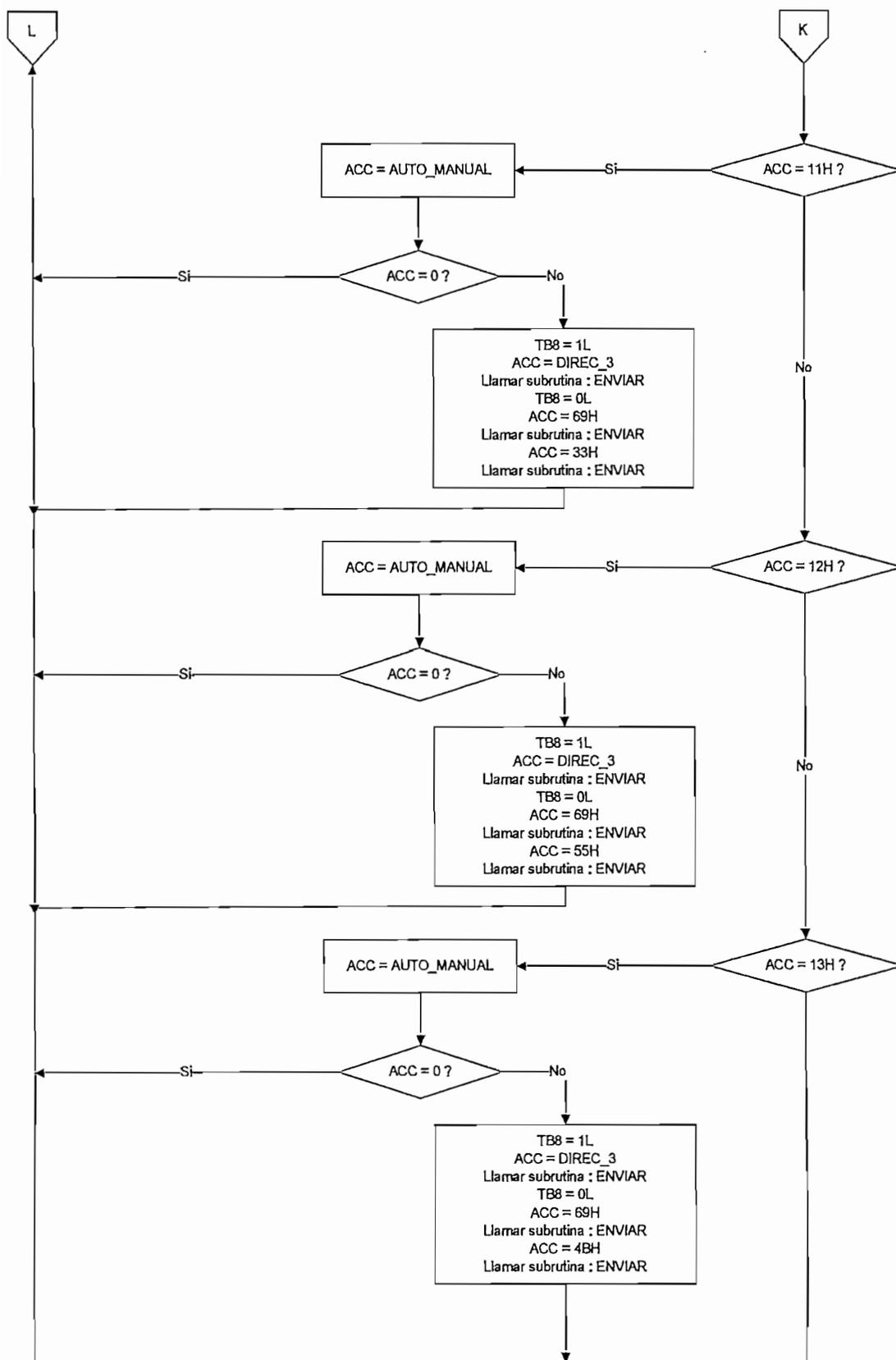


Figura 3.1.f

## SUBROUTINA INI\_4BITS

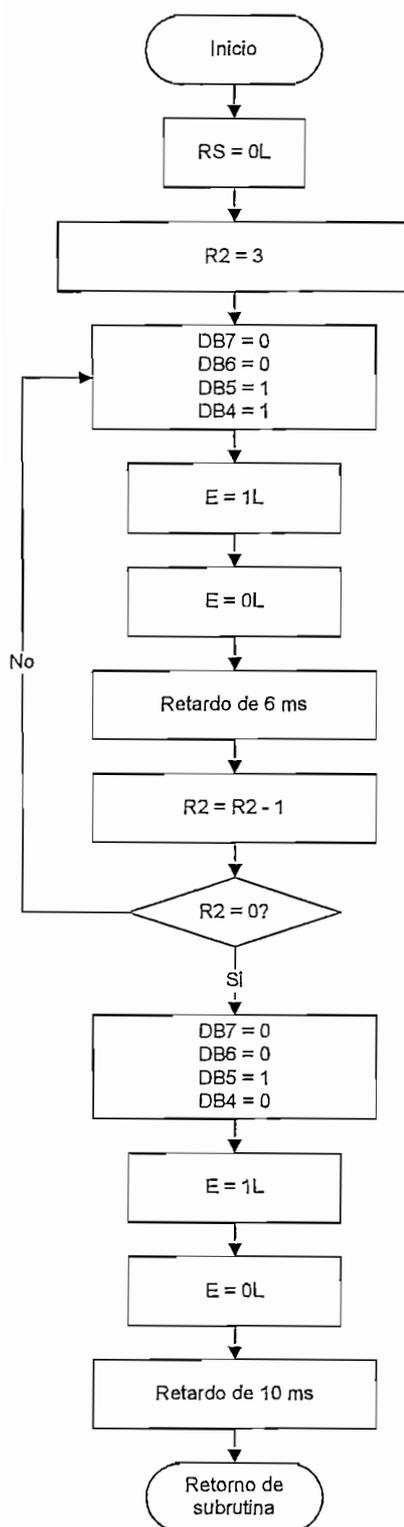


Figura 3.2

## SUBROUTINA COMANDO

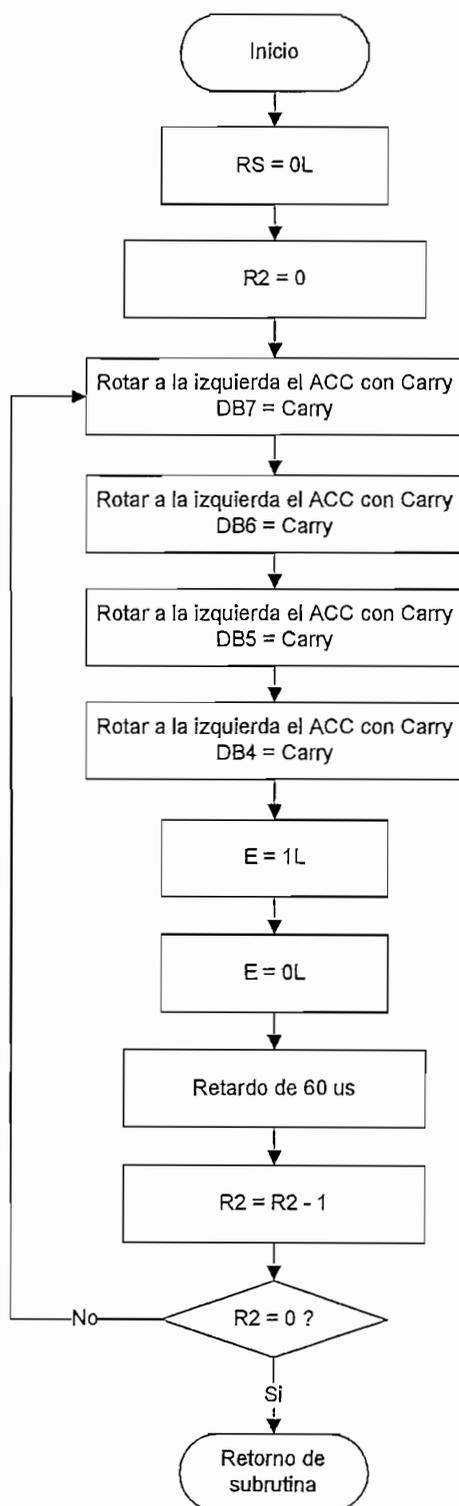


Figura 3.3

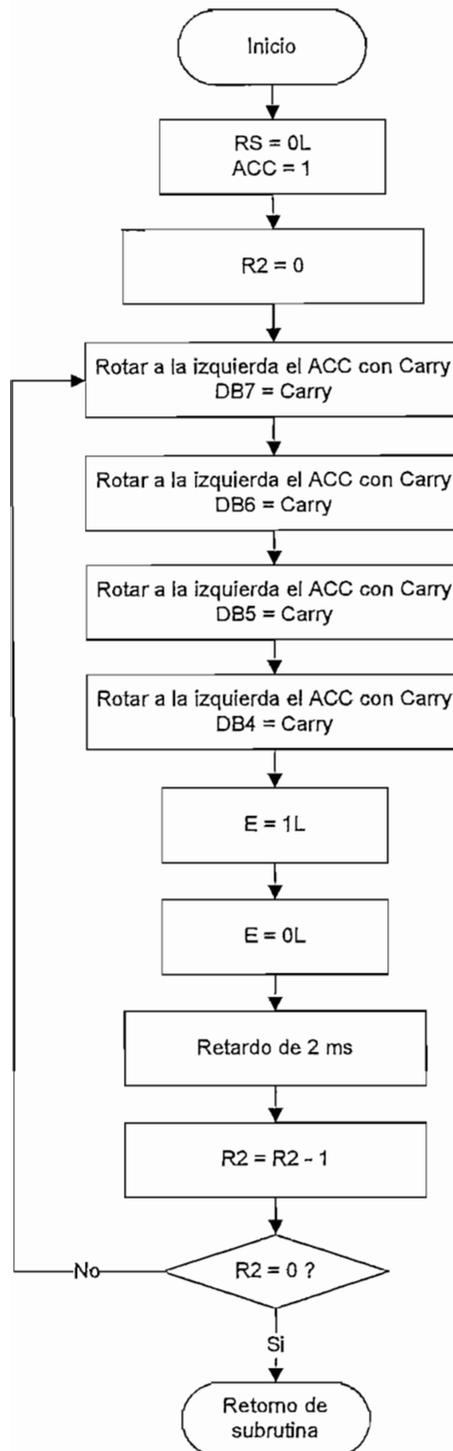
**SUBROUTINA CLS**

Figura 3.4

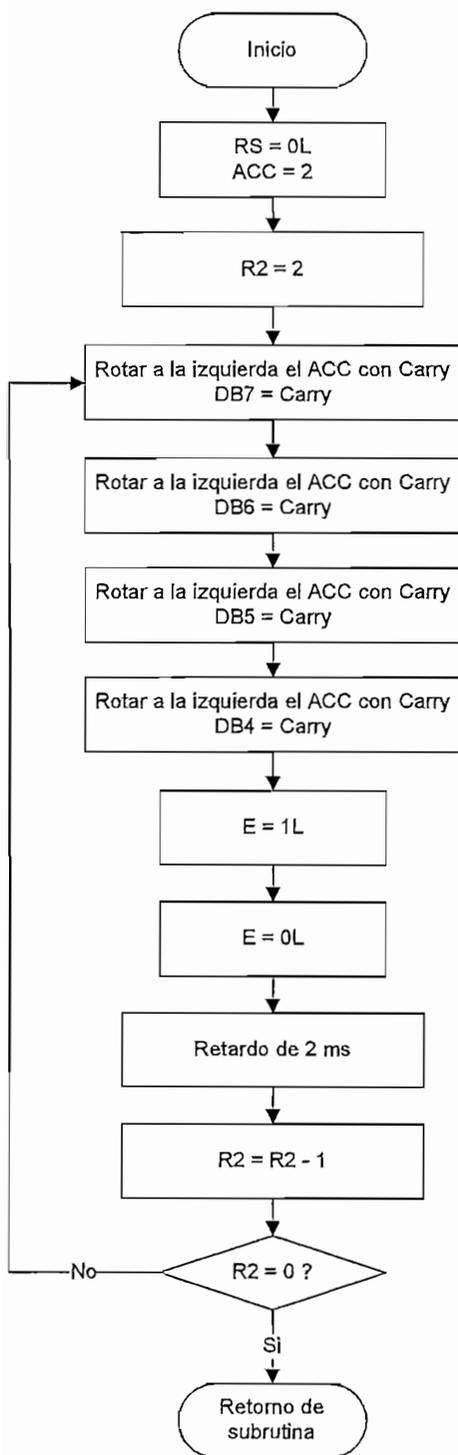
**SUBROUTINA HOME**

Figura 3.5

## SUBROUTINA CARÁCTER

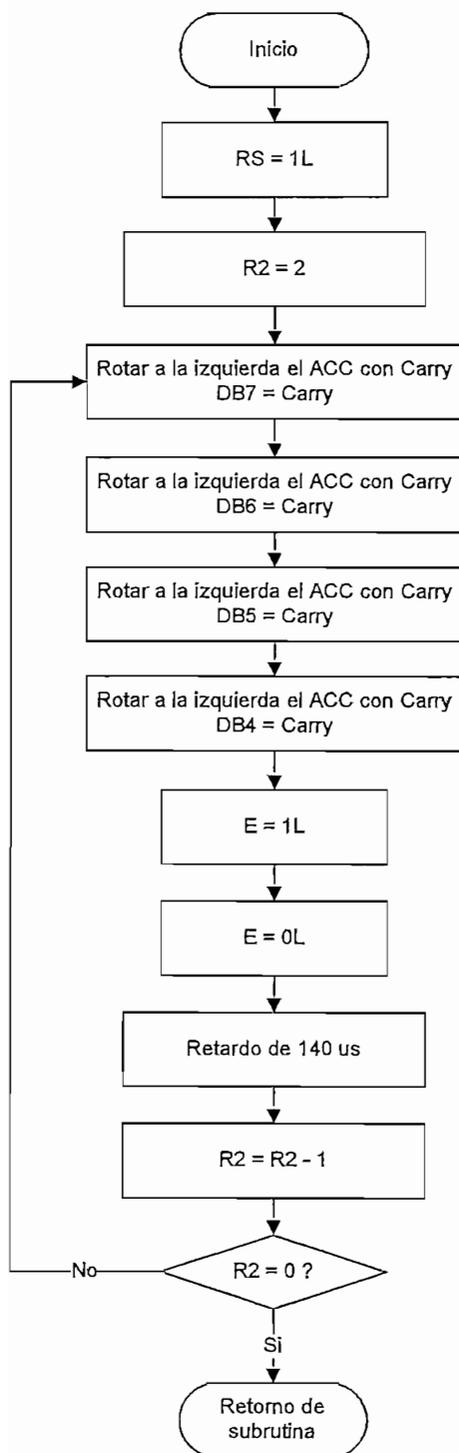


Figura 3.6

## SUBROUTINA LCD

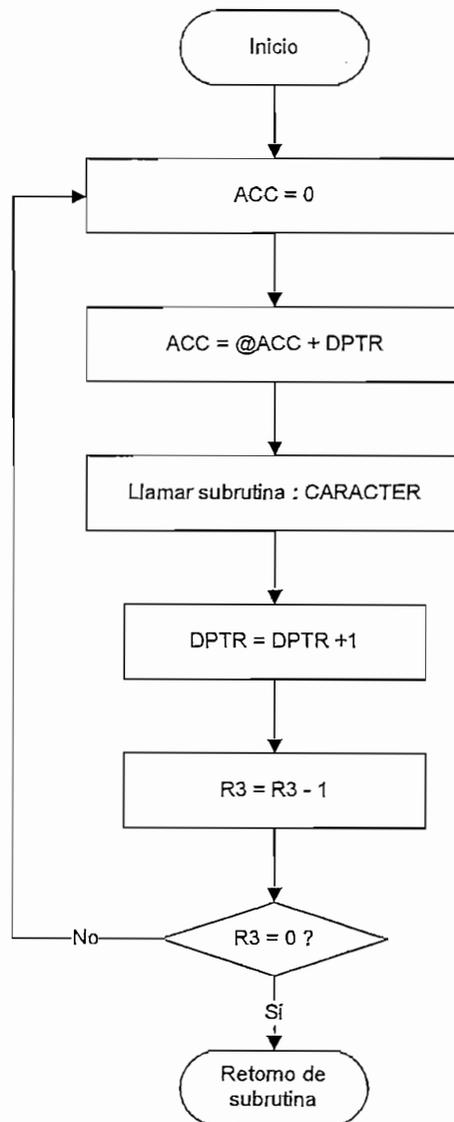


Figura 3.7

## SUBROUTINA TECLADO

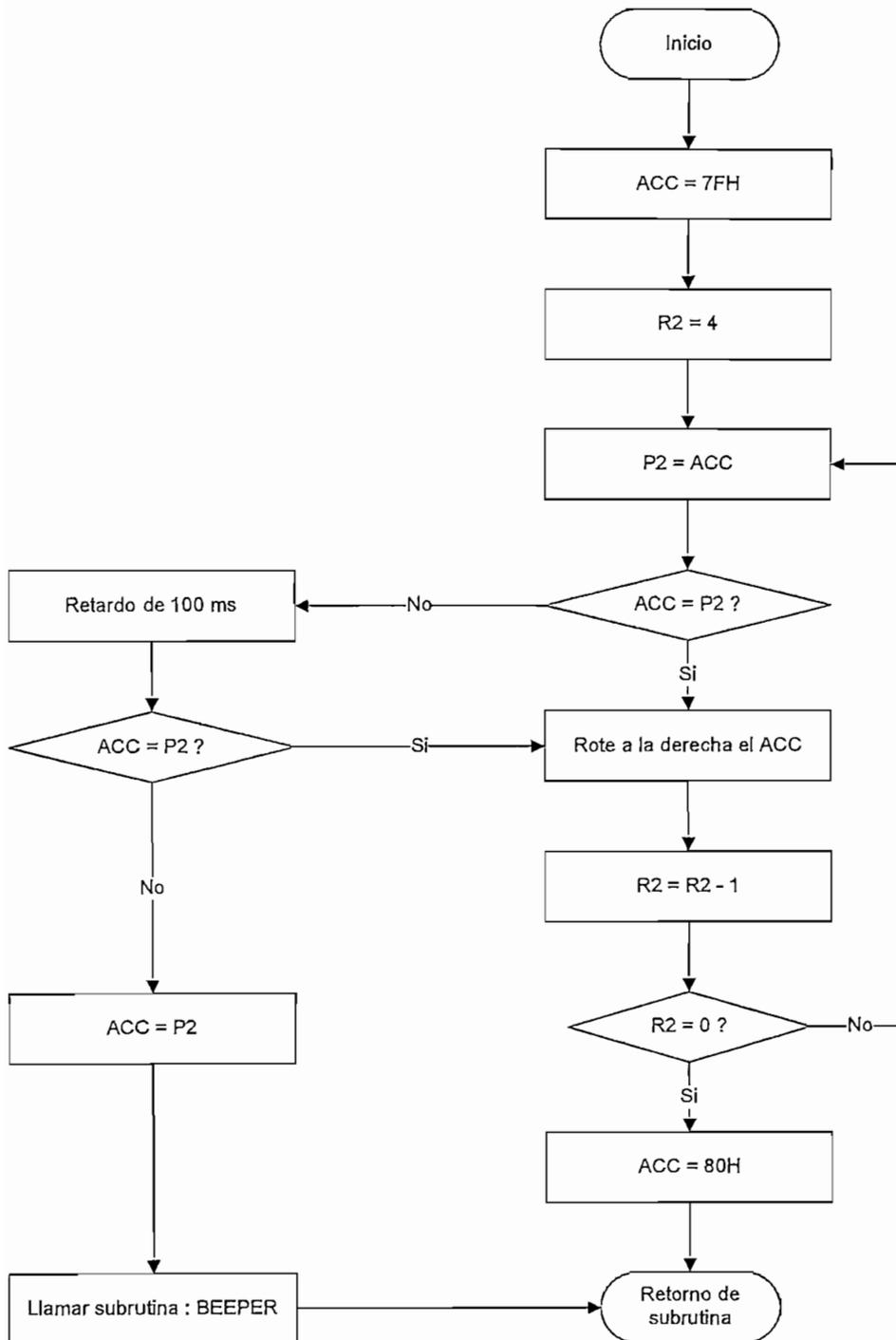


Figura 3.8

## SUBROUTINA BEEPER

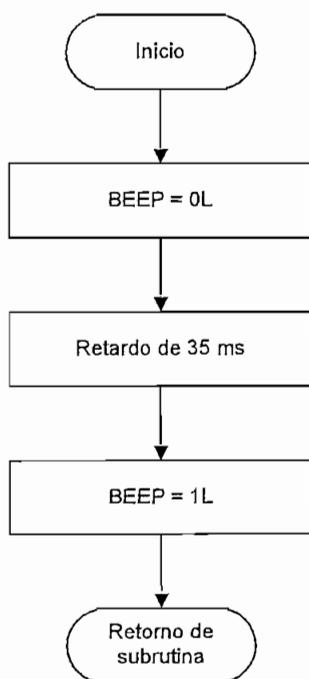


Figura 3.9

## SUBROUTINA COD2HEX

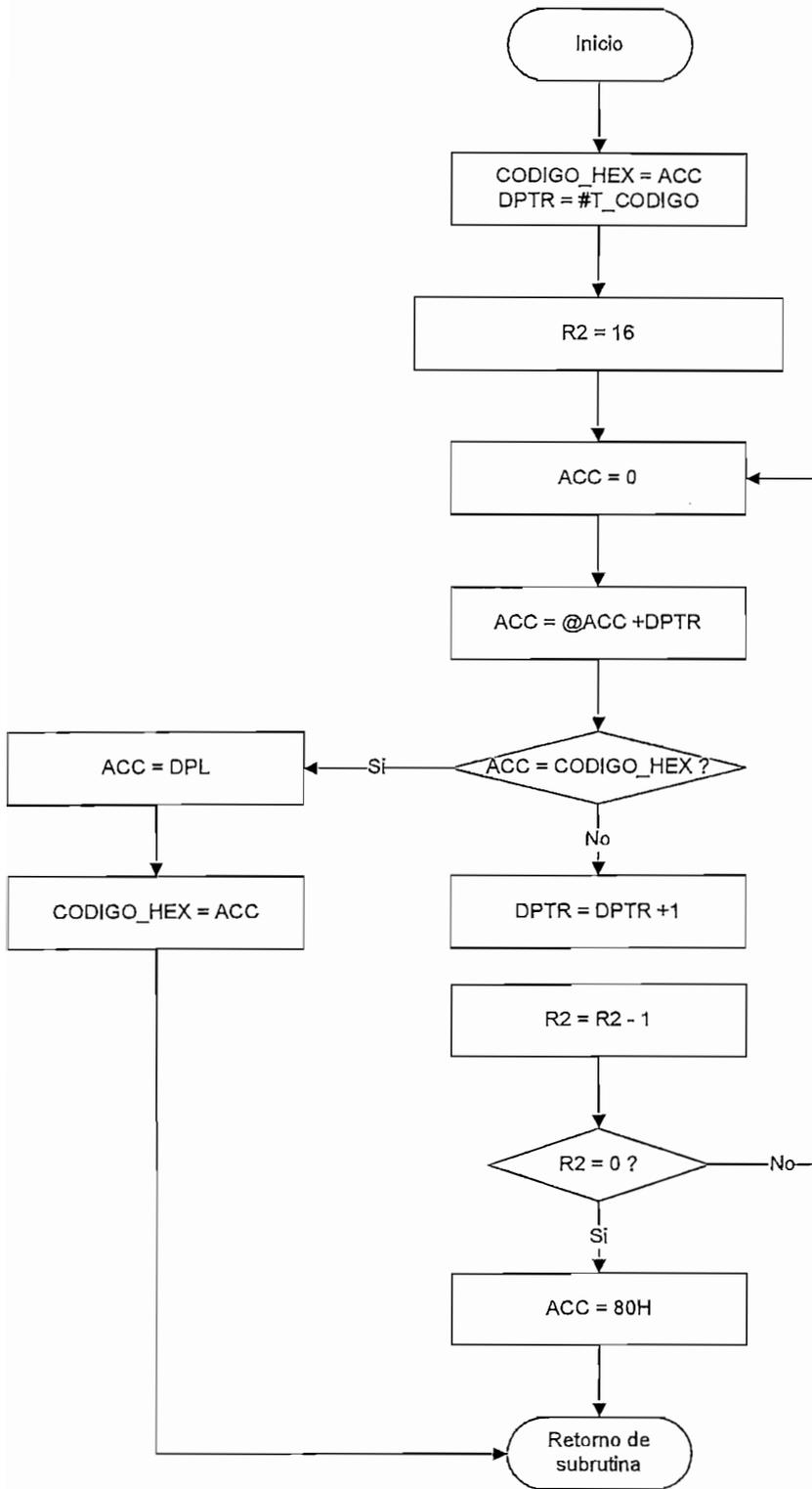


Figura 3.10

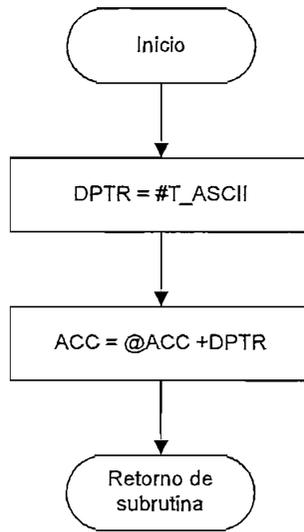
**SUBROUTINA HEX2ASCII**

Figura 3.11

## SUBROUTINA HEX2BCD

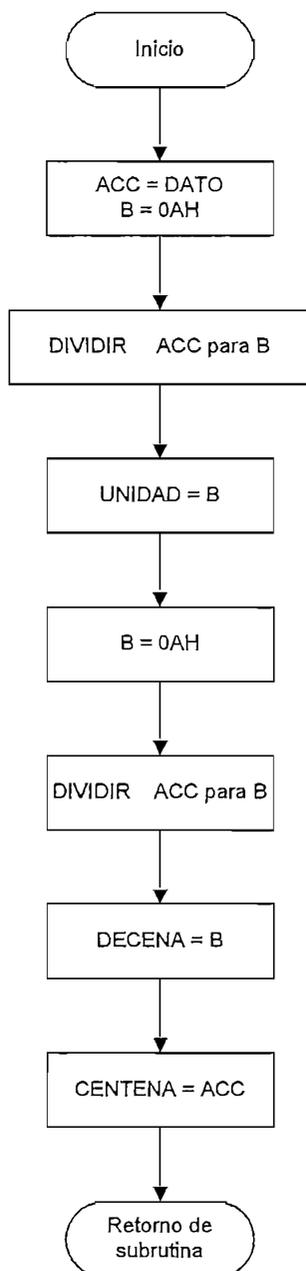


Figura 3.12

## SUBROUTINA BCD2HEX

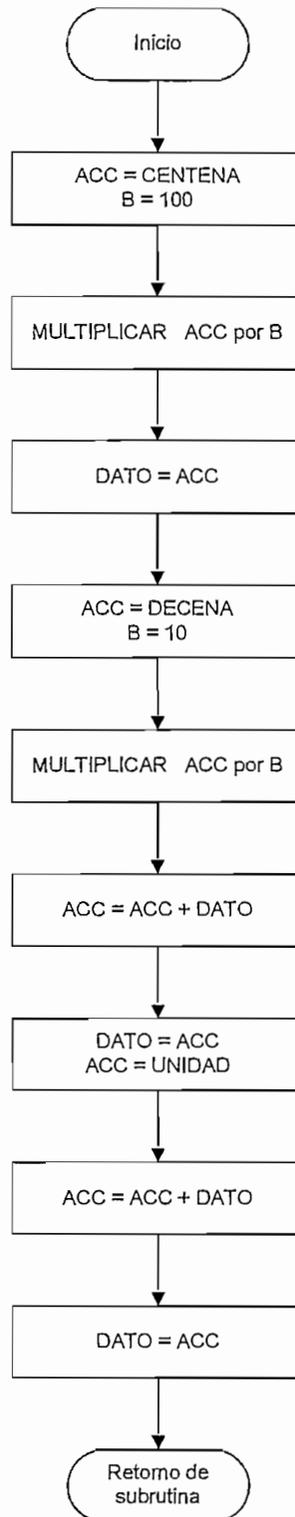


Figura 3.13

## SUBROUTINA INGRESE\_DATO

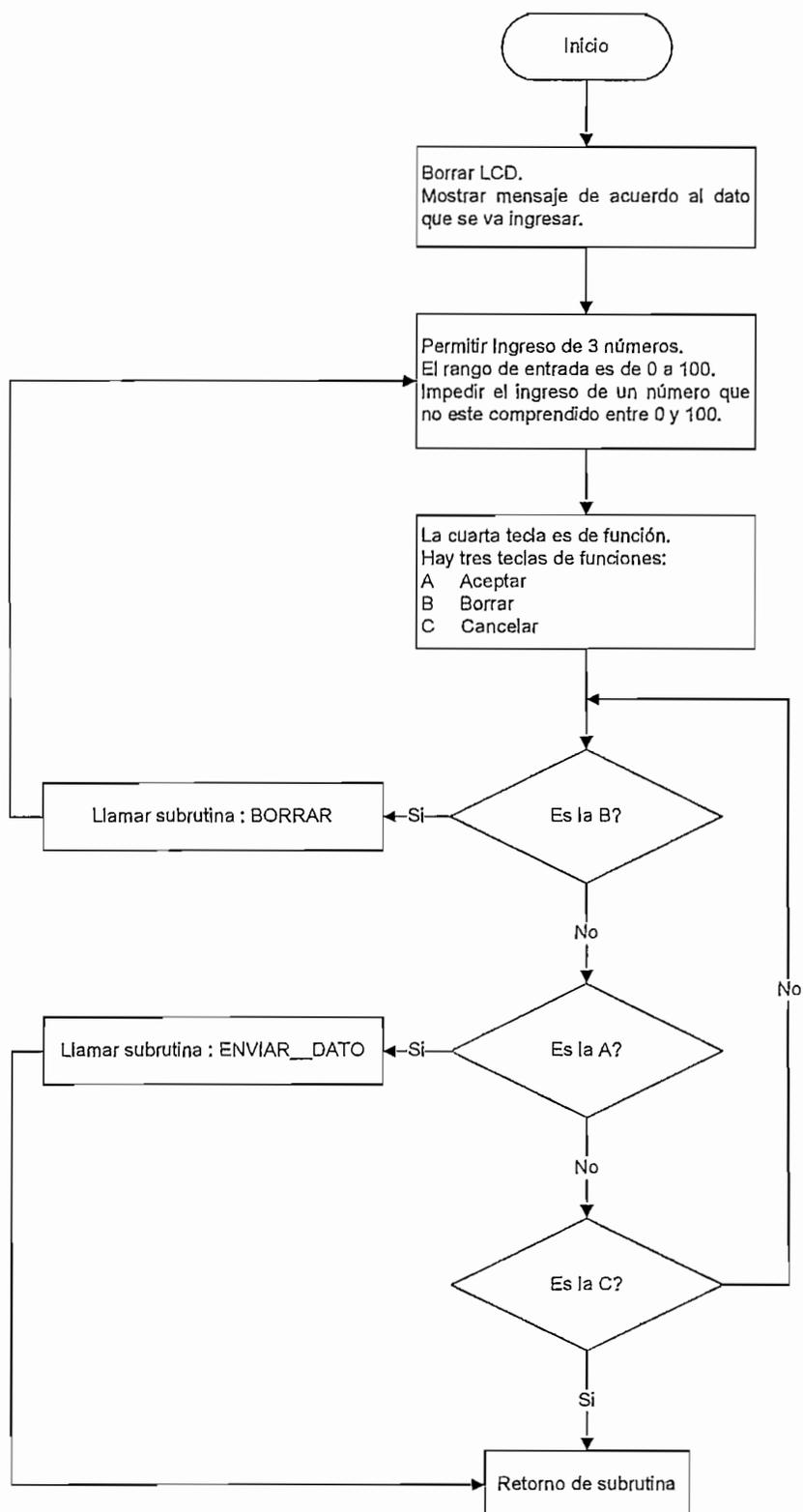


Figura 3.14

## SUBROUTINA VER\_TEMPERATURA

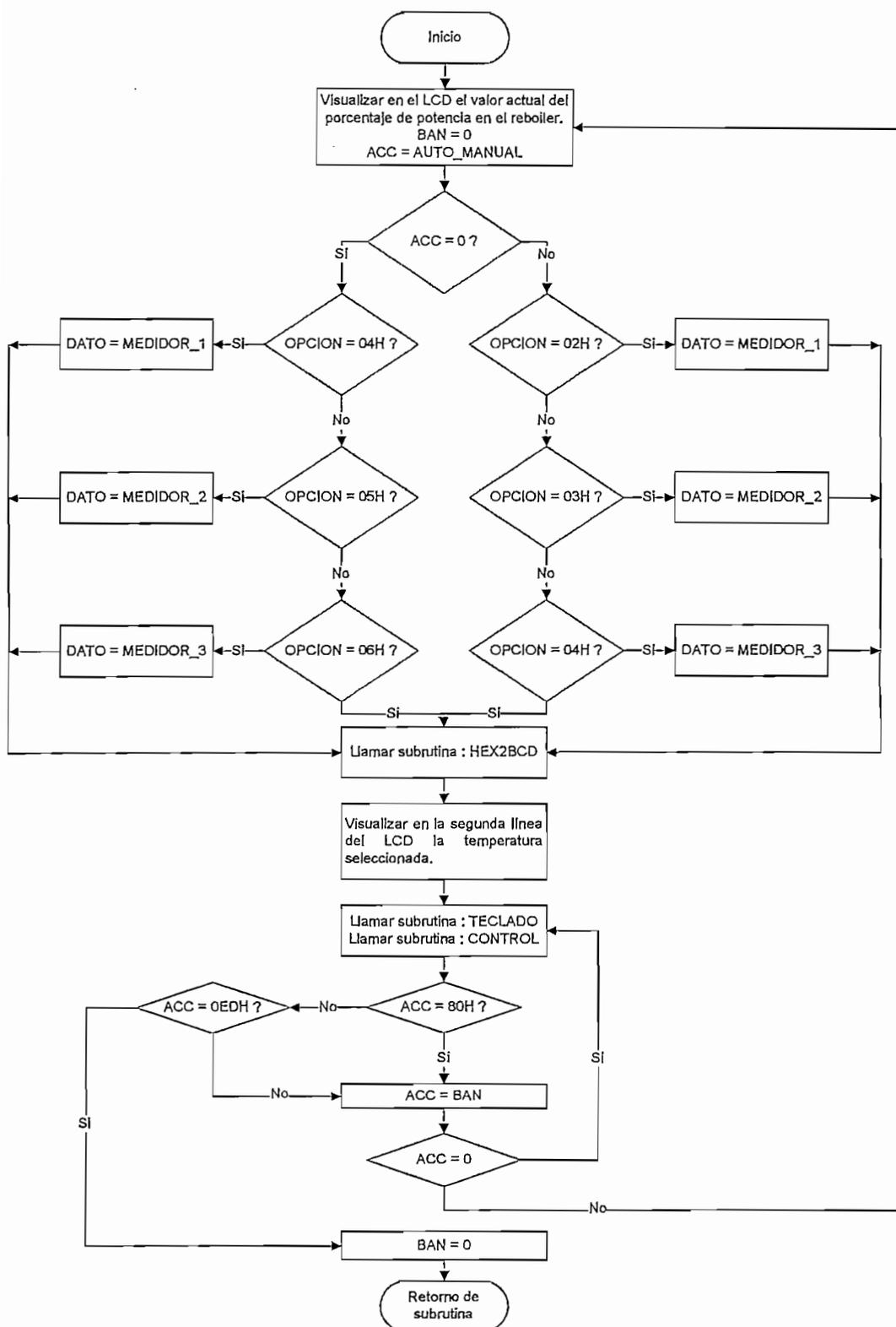


Figura 3.15

## SUBROUTINA INT\_SERIAL

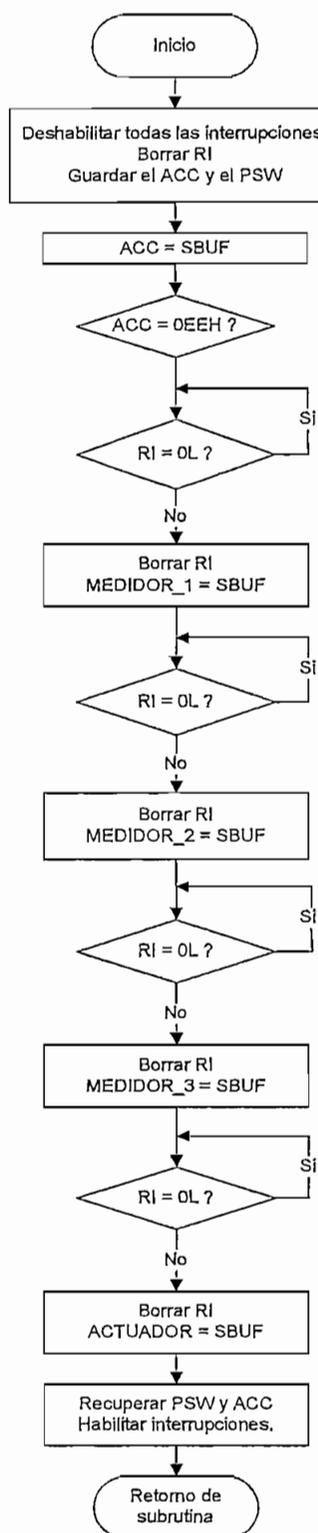


Figura 3.16

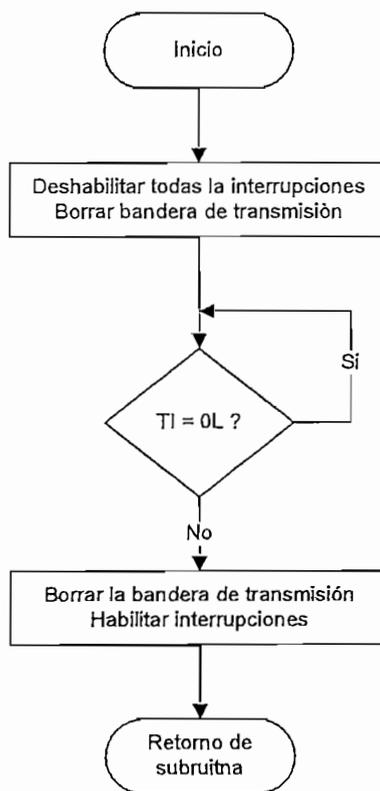
**SUBROUTINA ENVIAR**

Figura 3.17

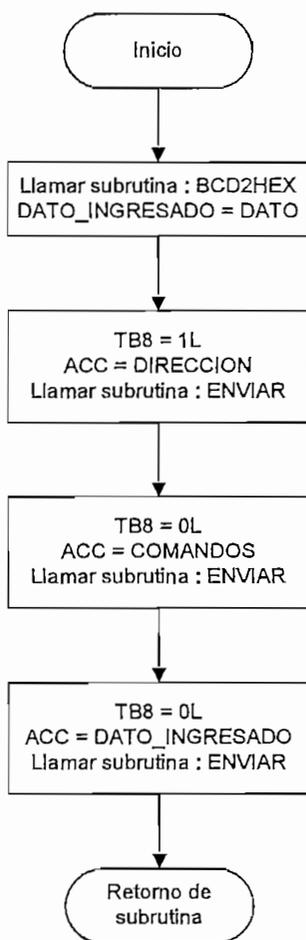
**SUBROUTINA ENVIAR\_DATO**

Figura 3.18

## SUBROUTINA BORRAR

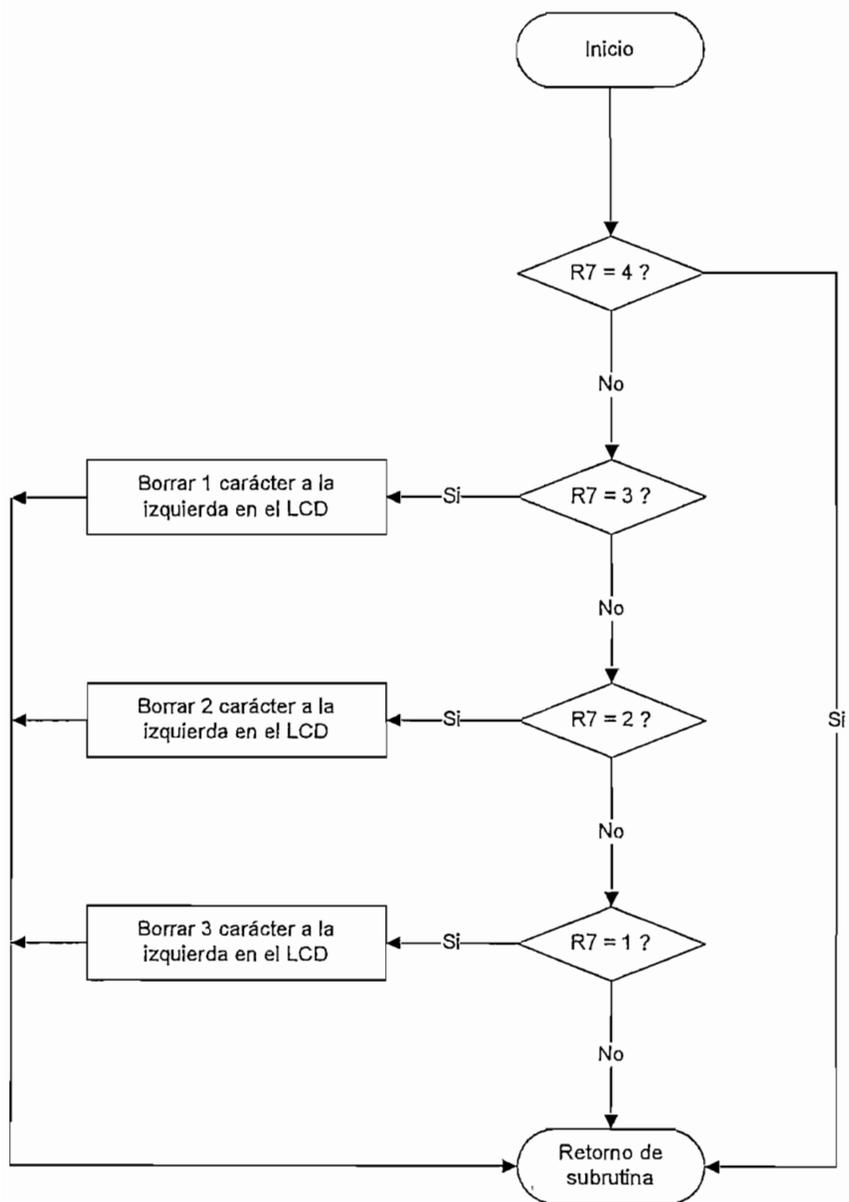


Figura 3.19

## SUBROUTINA CONTROL

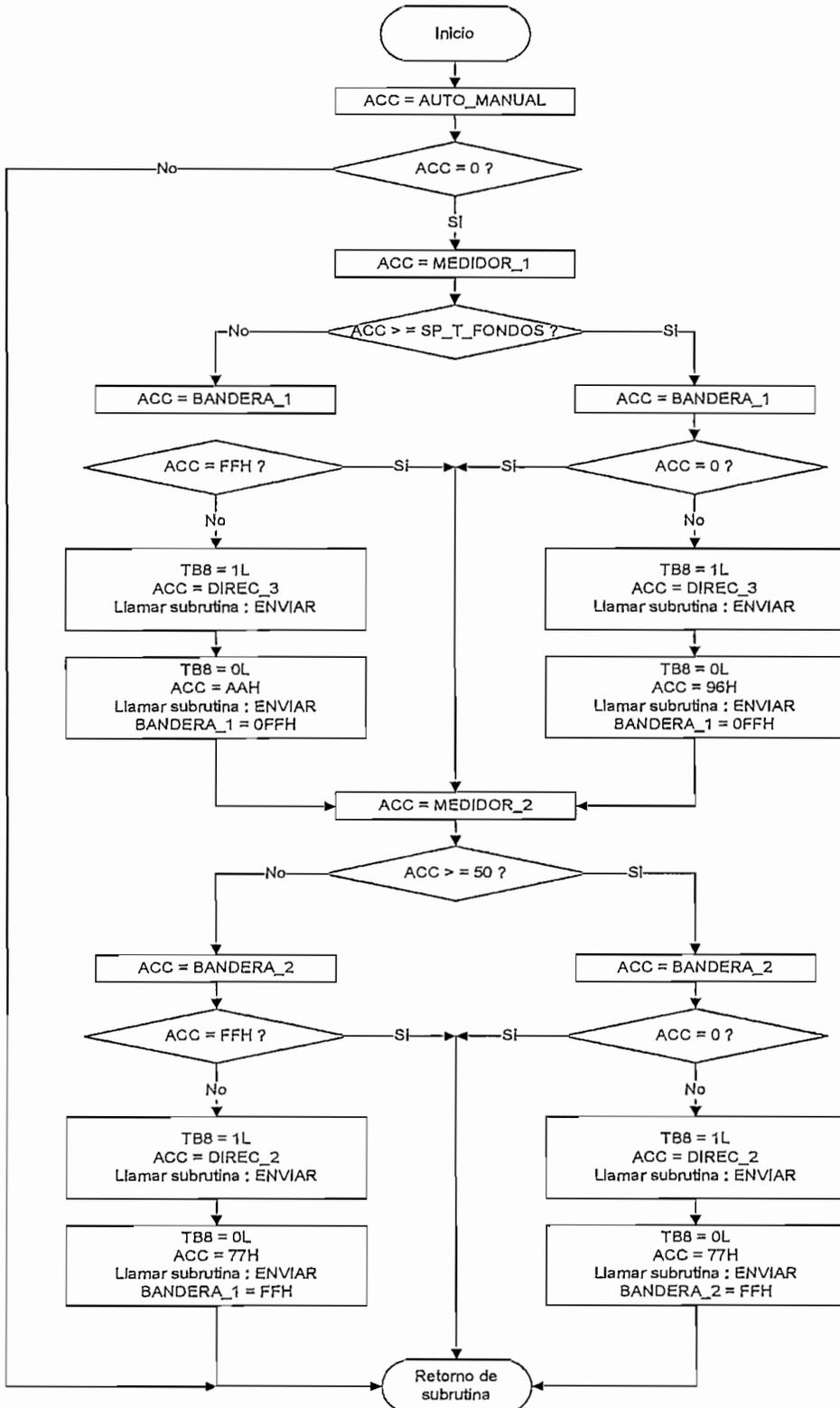


Figura 3.20

## SUBROUTINA BAJAR\_MENU

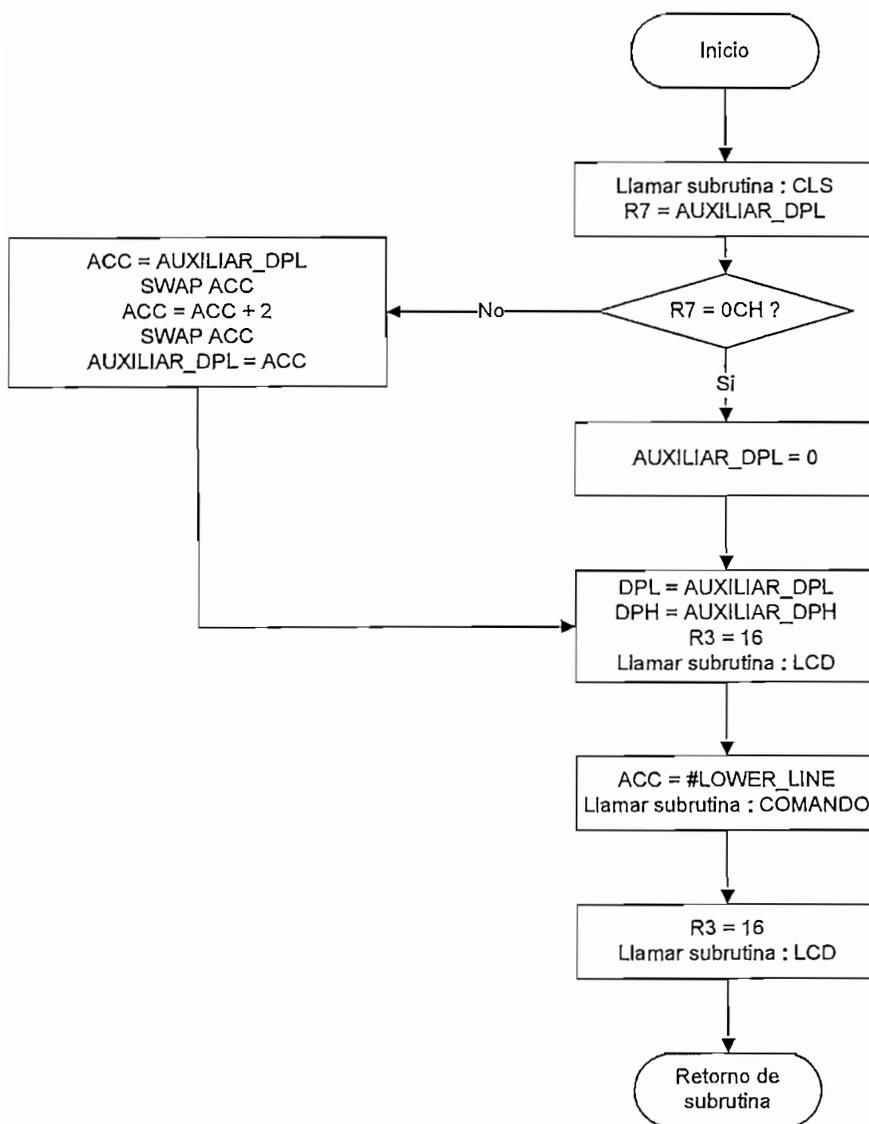


Figura 3.21

## SUBROUTINA SUBIR\_MENU

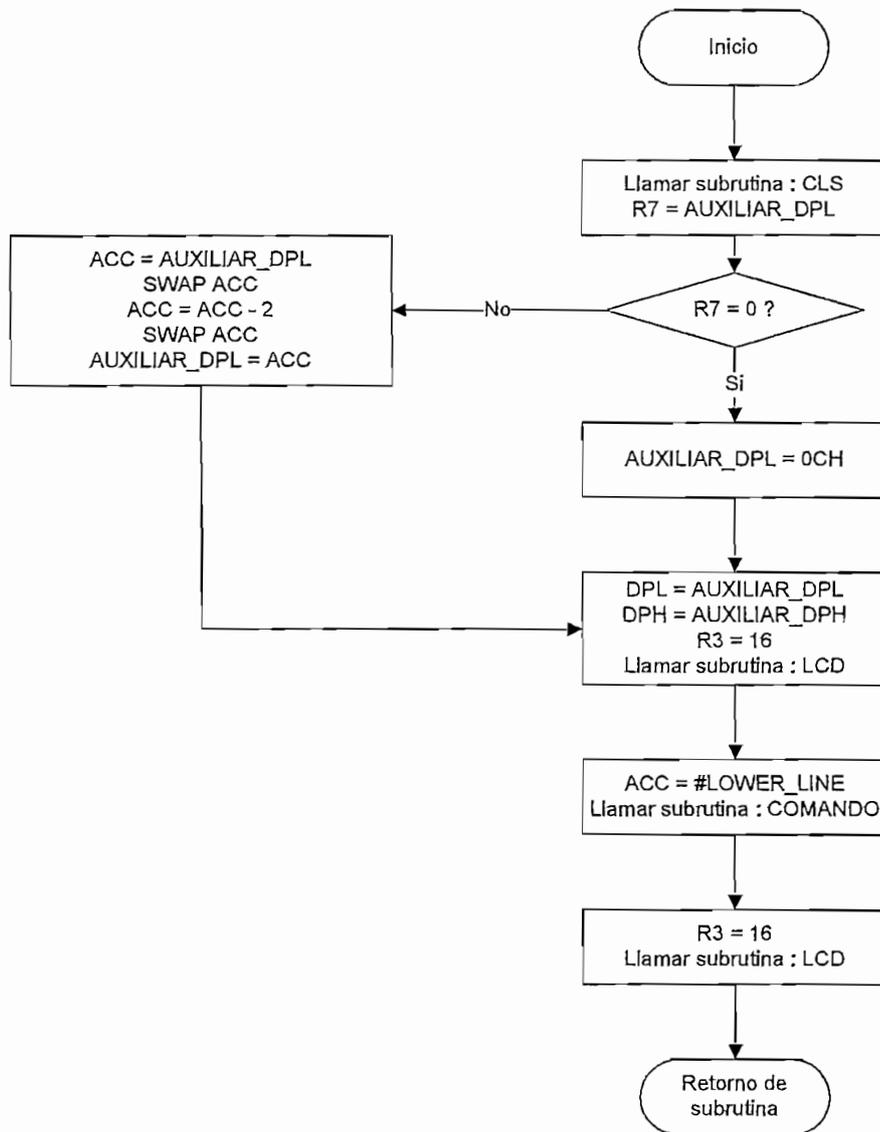


Figura 3.22

## **3.2 SUBROUTINAS DEL CONTROL DE TEMPERATURA DEL REBOILER**

El programa en lenguaje assembler que se ejecuta en el microcontrolador AT89C2051 para manejar el control de temperatura del reboiler se encuentra en el anexo 2.

Cabe recalcar que el análisis de los siguientes diagramas de flujo deber ser realizado conjuntamente con el listado de etiquetas que se incluye en el anexo 2.

### **3.2.1 PROGRAMA PRINCIPAL**

El programa principal del control de temperatura del reboiler en su inicio se encarga de borrar la memoria RAM, así como también configurar los parámetros de la comunicación serial a 19200 bps, 9 bits de datos y ambiente multiprocesador. También se colocan valores por defecto de los SET POINTs de temperatura de fondos y productos, la temperatura de disparo. Además se inicia con la potencia máxima de la resistencia, es decir el valor de la variable ACTUADOR igual a 0 y se coloca la dirección de este dispositivo igual a 11H. En este inicio del programa también se inicializa los DS1821 como termómetros de lecturas continuas. De ahí en adelante se entra en un bucle repetitivo; en el cual dependiendo del valor de R4 se ejecutará la subrutina MANUAL o la subrutina AUTOMATICO. El diagrama de flujo del programa principal, del cual se derivan el resto de subrutinas excepto las de interrupción, se muestra en la figura 3.24.

### **3.2.2 SUBROUTINA AUTOMÁTICO**

Lo primero que se hace es cargar dos contadores que serán los registros R6 y R7 para que cuenten 512 periodos de la señal de cruce por cero que se lee en el pin etiquetado como CRUCE\_CERO. Durante cada periodo de la señal de cruce por cero, se realiza el disparo del TRIAC, dependiendo del valor del actuador. El programa se salta la parte de disparo del TRIAC si el pin etiquetado como ENABLE se encuentra en un estado 1L. Una vez que los contadores hayan

terminado de contar 512 se procede a complementar el bit etiquetado como muestreo, que es la referencia para el usuario para indicar que se tomaron medidas de temperatura de la torre. También se toma medidas de temperatura y se las envía al panel principal. Después se vuelve a llamar a la subrutina ACT\_ENABLE para saber el estado del pin ENABLE, si es igual a 1L se sale de la subrutina, pero si su estado es igual a 0L se ejecuta la subrutina CONTROL\_TEMPE. El diagrama de flujo de esta subrutina se muestra en la figura 3.25.

### 3.2.3 SUBROUTINA MANUAL

La subrutina MANUAL es muy similar a la subrutina AUTOMATICO, la única diferencia es que aquí no se hace un llamado a la subrutina CONTROL\_TEMPE. En la subrutina MANUAL los registros R6 y R7 hacen también la función de contadores, que cuentan 512 periodos de la señal del detector de cruce por cero, que ingresa por el pin etiquetado como CRUCE\_CERO. En cada periodo de la señal de cruce por cero se realiza el disparo del TRIAC, al valor que tenga la variable ACTUADOR. Cabe decir que al no existir el llamado a la subrutina CONTROL\_TEMPE, la única forma de que el ACTUADOR, pueda cambiar de valor, es por medio del ingreso de algún valor que el usuario ingrese por teclado en el menú de control manual. Después de que los contadores terminen de contar los 512 periodos de la señal de cruce por cero, se complementa el bit etiquetado como MUESTREO, de ahí se toman medidas de temperatura y son enviadas al panel principal y terminado esto se retorna de la subrutina. El diagrama de flujo que explica el funcionamiento de esta subrutina se muestra en la figura 3.26

### 3.2.4 SUBROUTINA INT\_SERIAL

Al momento que se produce una interrupción serial debido a que el bit RB8 es 1L en el modo ambiente multiprocesador, esta subrutina lo primero que hace es deshabilitar todas las interrupciones, borra la bandera de recepción RI, guarda el acumulador en la pila y deshabilita el disparo al TRIAC. Después de eso mueve el dato que se encuentra en el SBUF y lo guarda en el acumulador. Primero se

compara si el valor del acumulador; es igual a 0FH, el cual indica a todos los dispositivos esclavos que se debe cambiar de modo automático a manual; y esto se lo realiza cambiando el valor del registro R4 de 00H a FFH, se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial. Si no es así ahora se verificará el valor del registro R4 para saber si se está en modo manual o automático. Después de esta comprobación se pasa a verificar si el valor guardado en el acumulador es igual a la dirección del dispositivo, si es así dependiendo del valor antes verificado del registro R4 se ejecuta la subrutina INT\_SER\_AUTOMATICO si el valor de R4 es igual a 00H, o la subrutina INT\_SER\_MANUAL si el valor de R4 es igual a FFH. En ambos casos antes de la llamada a la subrutina, se deshabilita el bit SM2 para que la información que viene en camino solo la reciba el dispositivo con el que coincidió la dirección. Pero si la dirección no coincide se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial. El diagrama de flujo de esta subrutina es el de la figura 3.27.

### 3.2.5 SUBROUTINA SERIAL\_AUTOMATICO

Esta subrutina es la encargada de recibir dos datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de temperatura del reboiler, es decir 11H y el registro R4 tiene el valor igual a 00H.

El primero de los datos recibidos es un comando, del cual dependerá en que variable se guardará el segundo dato. El segundo dato puede ser cualquiera de los siguientes:

- SET POINT de temperatura de fondos.
- SET POINT de temperatura de productos.
- Temperatura de disparo.
- Potencia de inicio del dispositivo de calentamiento.

Cuando el segundo dato es la potencia de inicio del dispositivo de calentamiento se realiza una transformación de porcentaje de potencia a ángulo de disparo por medio de la tabla T\_PORCENTAJE2GRADO. El diagrama de flujo de esta subrutina es el mostrado en la figura 3.28.

### 3.2.6 SUBROUTINA SERIAL\_MANUAL

Esta subrutina es la encargada de recibir dos datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de temperatura del reboiler, es decir 11H y el registro R4 tiene el valor igual a FFH.

El primero de los datos recibidos es un comando. Si el comando es 11H se espera recibir otro dato que será guardado en la variable ACTUADOR, pero previamente se hace una conversión de este valor que está en porcentaje de potencia a ángulo de disparo por medio de la tabla T\_PORCENTAJE2GRADO y después se retorna de la subrutina.

Si el comando es 0FH le indica al dispositivo que debe cambiar de modo manual a modo automático y esto se lo realiza colocando en el registro R4 el valor 00H. El diagrama de flujo de esta subrutina es el de la figura 3.29.

### 3.2.7 SUBROUTINA DALLAS\_RESET

Esta subrutina fue desarrollada en base al diagrama de tiempos mostrado en la figura 2.16, que es el diagrama de tiempos provisto por el fabricante para provocar un pulso de reset en el bus ONE WIRE del DS1821. Lo primero que hace el programa es colocar el bus en el estado 0L por un tiempo de 492us que son 12us más que el tiempo mínimo requerido. Después se muestrea el bus durante un tiempo de 71us esperando que el DS1821 ponga el bus en el estado 0L, conocido como pulso de presencia; sino es así, se enciende el bit etiquetado como ERROR. Una vez detectado el pulso de presencia se espera 250us; que son 10us más del tiempo máximo que le puede tomar al DS1821 liberar el bus. Si no se libera el bus

como se puede leer un 1L o 0L del DS1821. Antes de leer un dato del DS1821 primero hay que enviarle un comando de lectura con la subrutina DALLAS\_WRITE. Después se envía pulsos de 1L a 0L y luego a 1L con una duración un poco mayor a 1us. Esto le indica al DS1821 que puede enviar el bit, entonces 6us después el microprocesador AT89C2051 lee el valor que se encuentra en el bus ONE WIRE y lo mueve al CARRY para luego esperar 60us antes de enviar otro pulso para leer otro bit.

Cada vez que se realiza una lectura se realiza una rotación a la derecha con CARRY para ir almacenando en el acumulador uno a uno los bits que le envía el DS1821. Esta subrutina se muestra en la figura 3.32.

### 3.2.10 SUBROUTINA MEDIR

Esta subrutina hace uso de la subrutinas: DALLAS\_RESET, DALLAS\_WRITE y DALLAS\_READ, para tomar lecturas de cada uno de los DS1821 instalados en la torre. La medida tomada dentro del reboiler es almacenada en la variable MEDIDOR\_1, la medida tomada en la mitad de la torre es almacenada en la variable MEDIDOR\_2 y la medida tomada en la parte alta de la torre es almacenada en la variable MEDIDOR\_3.

El procedimiento para tomar una medida es primero ejecutar la subrutina DALLAS\_RESET. Después se carga el acumulador con el comando AAH, que es el comando que indica al DS1821 que debe enviar la lectura de temperatura de 8 bits hacia el microcontrolador, y se lo envía al DS1821 por medio de la subrutina DALLAS\_WRITE. Luego se ejecuta la subrutina DALLAS\_READ y el valor que se leyó se lo envía a las variables antes mencionadas.

Es importante aclarar que para cada DS1821 se tiene una subrutina de reset, lectura y escritura y se les distingue entre ellas por el índice en cada una. El diagrama de flujo de esta subrutina es el de la figura 3.33

### **3.2.11 SUBROUTINA CONTROL\_TEMPE**

Lo primero que hace esta subrutina es comparar el valor de temperatura de la parte más alta de la torre con la temperatura de disparo. Si el valor de temperatura medido en la parte más alta de la torre es mayor o igual a la temperatura de disparo, se carga a la variable ACTUADOR con el valor de la variable SER\_ACTUADOR y se apaga el LED conectado al pin etiquetado como ESTABLE. Caso contrario se verifica el valor de la variable CONTADOR, si es igual a 15 se ejecuta la acción de control y se complementa el pin etiquetado como ACCION el mismo que esta conectado a un LED; sino se sale de la subrutina.

La acción de control consiste primero en comparar el valor de temperatura medido en la parte más alta de la torre con el SET POINT de temperatura de productos. Si la lectura medida es mayor al SET POINT de temperatura de productos, se hace un llamado a la subrutina INCRE\_ANGULO, la cual incrementa en una unidad la variable ACTUADOR y por ende reduce la potencia del dispositivo de calentamiento. Caso contrario ahora pasa a comparar la lectura de temperatura dentro del reboiler, con el SET POINT de temperatura de producto. Si la temperatura medida dentro del reboiler es menor a la del SET POINT de temperatura de productos, se realiza un llamado a la subrutina DECRE\_ANGULO, la cual decrementa en una unidad la variable ACTUADOR y por ende aumenta la potencia del dispositivo de calentamiento. El diagrama de flujo de esta subrutina es el de la figura 3.34.

### **3.2.12 SUBROUTINA INCRE\_ANGULO**

Esta subrutina primero compara si el valor de la variable ACTUADOR es mayor o igual a la variable SER\_ACTUADOR. Si es así retorna de la subrutina, pero caso contrario incrementa el acumulador en una unidad y retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.35.

### 3.2.13 SUBROUTINA DECRE\_ANGULO

Esta subrutina compara si el valor de la variable ACTUADOR es igual a cero, si es así retorna de la subrutina; caso contrario se decrementa la variable ACTUADOR en una unidad. El diagrama de flujo de esta subrutina es el de la figura 3.36.

### 3.2.14 SUBROUTINA ACT\_ENABLE

Esta subrutina toma el valor del pin etiquetado como ENABLE y lo envía al CARRY para después complementar el CARRY y enviarlo al pin etiquetado como ERR\_ENABLE el mismo que esta conectado un LED. La función de este LED es la indicar si existe o no suficiente nivel de líquido dentro del reboiler.

Para finalizar se vuelve a tomar el valor del pin etiquetado como ENABLE y se lo envía al CARRY, después de esto se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.37.

### 3.2.15 SUBROUTINA ENVIAR\_DATOS

Lo primero que hace esta subrutina es deshabilitar todas las interrupciones después envía el comando EEH al puerto serial por medio de la subrutina ENVIAR, después envía las tres variables en donde se almacena las medidas de temperatura que son: MEDIDOR\_1, MEDIDOR\_2 y MEDIDOR\_3. Luego se envía el valor actual de la variable ACTUADOR, pero antes de enviar este dato se le realiza una transformación de ángulo de disparo a porcentaje de potencia por medio de la tabla T\_GRADO2PORCENTAJE mostrada en la tabla 4.6. Una vez hecha esta transformación se envía el dato. Finalmente se habilitan las interrupciones y se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.38.

### **3.2.16 SUBROUTINA ENVIAR**

Antes de llamar a esta subrutina hay que cargar en el acumulador el dato que va a ser enviado. En la subrutina ENVIAR se mueve al SBUF lo del acumulador y con eso empieza la transmisión del dato. Cuando se haya terminado de enviar se borra la bandera de transmisión y se retorna de la subrutina. El diagrama de flujo de esta subrutina se muestra en la figura 3.39.

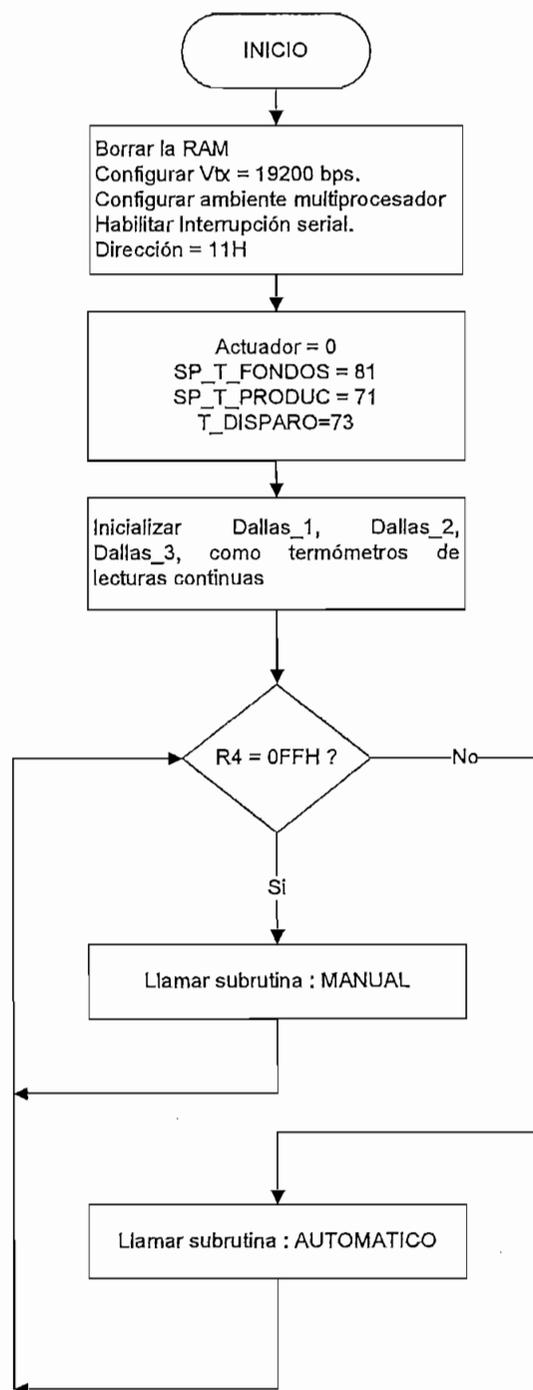
**PROGRAMA PRINCIPAL**

Figura 3.24

## SUBROUTINA AUTOMATICO

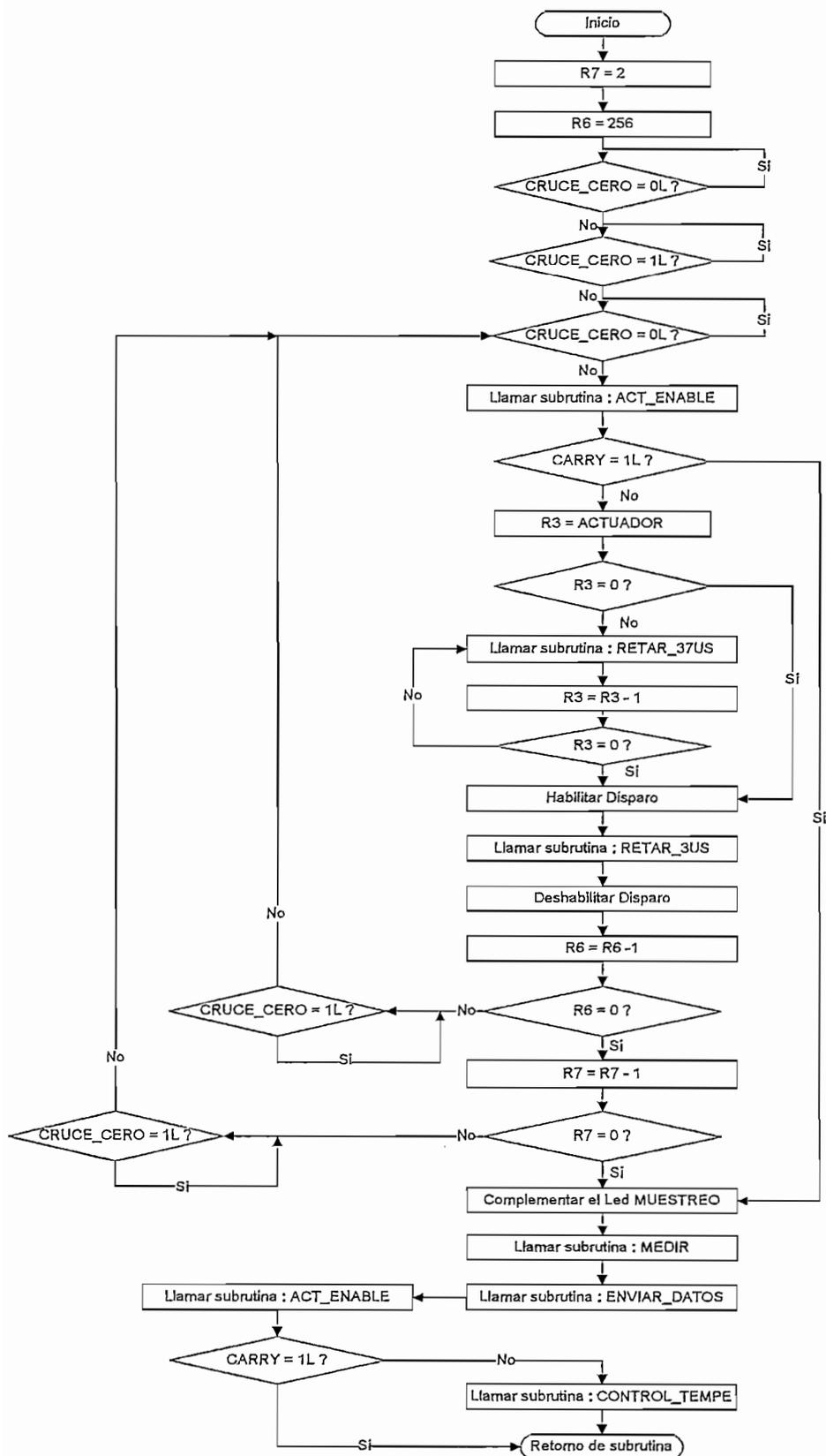


Figura 3.25

## SUBROUTINA MANUAL

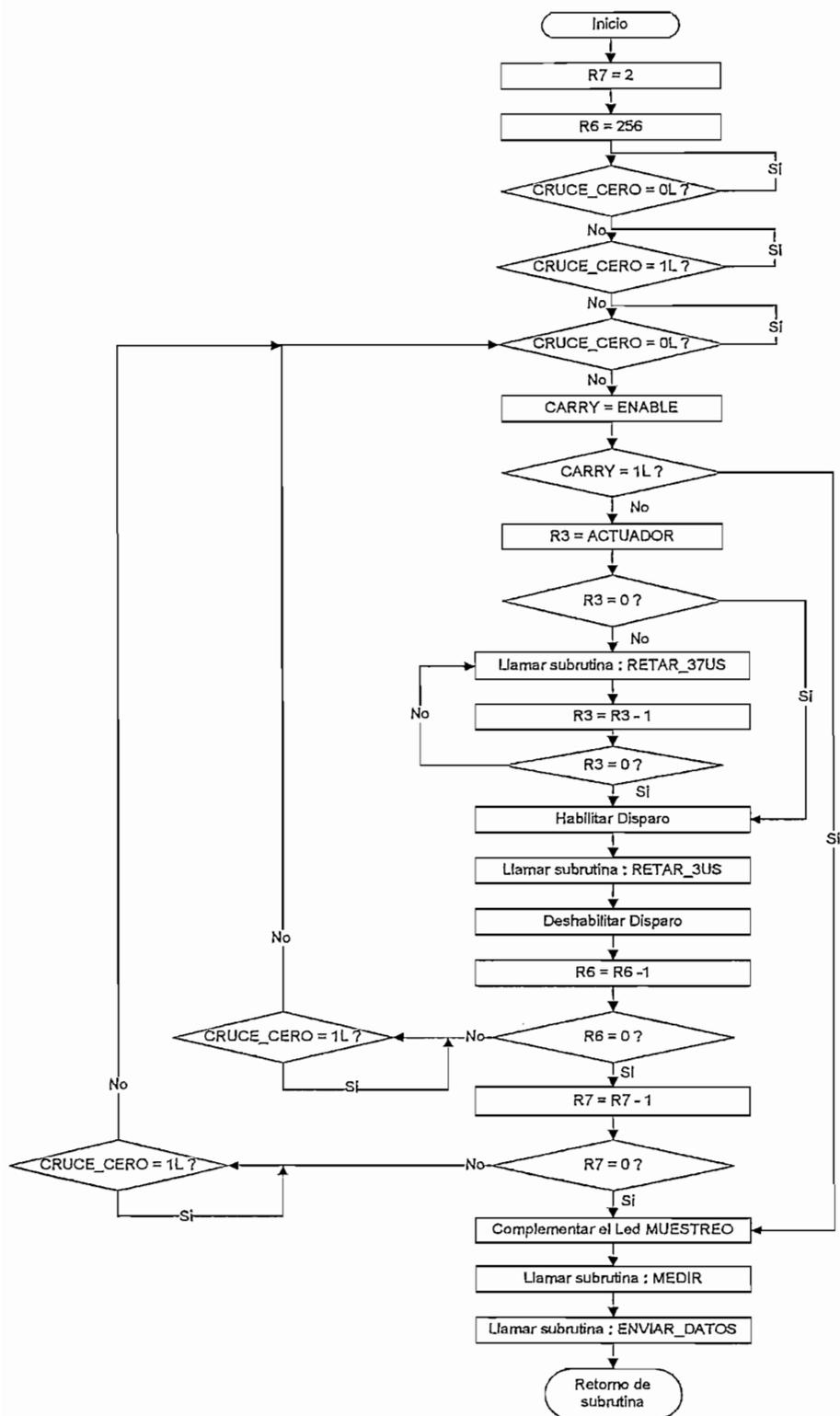


Figura 3.26

## SUBROUTINA INT\_SERIAL

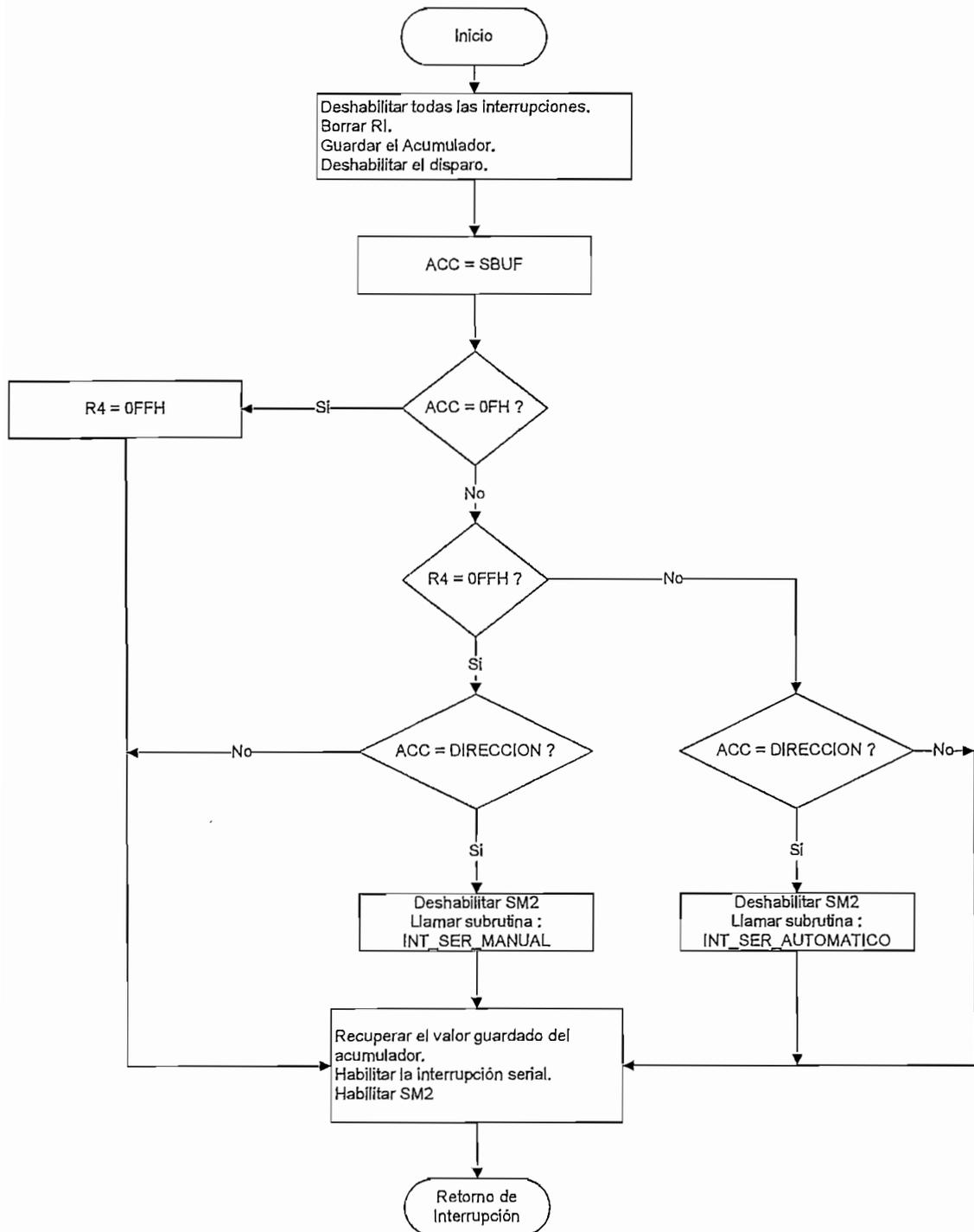


Figura 3.27

## SUBROUTINA SERIAL\_AUTOMATICO

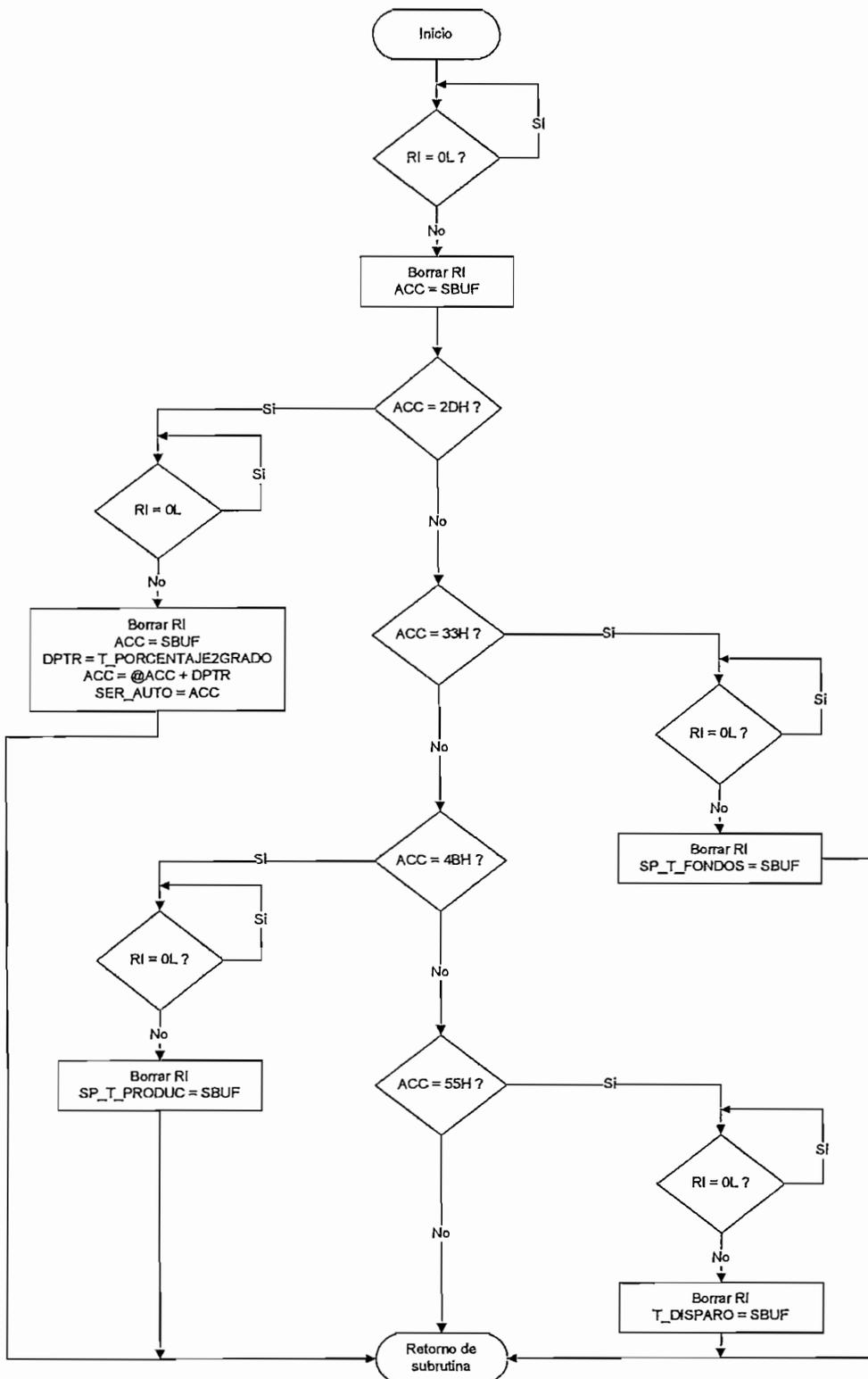


Figura 3.28

## SUBROUTINA SERIAL\_MANUAL

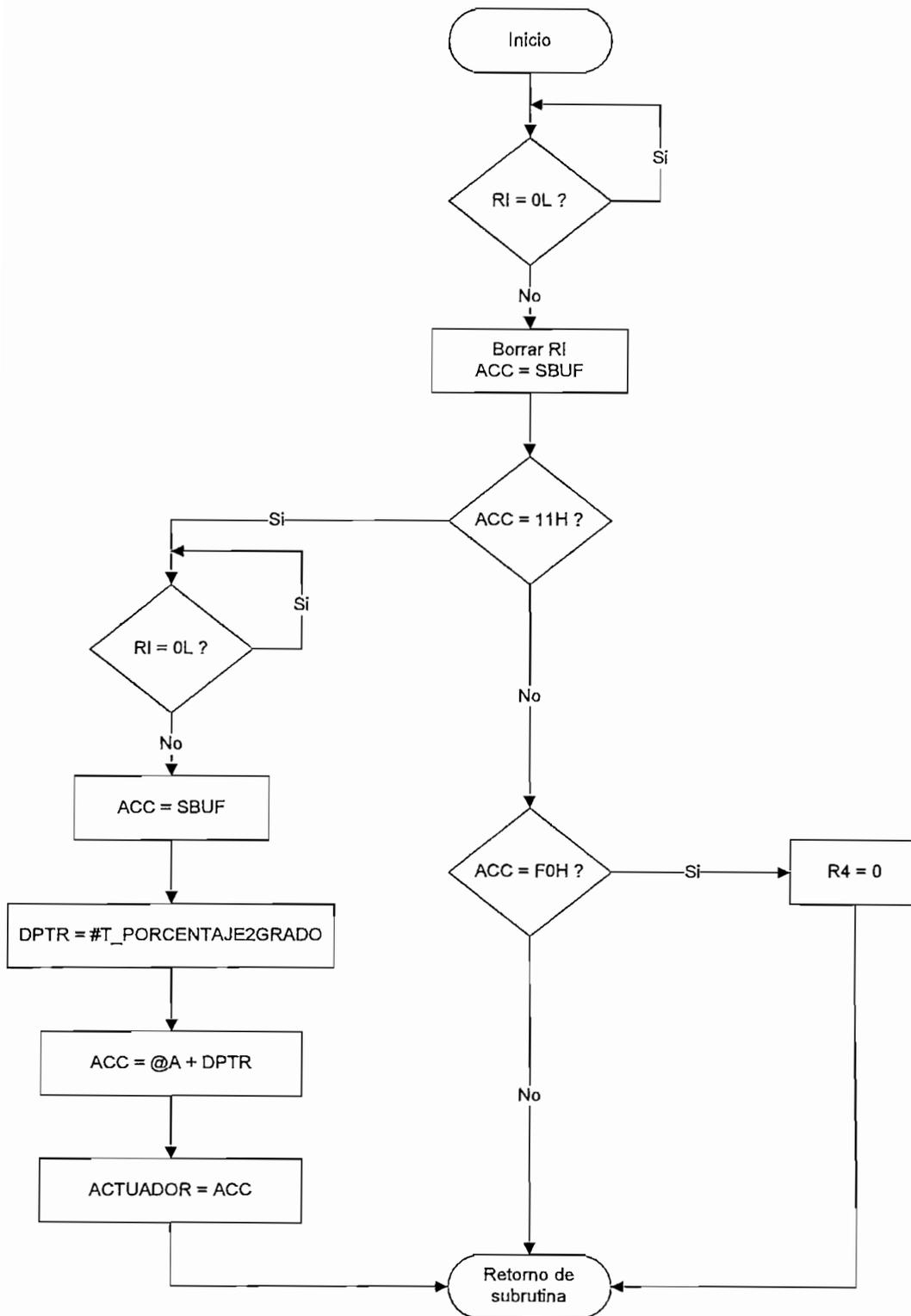


Figura 3.29

## SUBROUTINA DALLAS\_RESET

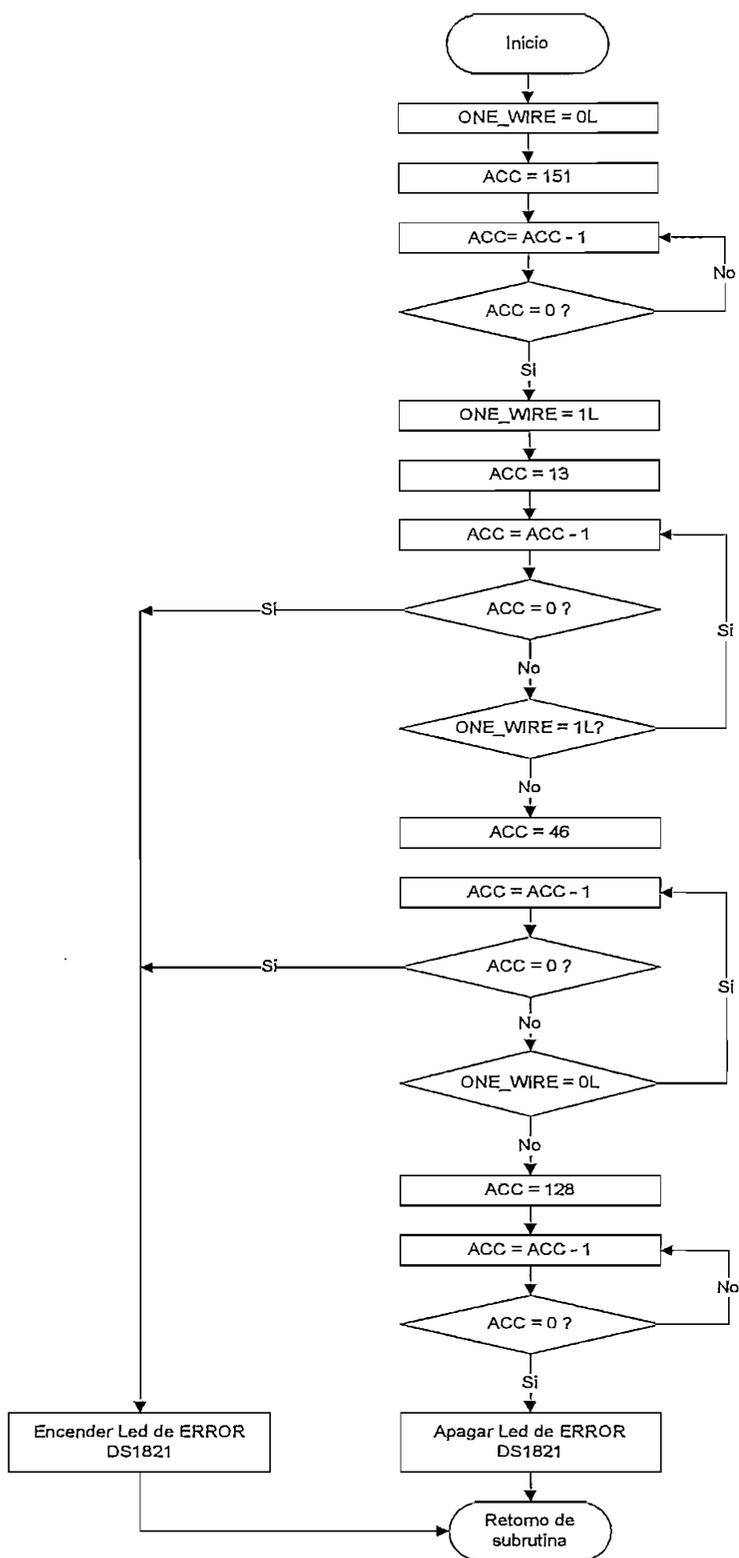


Figura 3.30

## SUBROUTINA DALLAS\_WRITE

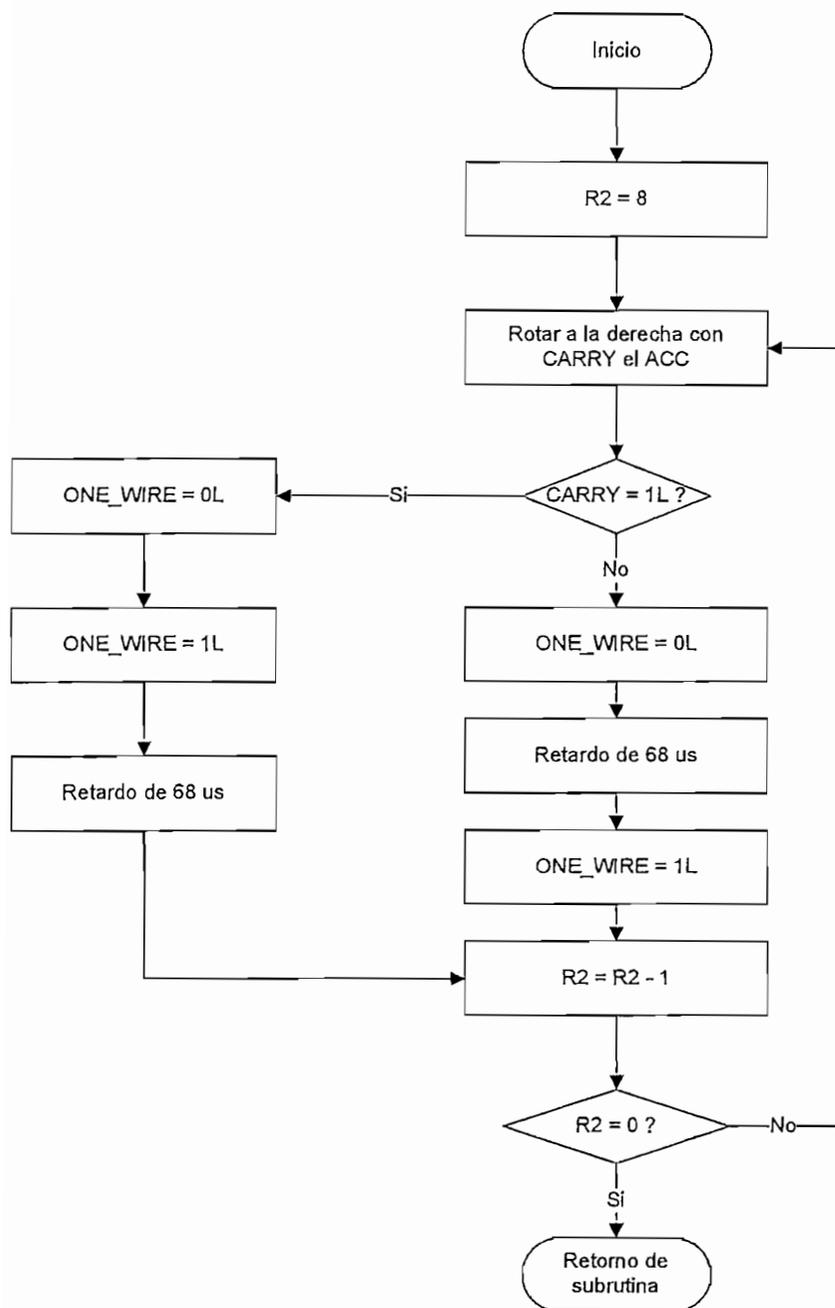


Figura 3.31

## SUBROUTINA DALLAS\_READ

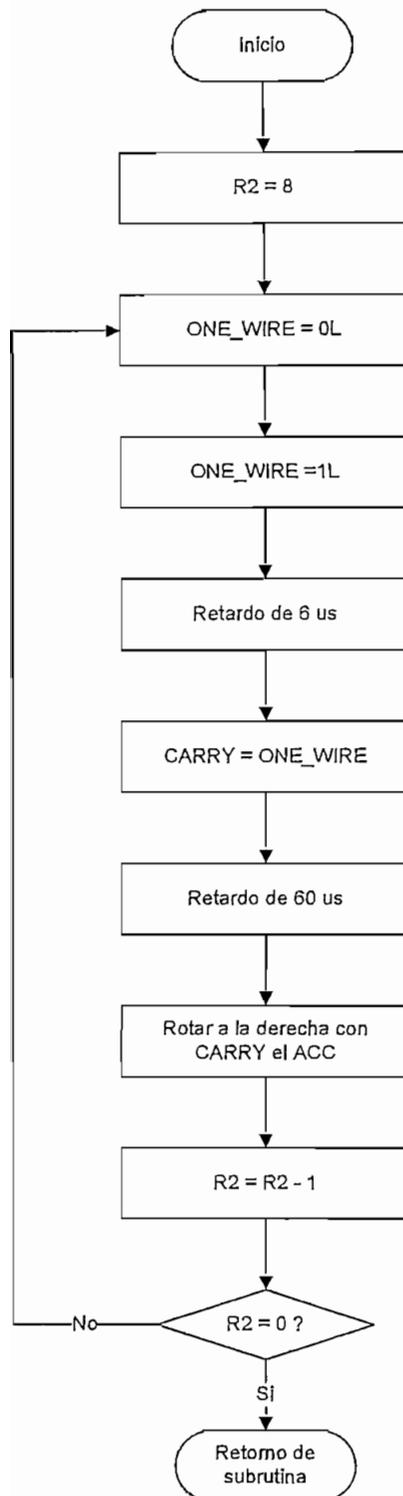


Figura 3.32

## SUBROUTINA MEDIR

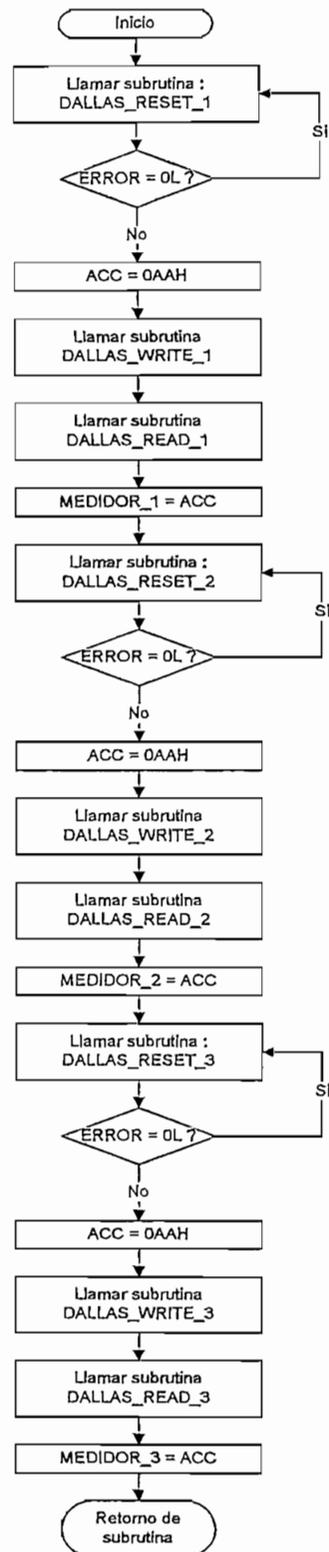


Figura 3.33

## SUBROUTINA CONTROL\_TEMPE

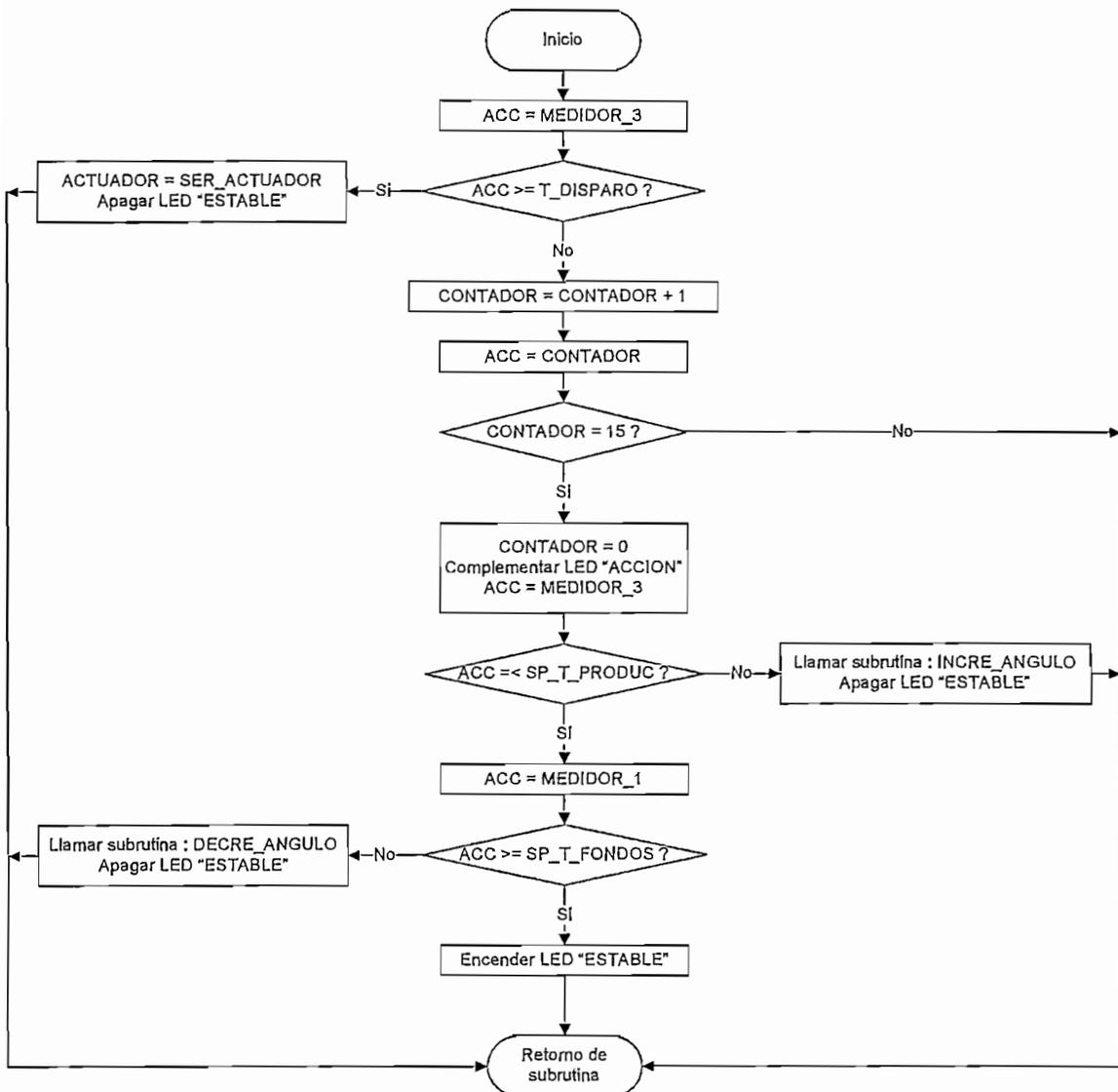


Figura 3.34

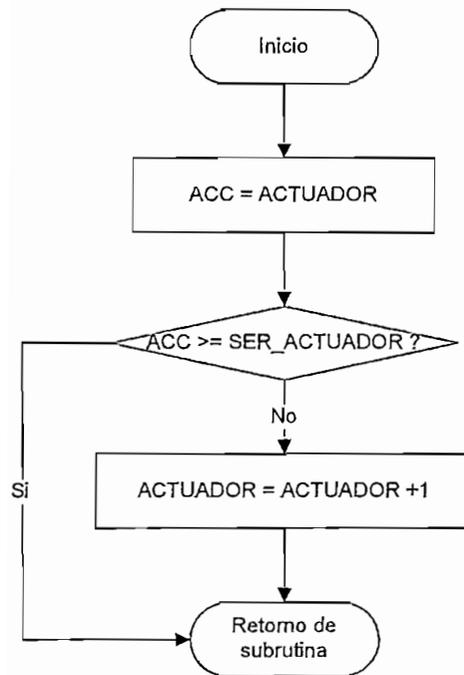
**SUBROUTINA INCRE\_ANGULO**

Figura 3.35

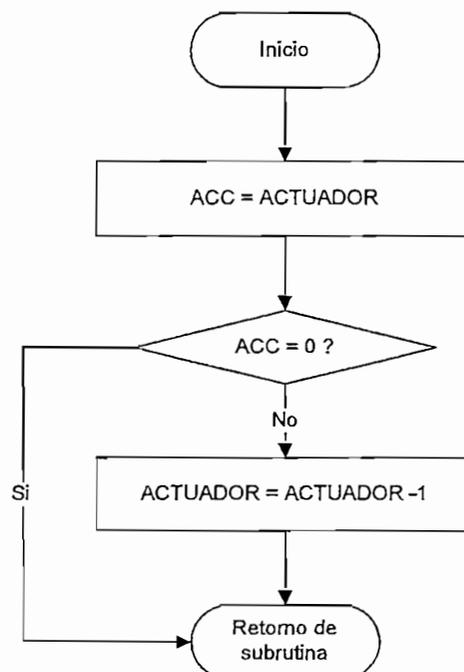
**SUBROUTINA DECRE\_ANGULO**

Figura 3.36

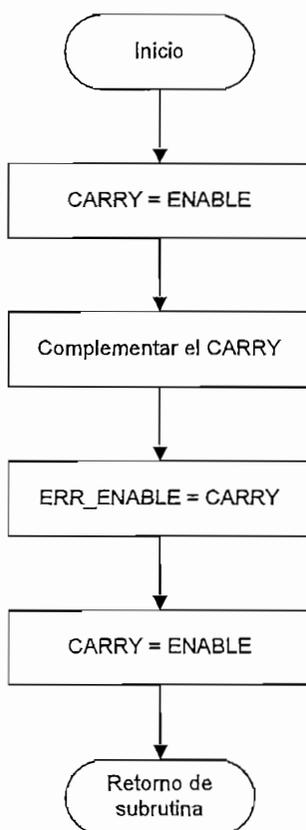
**SUBROUTINA ACT\_ENABLE**

Figura 3.37

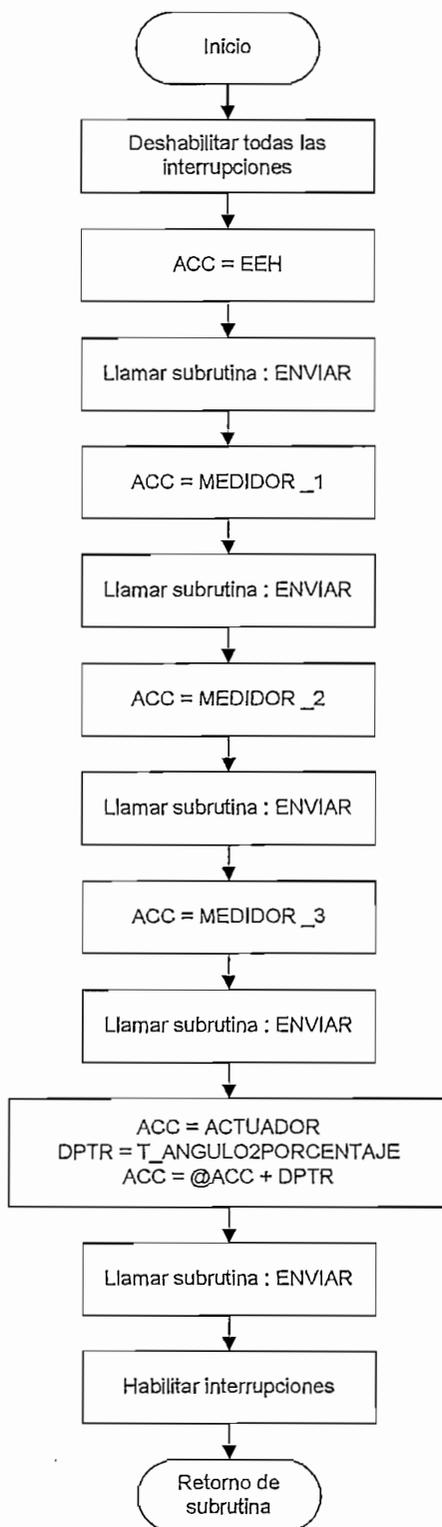
**SUBROUTINA ENVIAR\_DATOS**

Figura 3.38

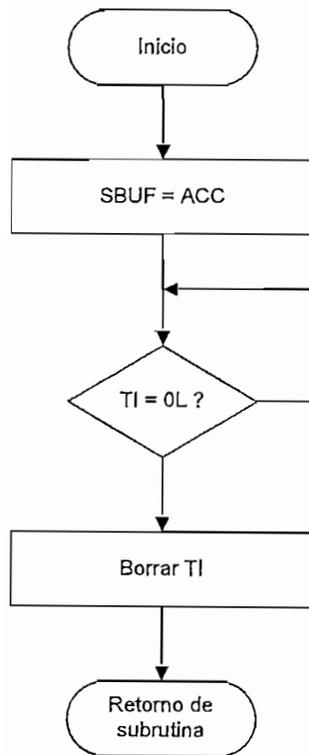
**SUBROUTINA ENVIAR**

Figura 3.39

### **3.3 SUBROUTINAS PARA EL CONTROL DEL NIVEL DEL REBOILER**

El programa en lenguaje ensamblador que se ejecuta en el microcontrolador AT89C51 para manejar el control de nivel del reboiler se encuentra en el anexo 3. Cabe recalcar que el análisis de los siguientes diagramas de flujo deber ser realizado conjuntamente con el listado de etiquetas que se incluye en el anexo 3.

#### **3.3.1 PROGRAMA PRINCIPAL**

El programa principal del control de nivel del reboiler en su inicio se encarga de borrar la memoria RAM, así como también configurar los parámetros de la comunicación serial a 19200bps, 9 bits de datos y ambiente multiprocesador, y también asigna la dirección 11H a este dispositivo. Se inicializan variables como NIVEL, MAX\_ACT, MIN\_ACTB y SECUENCIA. De igual forma se deshabilita el pin etiquetado como ENABLE, el mismo que envía una señal al control de temperatura del reboiler para que ejecute el disparo del TRIAC. Después de esto se llama a la subrutina CALIBRAR la misma que se encarga de cerrar por completo la válvula de fondo y poner a cero los contadores que cuentan el número de pasos que gira el motor paso a paso. Luego se inicializan las variables AUX\_MAX\_ACT y AUX\_MIN\_ACT. De ahí en adelante se entra en un bucle repetitivo; en el cual dependiendo del valor de R4 se ejecutará la subrutina MANUAL o la subrutina AUTOMATICO. El diagrama de flujo del programa principal, del cual se derivan el resto de subrutinas excepto las de interrupción, se muestra en la figura 3.40.

#### **3.3.2 SUBROUTINA AUTOMÁTICO**

La subrutina AUTOMATICO en un inicio se encarga de llamar a las subrutinas ACT\_SENSORES y SEN\_NIVEL. Después se compara el valor de la variable BANDERA. Si BANDERA es igual a cero implica que no se ha alcanzado el SET POINT de temperatura de fondos por lo tanto se cierra por completo la válvula y se coloca la BANDERA\_1 con un valor igual a 00H. Pero si el valor de BANDERA no

es igual a cero, ahora se pasa a verificar el valor de BANDERA\_1. Si BANDERA es igual a cero se ejecuta una subrutina que abre la válvula a la mínima cantidad que el usuario ingreso por teclado para obtener fondos, realizado esto se cambia el valor de BANDERA\_1 a FFH para que no se vuelva a ejecutar esta subrutina a menos que se cierre otra vez por completo la válvula y cambie el valor de BANDERA\_1. Pero si el valor de BANDERA\_1 no es cero, entonces se coloca en BANDERA\_1 el número FFH y se guarda en el acumulador lo de la localidad de memoria etiquetada como NIVEL. Si el valor del acumulador es menor o igual a 6 se llama a la subrutina CERRAR, pero si el valor del acumulador es mayor a 6 entonces se llama a la subrutina ABRIR. El diagrama de flujo de esta subrutina es el mostrado en la figura 3.41.

### 3.3.3 SUBROUTINA MANUAL

Lo primero que realiza esta subrutina es llamar a la subrutina ACT\_SENSORES y la subrutina SEN\_NIVEL. Una vez hecho esto cambia los valores de MAX\_ACT y MIN\_ACT a 60 y 0 respectivamente. Después se mueve al acumulador lo de la variable COMANDO, dependiendo de que valor tenga el acumulador se puede ejecutar una de las siguientes opciones.

- Abrir la válvula 10 pasos.
- Cerrar la válvula 10 pasos.
- Abrir por completo la válvula.
- Cerrar por completo la válvula.
- No hacer nada.

El diagrama de flujo de esta subrutina es el de la figura 3.42.

### 3.3.4 SUBROUTINA INT\_SERIAL

Al momento que se produce una interrupción serial debido a que el bit RB8 es 1L en el modo ambiente multiprocesador, esta subrutina lo primero que hace es deshabilitar todas las interrupciones, borra la bandera de recepción RI y guarda el

acumulador en la pila. Después de eso mueve el dato que se encuentra en el SBUF y se lo guarda en el acumulador. Primero se compara si el valor del acumulador; es igual a 0FH, el cual indica a todos los dispositivos esclavos que se debe cambiar de modo automático a manual; y esto se lo realiza cambiando el valor del registro R4 de 00H a FFH y también guarda el valor de las variables MAX\_ACT y MIN\_ACT en las variables AUX\_MAX\_ACT y AUX\_MIN\_ACT respectivamente y se coloca en la variable COMANDO el valor 55H. Una vez realizado esto se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial.

Si el valor del acumulador no es igual a 0FH ahora se verificará el valor del registro R4 para saber si se está en modo manual o automático. Después de esta comprobación se pasa a verificar si el valor guardado en el acumulador es igual a la dirección del dispositivo, si es así dependiendo del valor antes verificado del registro R4 se ejecuta la subrutina INT\_SER\_AUTOMATICO si el valor de R4 es igual a 00H, o la subrutina INT\_SER\_MANUAL si el valor de R4 es igual a FFH. En ambos casos antes de la llamada a la subrutina, se deshabilita el bit SM2 para que la información que viene en camino solo la reciba el dispositivo con el que coincidió la dirección. Pero si la dirección no coincide se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial. El diagrama de flujo de esta subrutina es el de la figura 3.43.

### 3.3.5 SUBROUTINA SERIAL\_AUTOMÁTICO

Esta subrutina es la encargada de recibir dos datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de nivel del reboiler, es decir 33H y el registro R4 tiene el valor igual a 00H.

Después de lo anterior se pueden recibir uno o dos datos. Cuando se recibe dos datos el primero es un comando del cual dependerá en que variable se

almacenará el segundo dato. El segundo dato puede ser cualquiera de los siguientes:

- Máximo valor del actuador.
- Mínimo valor del actuador.

Si solo se recibe un dato, este será un comando que cambie el valor de la localidad de memoria etiqueta como BANDERA, la cual indica si se alcanzado o no el SET POINT de temperatura de fondos. El diagrama de flujo de esta subrutina es el de la figura 3.44.

### 3.3.6 SUBROUTINA SERIAL\_MANUAL

Esta subrutina es la encargada de recibir dos datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de nivel del reboiler, es decir 33H y el registro R4 tiene el valor igual a FFH.

El primero de los datos recibidos es un comando. Si el comando es 69H se espera recibir otro dato que será guardado en la variable COMANDO, cuyo valor determina como funcionará la subrutina MANUAL.

Si el comando es 0FH le indica al dispositivo que debe cambiar de modo manual a modo automático y esto se lo realiza colocando en el registro R4 el valor 00H y restaurando los valores de MAX\_ACT y MIN\_ACT que fueron guardados previamente en las variables AUX\_MAX\_ACT y AUX\_MIN\_ACT. El diagrama de flujo de esta subrutina es el de la figura 3.45.

### 3.3.7 SUBROUTINA CALIBRAR

Esta subrutina primero comienza con poner la velocidad del motor a 2 pasos por segundo y abrir 20 pasos la válvula de fondos. Después se cambia la velocidad a 50 pasos por segundo y se llama a la subrutina CERRAR hasta que se detecte el

SENSOR\_1. Una vez que se haya detectado el SENSOR\_1, se muestrea SENSOR\_2 durante 20 pasos. Si dentro de esos 20 pasos no se detectó el SENSOR\_2 implica que la válvula está cerrada por completo. Entonces se abre la válvula 20 pasos cambiando la velocidad a 2 pasos por segundo y después se llama a la subrutina CERRAR hasta detectar al SENSOR\_1. Ya detectado el SENSOR\_1, se colocan en las variables CONTADOR\_1 y CONTADOR\_2 el número 0 y se retorna de la subrutina.

Por otro lado si cerrando la válvula se detectó el SENSOR\_1 y si dentro de los 20 pasos siguientes se detecta el SENSOR\_2, se sigue cerrando la válvula hasta detectar de nuevo al SENSOR\_1. El diagrama de esta subrutina es el de la figura 3.46.

### 3.3.8 SUBROUTINA SEN\_NIVEL

Esta subrutina muestrea pin a pin los pines del pósito 0. Si alguno de ellos cambia de estado de 1L a 0L se ejecuta un retardo de 10ms para eliminar los rebotes que pueda existir o ruido eléctrico que pueda provocar una falsa entrada. Dependiendo que pin del pósito se encontró en un estado 0L esta subrutina asigna a la variable NIVEL un número comprendido entre 1 y 8; y se envía hacia el pósito 1, un código para poder visualizar en el DISPLAY de 7 segmentos en que nivel se encuentra el líquido dentro del reboiler. La tabla 4.4 hace una relación del nivel con el valor código que hay que enviar hacia el pósito 1. El diagrama de flujo de esta subrutina se muestra en la figura 3.47.

### 3.3.9 ACT\_SENSORES

Esta subrutina sirve para encender los LEDs conectados a los pines etiquetados como LED\_SENSOR\_1 y LED\_SENSOR\_2; dependiendo del estado en que se encuentran los pines de los optoacopladores etiquetados como SENSOR\_1 y SENSOR\_2 respectivamente. Lo que se hace es mover el valor de alguna de las entradas al CARRY, y después se envía lo del CARRY a una de las salidas de los LEDs. El diagrama de flujo de esta subrutina es el de la figura 3.48

### 3.3.10 SUBROUTINA ABRIR

Esta subrutina primero mueve el valor de `CONTADOR_2`, al acumulador para poder compararlo con el valor de `MAX_ACT`. Si el valor de `CONTADOR_2` es mayor o igual al valor límite máximo del actuador, se retorna de la subrutina. Caso contrario se mueve al acumulador el valor de la variable `SECUENCIA`, y se la hace rotar a la izquierda el acumulador, porque en ese sentido la válvula se abre. Después este valor es nuevamente guardado en la variable `SECUENCIA`, para ser utilizado en una subsiguiente vez. Después se hacen algunas operaciones booleanas para enviar sólo los 4 bits menos significativos del acumulador, hacia los 4 bits menos significativos del pórtico 2; sin alterar el estado de las demás líneas del pórtico 2. Como se hace una escritura en el pórtico 2 se necesita nuevamente habilitar las entradas del pórtico 2 que son los pines etiquetados como `SENSOR_1` Y `SENSOR_2`. Luego se genera un retardo que depende del valor que tengan las variables `AUX_VAR1`, `AUX_VAR2` y `AUX_VAR3`; pero en general existen 2 retardos: de 20ms y de 500ms. Después de terminado el retardo se hace un llamado a la subrutina `INCREMENTAR` y se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.49.

### 3.3.11 SUBROUTINA CERRAR

Lo primero que se hace en esta subrutina es copiar el valor de `CONTADOR_1` en el acumulador para compararlo si es igual a cero. Si no es igual a cero la válvula se cierra un paso. Pero si el valor es igual a cero se pasa a verificar el valor de `CONTADOR_2`. Si el valor de `CONTADOR_2` es igual o menor al mínimo valor del actuador entonces se hace un retorno de subrutina; pero por lo contrario si es mayor se procede a cerrar la válvula un paso de la siguiente forma. Se mueve al acumulador el valor de la variable `SECUENCIA` y se hace rotar al acumulador hacia la derecha. Después el valor de acumulador es enviado a la variable `SECUENCIA`, para ser utilizado en una próxima vez. Luego se hacen algunas operaciones booleanas para enviar los 4 bits menos significativos del acumulador a los cuatro bits más significativos del pórtico 2, sin alterar el resto de pines del pórtico 2. Una vez realizado esto se habilitan las entradas del pórtico 2 y se llama

a una subrutina de retardo que puede ser de 2 tiempos: 20ms o 500ms. Una vez hecho esto se llama a la subrutina DECREMENTAR, para después entrar en una parte del programa en donde se compara si CONTADOR\_1 y CONTADOR\_2 son igual a cero y si el bit SENSOR\_1 es igual a cero para de esa forma encender o no el LED conectado al bit etiquetado como ERROR. Que los contadores CONTADOR\_1 y CONTADOR\_2 sean cero y que el bit SENSOR\_1 no sea cero; implica que el motor se está atascando en algún lado y toca centrarlo de nuevo; por ende se enciende el LED conectado al pin etiquetado como ERROR. El diagrama de flujo de esta subrutina es el mostrado en la figura 3.50.

### **3.3.12 SUBROUTINA INCREMENTAR**

La subrutina INCREMENTAR sirve para incrementar la variable CONTADOR\_1 en una unidad cada vez que se ejecuta la misma. Pero por cada 10 incrementos de la variable CONTADOR\_1, se incrementa una vez la variable CONTADOR\_2 y cuando esto pasa se coloca en la variable CONTADOR\_1, el valor 0. El diagrama de flujo de esta subrutina es el de la figura 3.51.

### **3.3.13 SUBROUTINA DECREMENTAR**

La subrutina DECREMENTAR sirve para decrementar la variable CONTADOR\_1 en una unidad cada vez que se ejecuta la misma. Pero por cada 10 decrementos de la variable CONTADOR\_1, se decrementa en una unidad la variable CONTADOR\_2 y cuando esto pasa se coloca en la variable CONTADOR\_1 el valor de 9. El diagrama de flujo de esta subrutina 3.52.

### **3.3.14 SUBROUTINA ACT\_MOTOR**

Esta subrutina sirve para abrir la válvula de fondos al valor de la variable MIN\_ACT. Lo primero que se hace es mover al acumulador el valor de la variable CONTADOR\_2 y se lo compara con el valor de la variable MIN\_ACT. Si el valor de CONTADOR\_2 es igual al valor de MIN\_ACT se sale de la subrutina. Pero si el valor del CONTADOR\_2 es mayor que el valor de MIN\_ACT se llama a la

subrutina CERRAR el número de veces que sea necesario hasta que la variable CONTADOR\_2 alcance el valor de la variable MIN\_ACT. Si por el contrario el valor de CONTADOR\_2 es menor al valor de MIN\_ACT, se llama a la subrutina ABRIR el número de veces que sea necesario hasta que el valor de la variable CONTADOR, alcance el valor de la variable MIN\_ACT. El diagrama de flujo de esta subrutina es el de la figura 3.53

## PROGRAMA PRINCIPAL

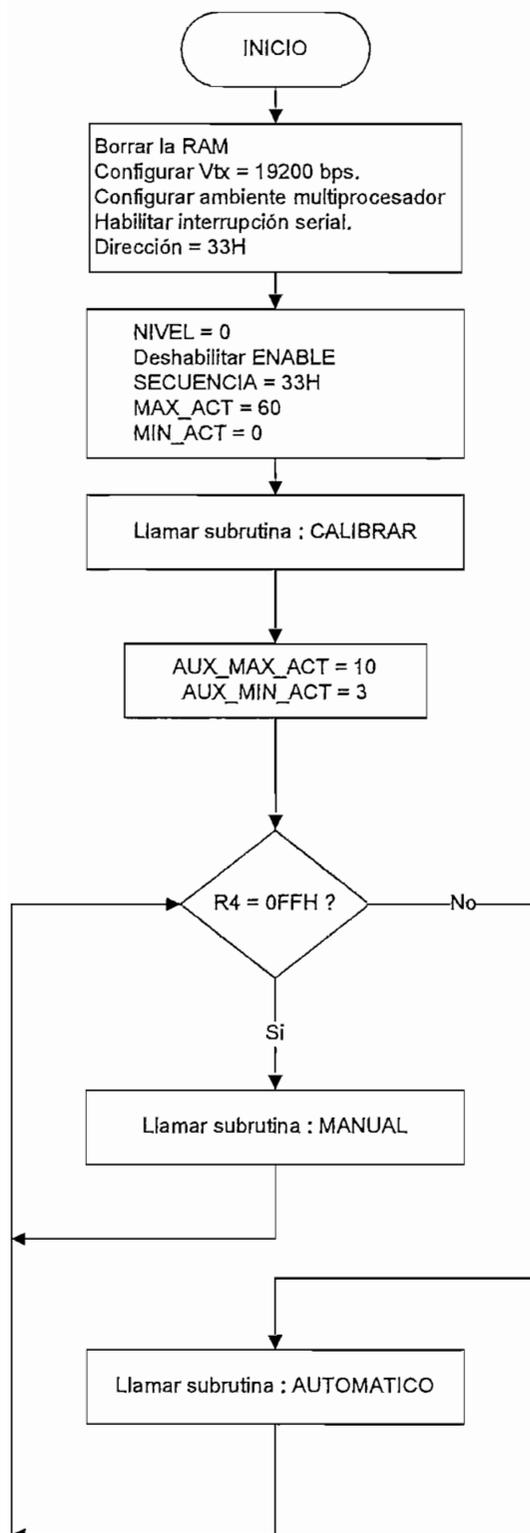


Figura 3.40

## SUBRUTINA AUTOMATICO

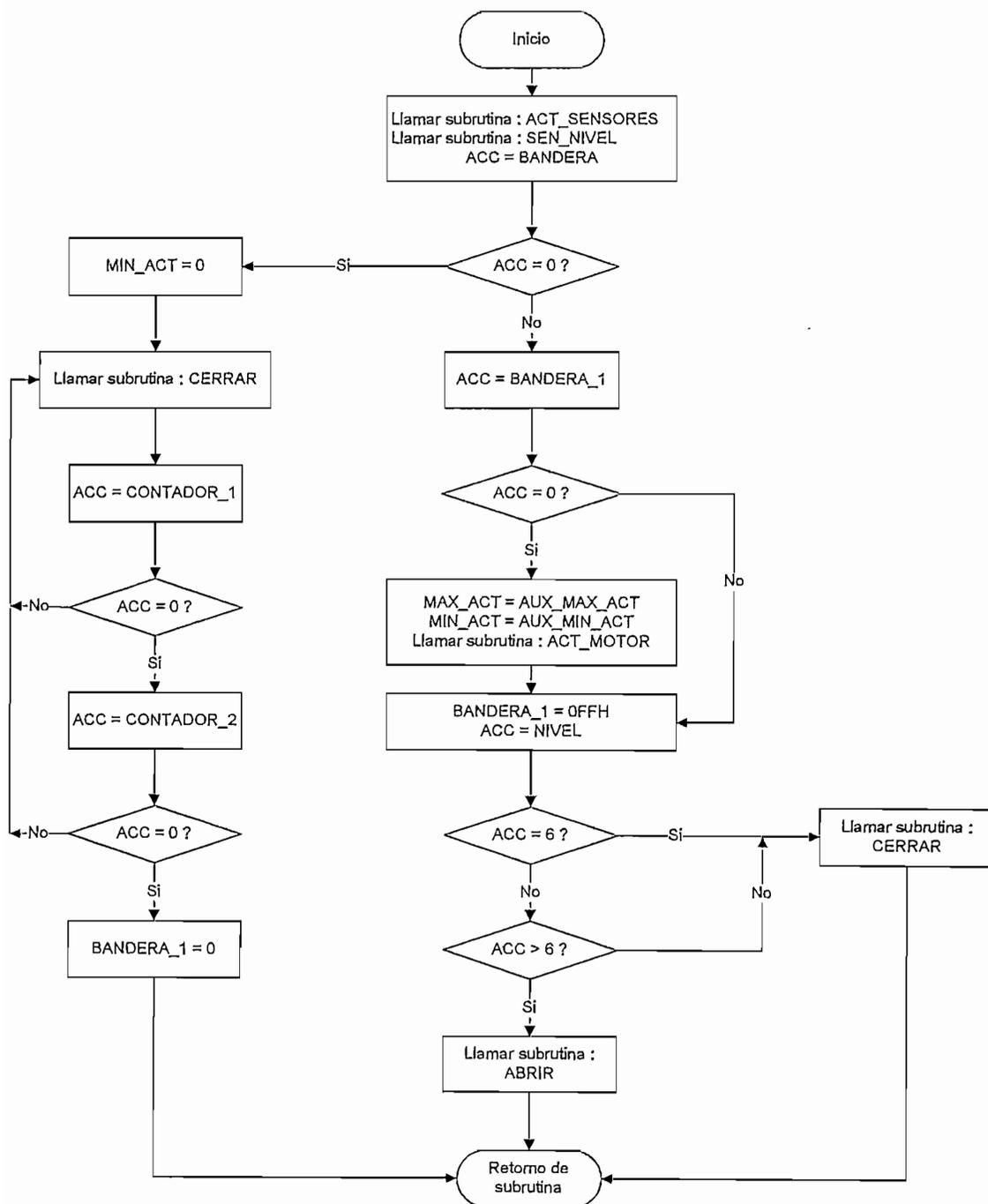


Figura 3.41

### SUBROUTINA MANUAL

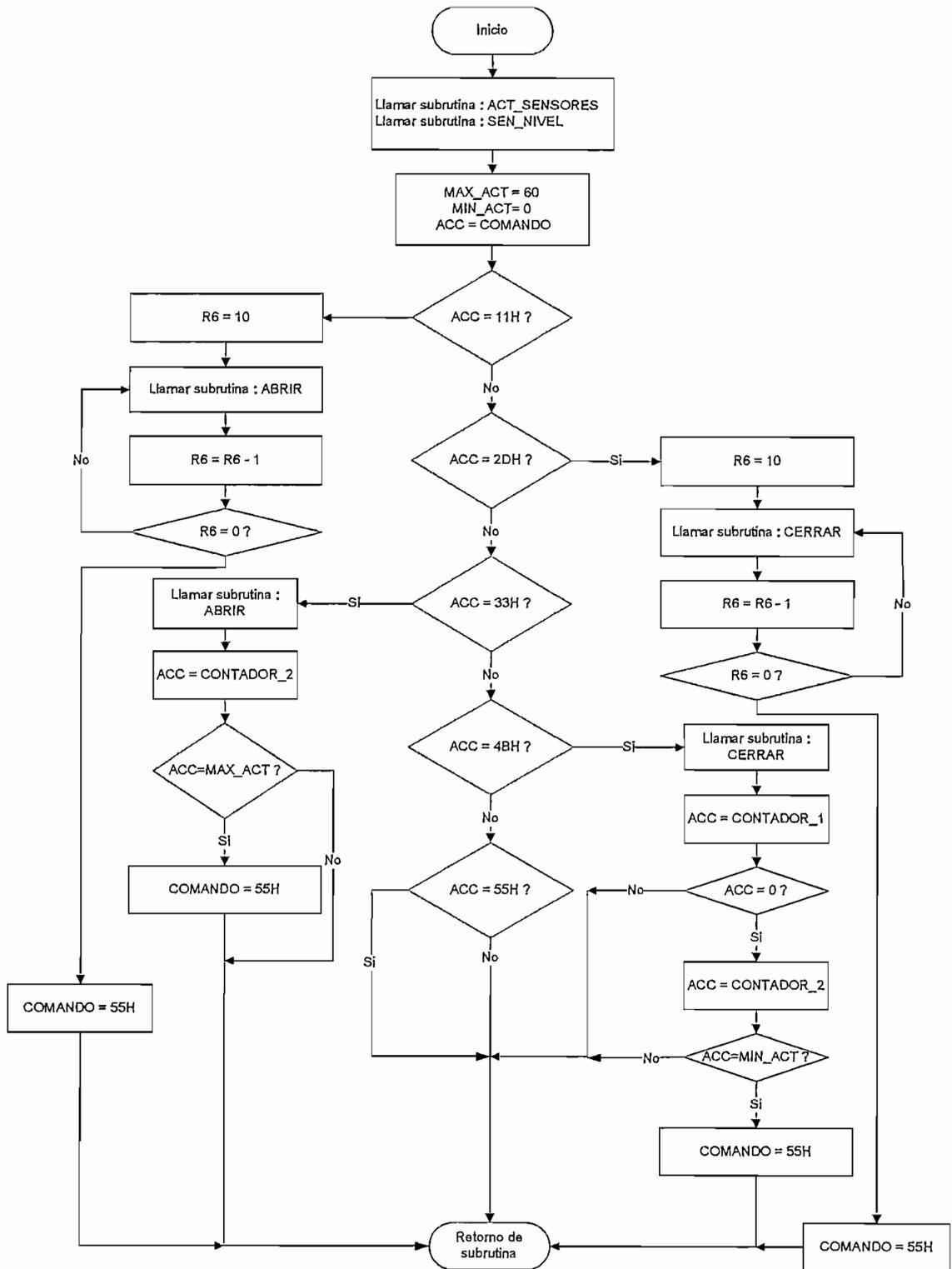


Figura 3.42

## SUBROUTINA INT\_SERIAL

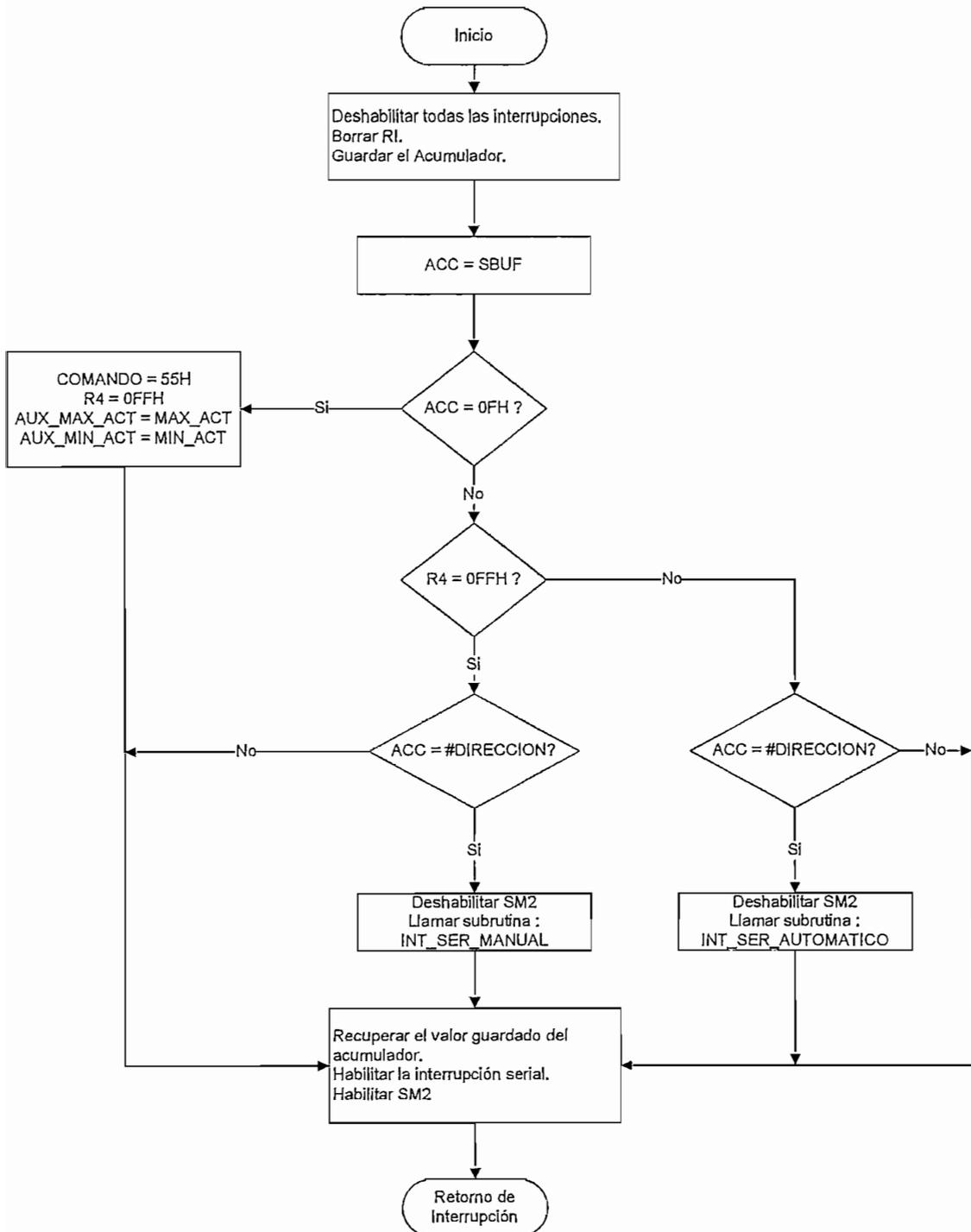


Figura 3.43

## SUBROUTINA SERIAL\_AUTOMATICO

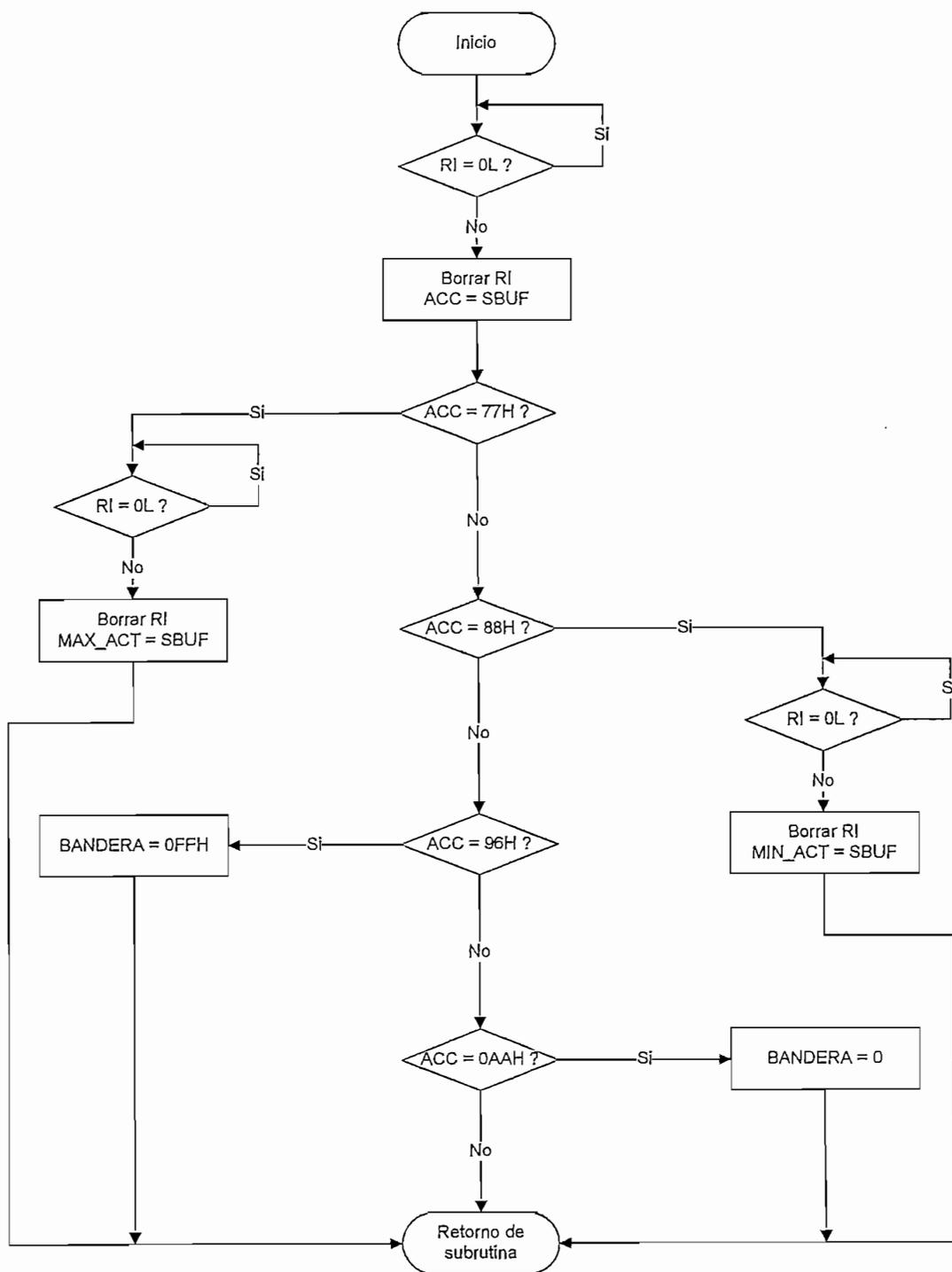


Figura 3.44

## SUBROUTINA SERIAL\_MANUAL

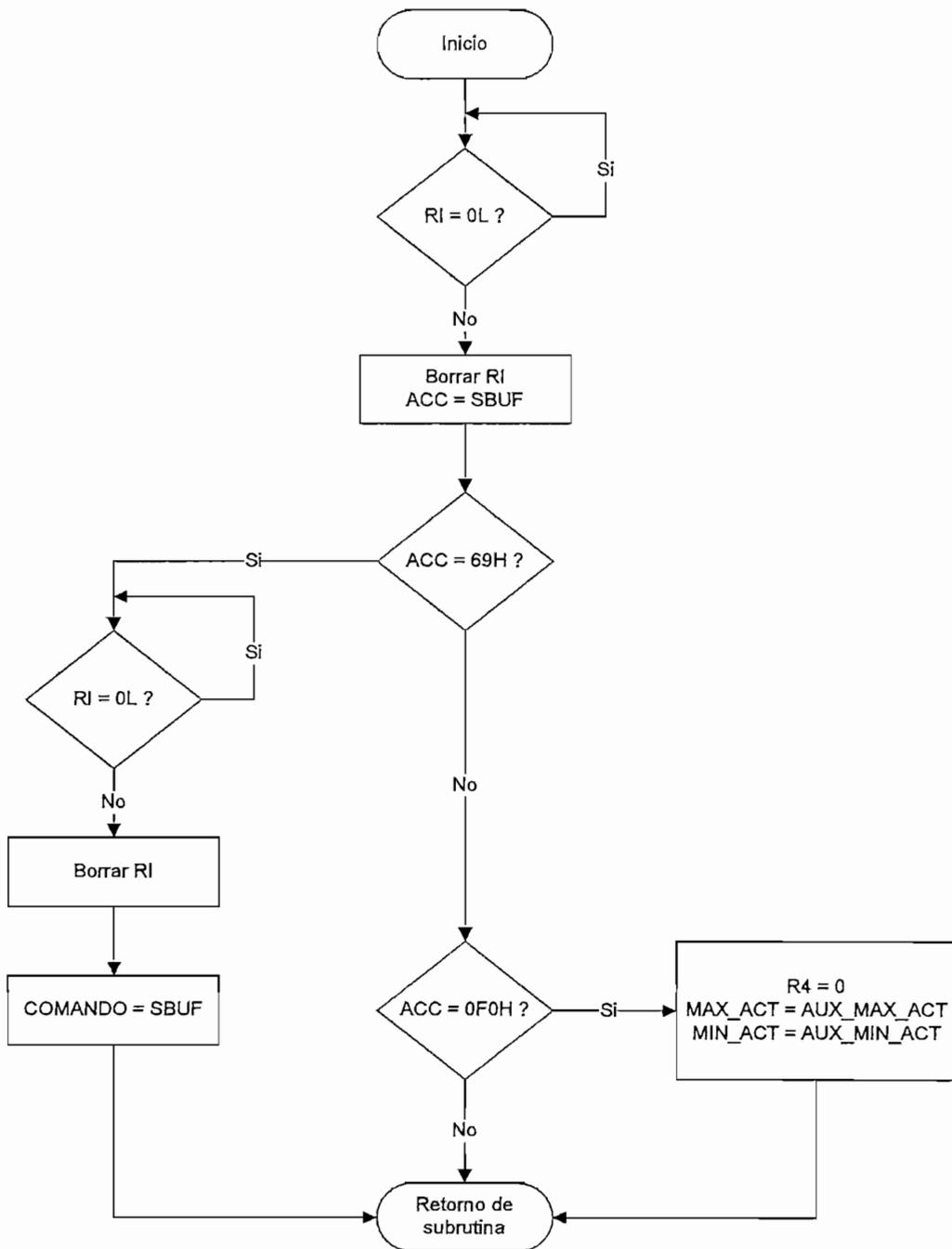


Figura 3.45

## SUBROUTINA CALIBRAR

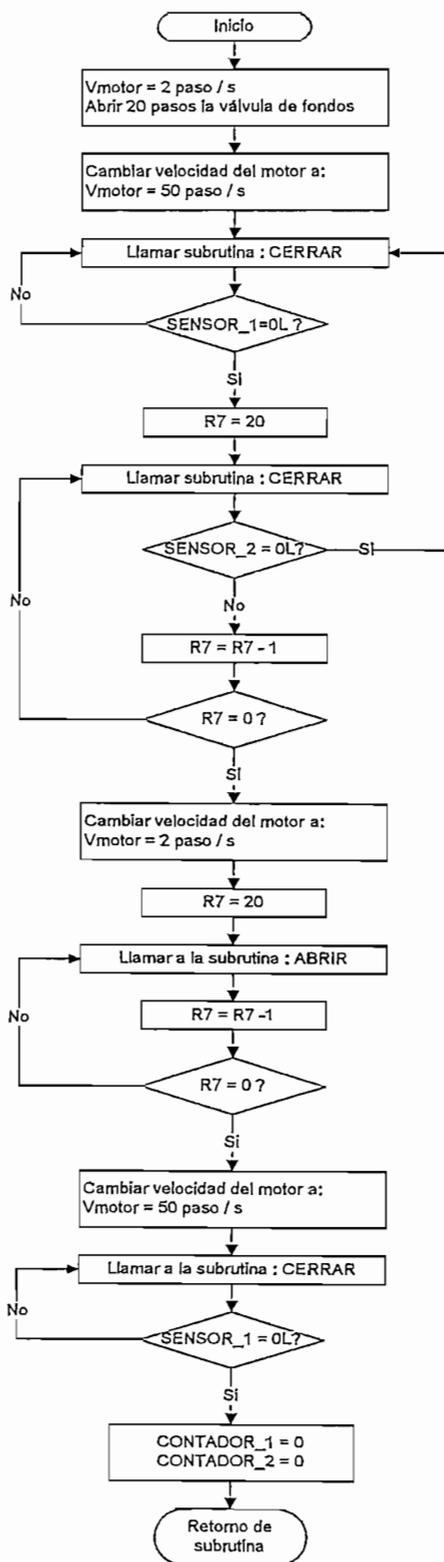


Figura 3.46

## SUBROUTINA SEN\_NIVEL

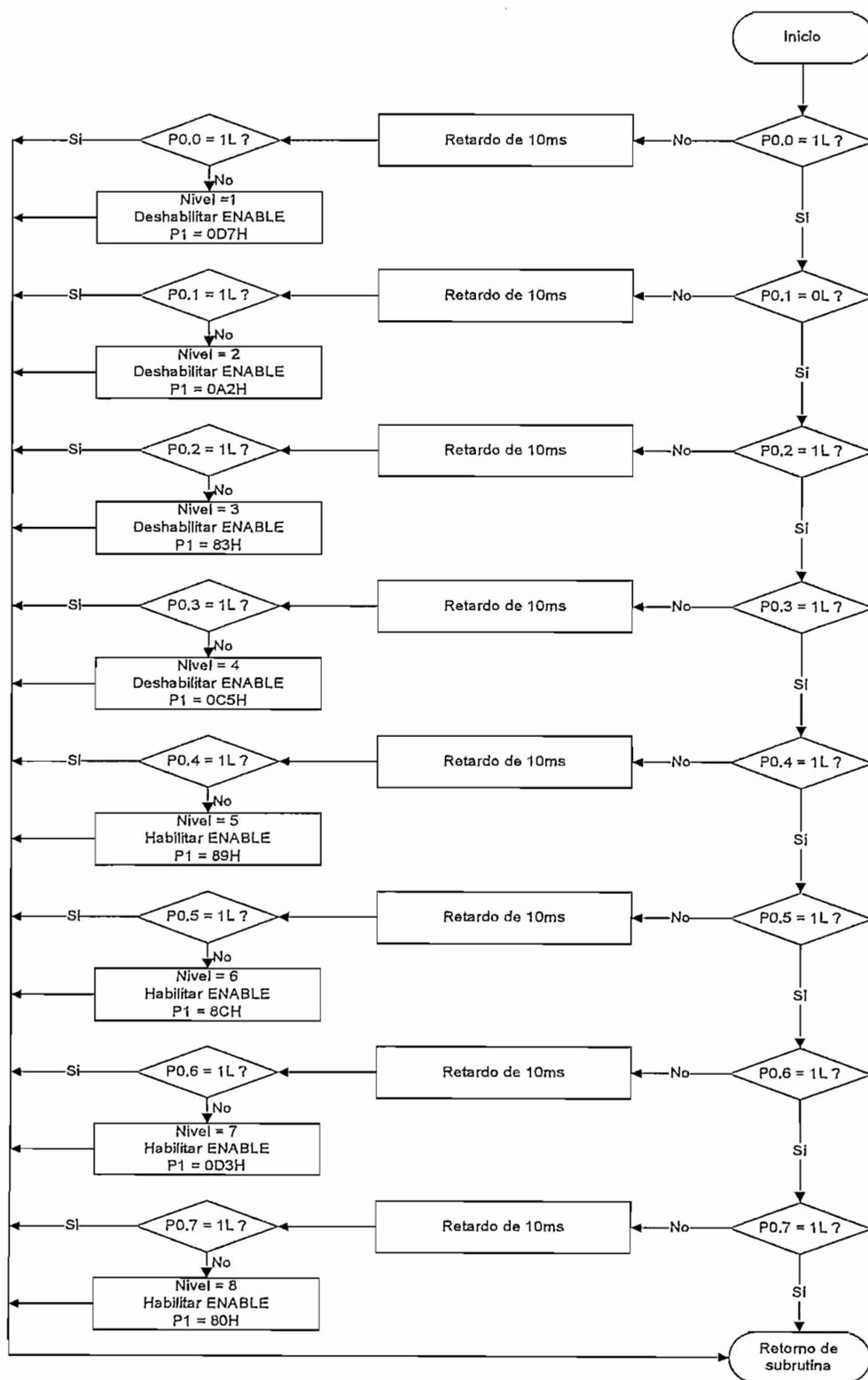


Figura 3.47

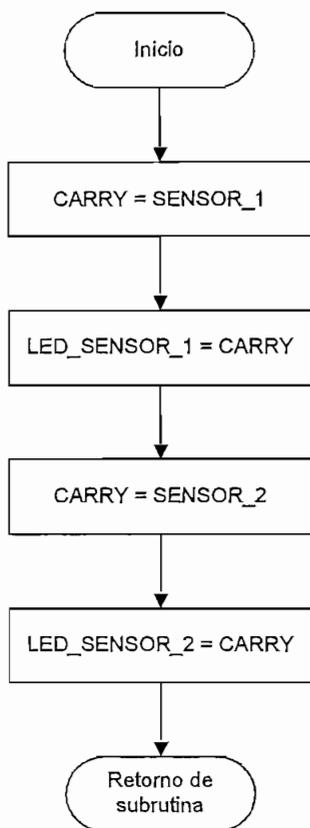
**SUBROUTINA ACT\_SENTORES**

Figura 3.48

## SUBROUTINA ABRIR

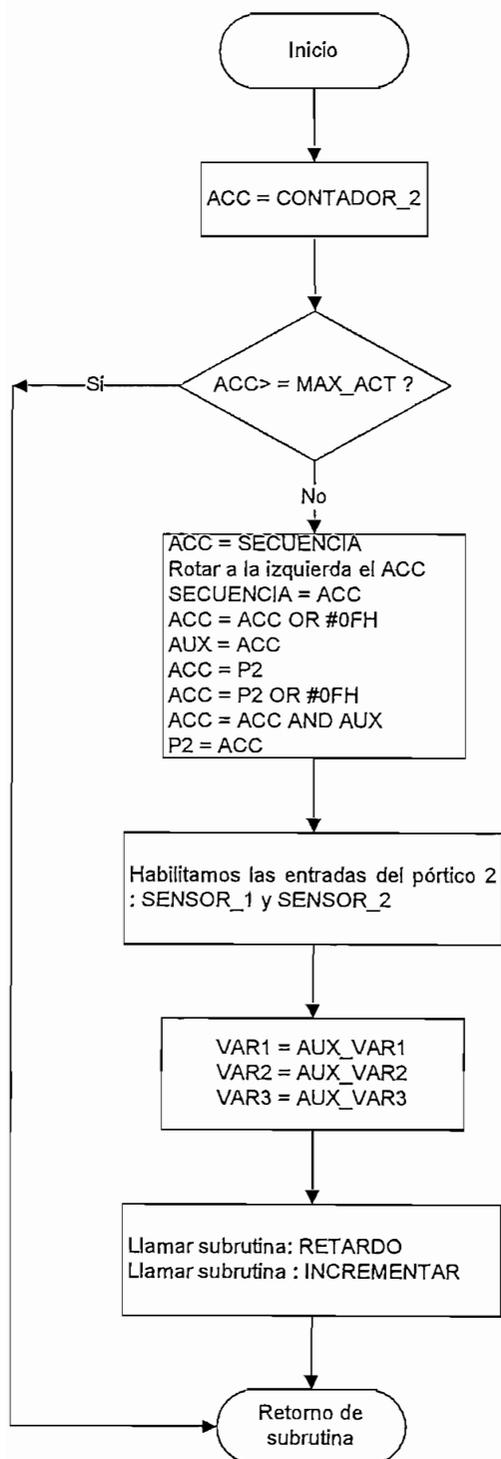


Figura 3.49

## SUBROUTINA CERRAR

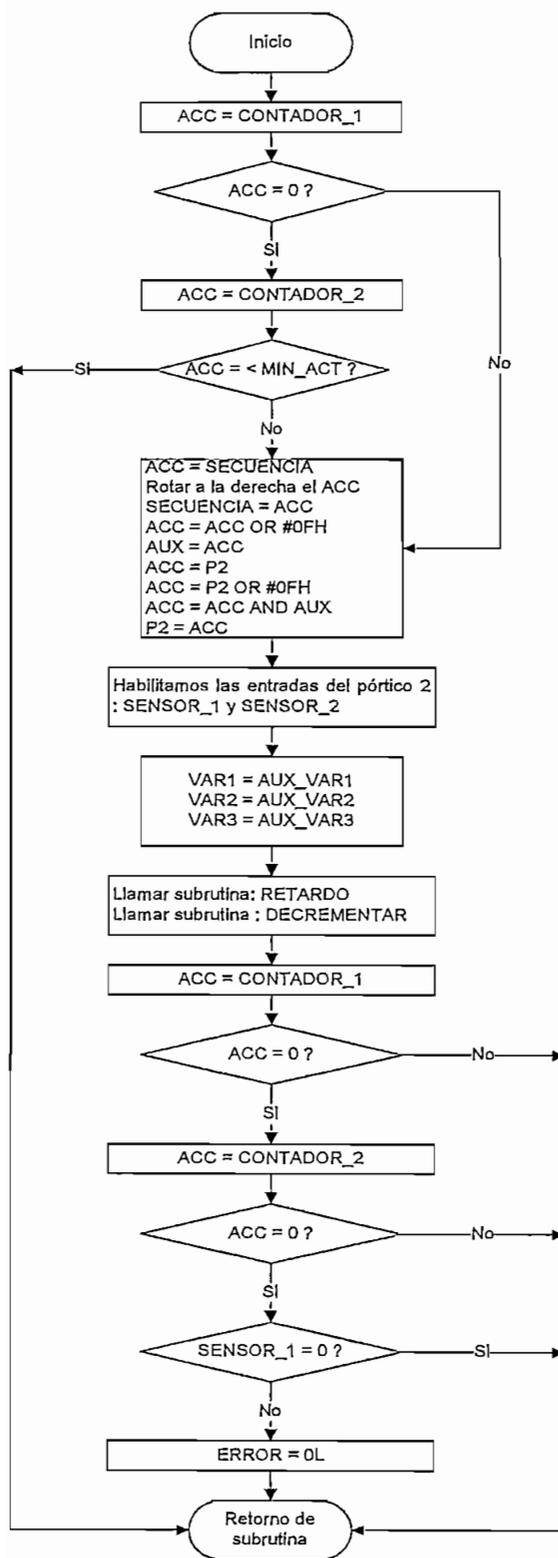


Figura 3.50

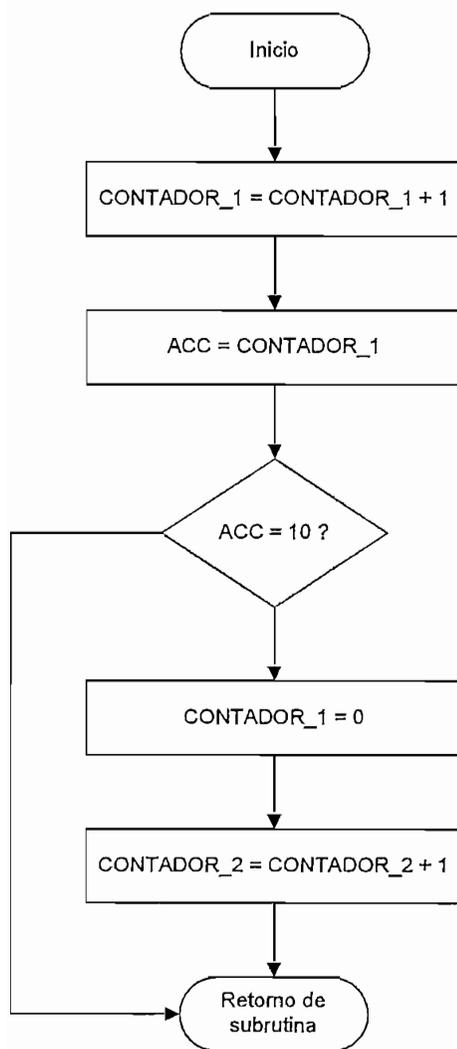
**SUBROUTINA INCREMENTAR**

Figura 3.51

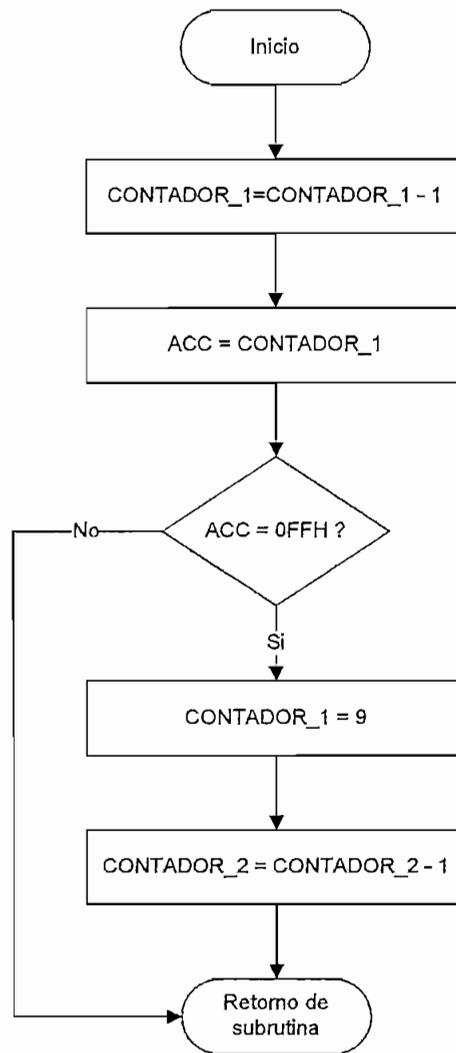
**SUBROUTINA DECREMENTAR**

Figura 3.52

## SUBROUTINA ACT\_MOTOR

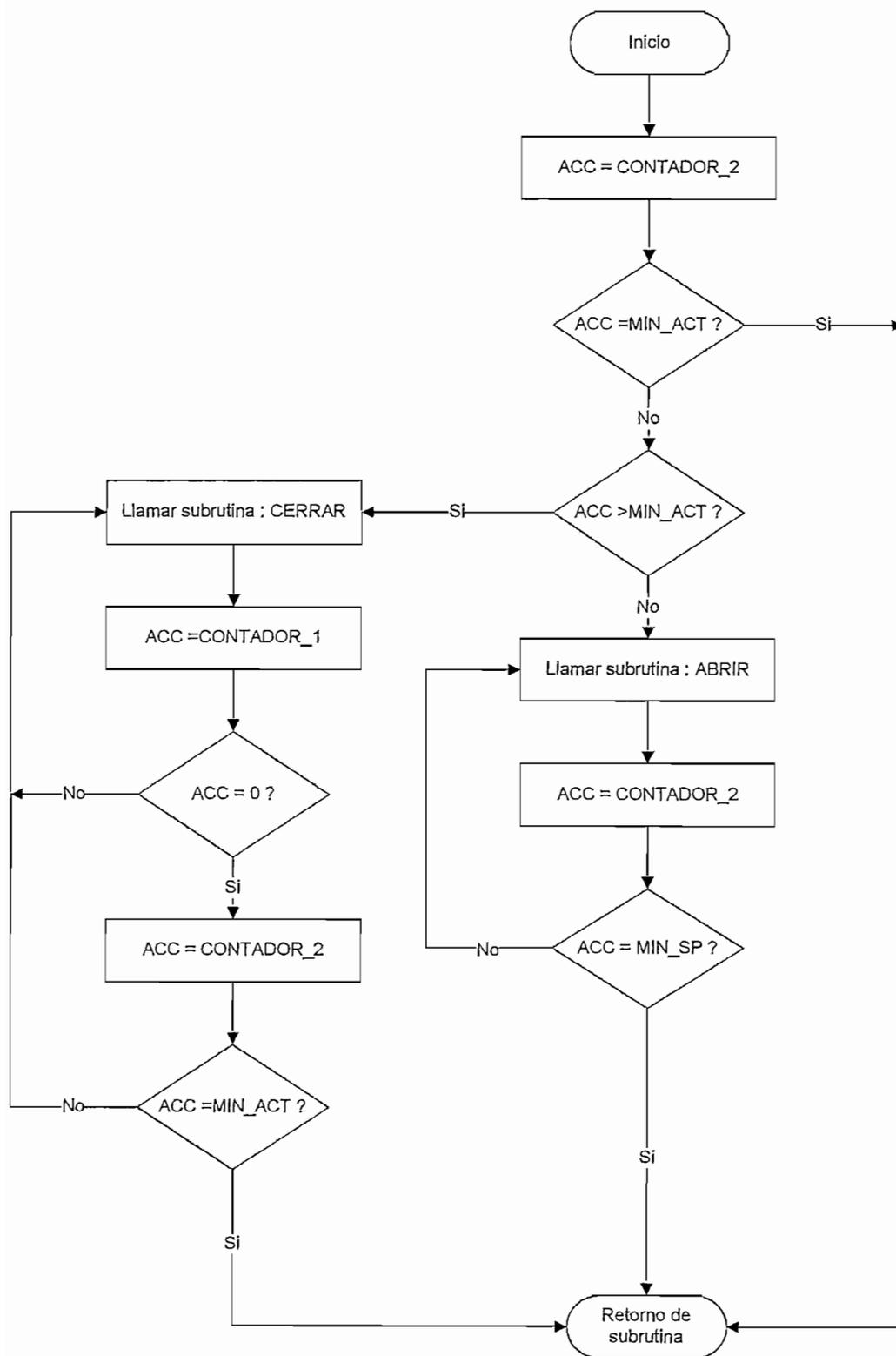


Figura 3.53

### **3.4 SUBROUTINAS PARA EL CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS**

El programa en lenguaje ensamblador que se ejecuta en el microcontrolador AT89C2051 para manejar el control del acumulador, ventilador y bombas se encuentra en el anexo 4. El análisis de los siguientes diagramas de flujo deber ser realizado conjuntamente con el listado de etiquetas que se incluye en el anexo 4.

#### **3.4.1 PROGRAMA PRINCIPAL**

El programa principal del control del acumulador, ventilador y bombas en su inicio se encarga de borrar la memoria RAM, así como de configurar los parámetros de la comunicación serial a 19200 bps, 9 bits de datos y ambiente multiprocesador, y también asigna la dirección 2DH a este dispositivo. Inicializa la variable TIEMPO con el valor 30 que equivale a 3 segundos de encendido de la bomba de cosecha. De ahí en adelante se entra en un bucle repetitivo; en el cual dependiendo del valor de R4 se ejecutará la subrutina MANUAL o la subrutina AUTOMATICO. El diagrama de flujo del programa principal, del cual se derivan el resto de subrutinas excepto las de interrupción, se muestra en la figura 3.54.

#### **3.4.2 SUBROUTINA AUTOMATICO**

Lo primero que hace esta subrutina es cargar a R5 con el valor 0, esto lo hace para que cuando se ejecute la subrutina MANUAL, se apague el REFLUJO, COSECHA y ALIMENTACIÓN. Después se muestrea el sensor de nivel de 3 niveles que se encuentra dentro del acumulador y esta conectado a los pines P1.5, P1.6 y P1.7. Existen 4 dispositivos que activa este control y son: bomba de realimentación, bomba de cosecha, bomba de alimentación y el ventilador. La activación de cada uno de ellos tiene relacionado un LED que están etiquetados como: L\_REFLUJO, L\_COSECHA, L\_ALIMENTAR y L\_VENTILAR. Las salidas que activan cada dispositivo se etiquetan igual pero sin el prefijo L\_.

Cuando se detecta el nivel inferior del acumulador que es P1.7 se apagan la cosecha y el reflujo. Si se detecta el segundo nivel del reboiler que es P1.6; se enciende el reflujo y la cosecha por el tiempo determinado por el producto de la variable TIEMPO por 100ms. Cada vez que se vuelva a detectar el nivel 2 se enciende la bomba de cosecha por el tiempo establecido. Pero si se detecta el nivel 3 que es el pin P1.5: se encienden permanentemente la bomba de cosecha y de reflujo hasta disminuir el nivel de líquido al nivel 2. Después se pasa a detectar primero el nivel dentro del tanque de reserva que es P1.3. Si el tanque de reserva está vacío se apaga la bomba de alimentación; pero si esta lleno depende del sensor del tanque de alimentación si se enciende o no la bomba de alimentación y al final se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.55.

### 3.4.3 SUBROUTINA MANUAL

En esta subrutina primero se empieza por apagar el reflujo la cosecha y la alimentación y poniendo en el registro R5 el valor 0FFH. Después se mueve al acumulador el valor de la variable COMANDO; dependiendo de que valor tiene el acumulador se ejecuta una de las siguientes funciones:

- Conmutar el encendido o apagado del REFLUJO y después mover a la variable comando el número 55H.
- Conmutar el encendido o apagado de la COSECHA y después mover a la variable comando el número 55H.
- Conmutar el encendido o apagado de la ALIMENTACION y después mover a la variable comando el número 55H.
- Conmutar el encendido o apagado del VENTILADOR y después mover a la variable comando el número 55H.
- No hacer nada.

Luego de haberse ejecutado una de las anteriores opciones se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.56.

### 3.4.4 SUBROUTINA INT\_SERIAL

Al momento que se produce una interrupción serial debido a que el bit RB8 es 1L en el modo ambiente multiprocesador, esta subrutina lo primero que hace es deshabilitar todas las interrupciones, borra la bandera de recepción RI y guarda el acumulador en la pila. Después de eso mueve el dato que se encuentra en el SBUF y se lo guarda en el acumulador. Primero se compara si el valor del acumulador; es igual a 0FH, el cual indica a todos los dispositivos esclavos que se debe cambiar de modo automático a manual; y esto se lo realiza cambiando el valor del registro R4 de 00H a FFH. Una vez realizado esto se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial.

Si el valor del acumulador no es igual a 0FH ahora se verificará el valor del registro R4 para saber si se está en modo manual o automático. Después de esta comprobación se pasa a verificar si el valor guardado en el acumulador es igual a la dirección del dispositivo, si es así dependiendo del valor antes verificado del registro R4 se ejecuta la subrutina INT\_SER\_AUTOMATICO si el valor de R4 es igual a 00H, o la subrutina INT\_SER\_MANUAL si el valor de R4 es igual a FFH. En ambos casos antes de la llamada a la subrutina, se deshabilita el bit SM2 para que la información que viene en camino solo la reciba el dispositivo con el que coincidió la dirección. Pero si la dirección no coincide se recupera el valor guardado del acumulador, se vuelve a habilitar la interrupción serial, se habilita el bit SM2, y se retorna de la interrupción serial. El diagrama de flujo de esta subrutina es el de la figura 3.57.

### 3.4.5 SUBROUTINA SERIAL\_AUTOMÁTICO

Esta subrutina es la encargada de recibir datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de nivel del reboiler, es decir 2DH y el registro R4 tiene el valor igual a 00H.

Si se recibe el dato 77H sirve para conmutar el encendido o apagado del ventilador. Pero si se recibe el dato 88H es para indicar que el siguiente dato debe almacenarse en la variable TIEMPO. El diagrama de flujo de esta subrutina es el de la figura 3.58

#### **3.4.6 SUBROUTINA SERIAL\_MANUAL**

Esta subrutina es la encargada de recibir datos a través del puerto serial, después de que se ha detectado que la dirección que se ha recibido corresponde a la del control de nivel del reboiler, es decir 2DH y el registro R4 tiene el valor igual a FFH.

Si se recibe el número 69H se espera recibir otro dato que será guardado en la variable COMANDO, cuyo valor determina como funcionará la subrutina MANUAL. Si se recibe el número 0FH le indica al dispositivo que debe cambiar de modo manual a modo automático y esto se lo realiza colocando en el registro R4 el valor 00H. Después se retorna de la subrutina. El diagrama de flujo de esta subrutina es el de la figura 3.59.

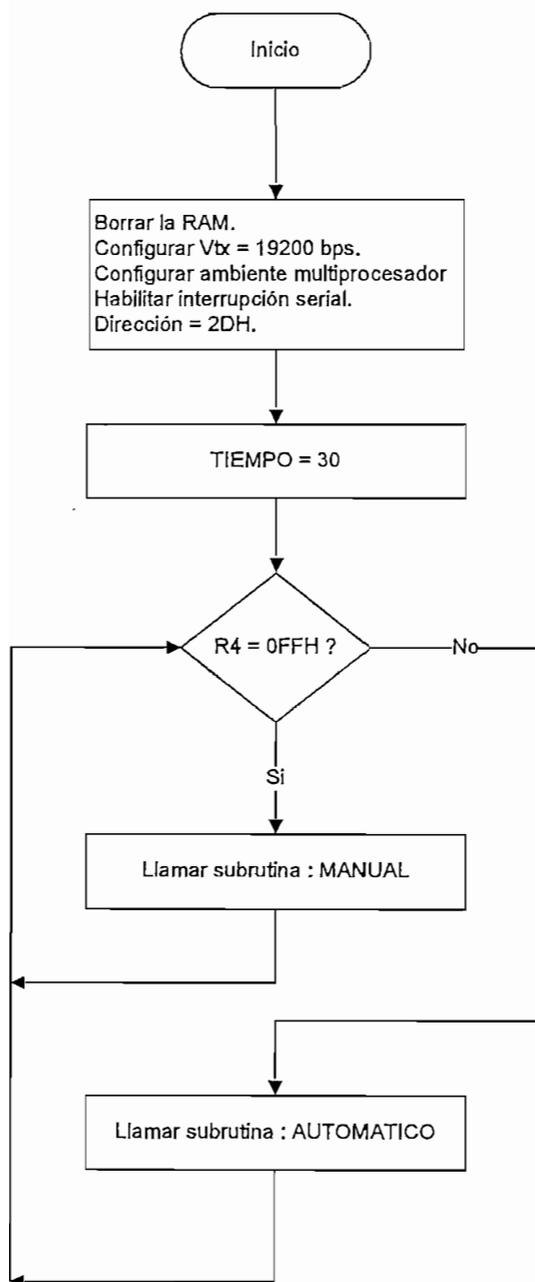
**PROGRAMA PRINCIPAL**

Figura 3.54

## SUBROUTINA AUTOMATICO

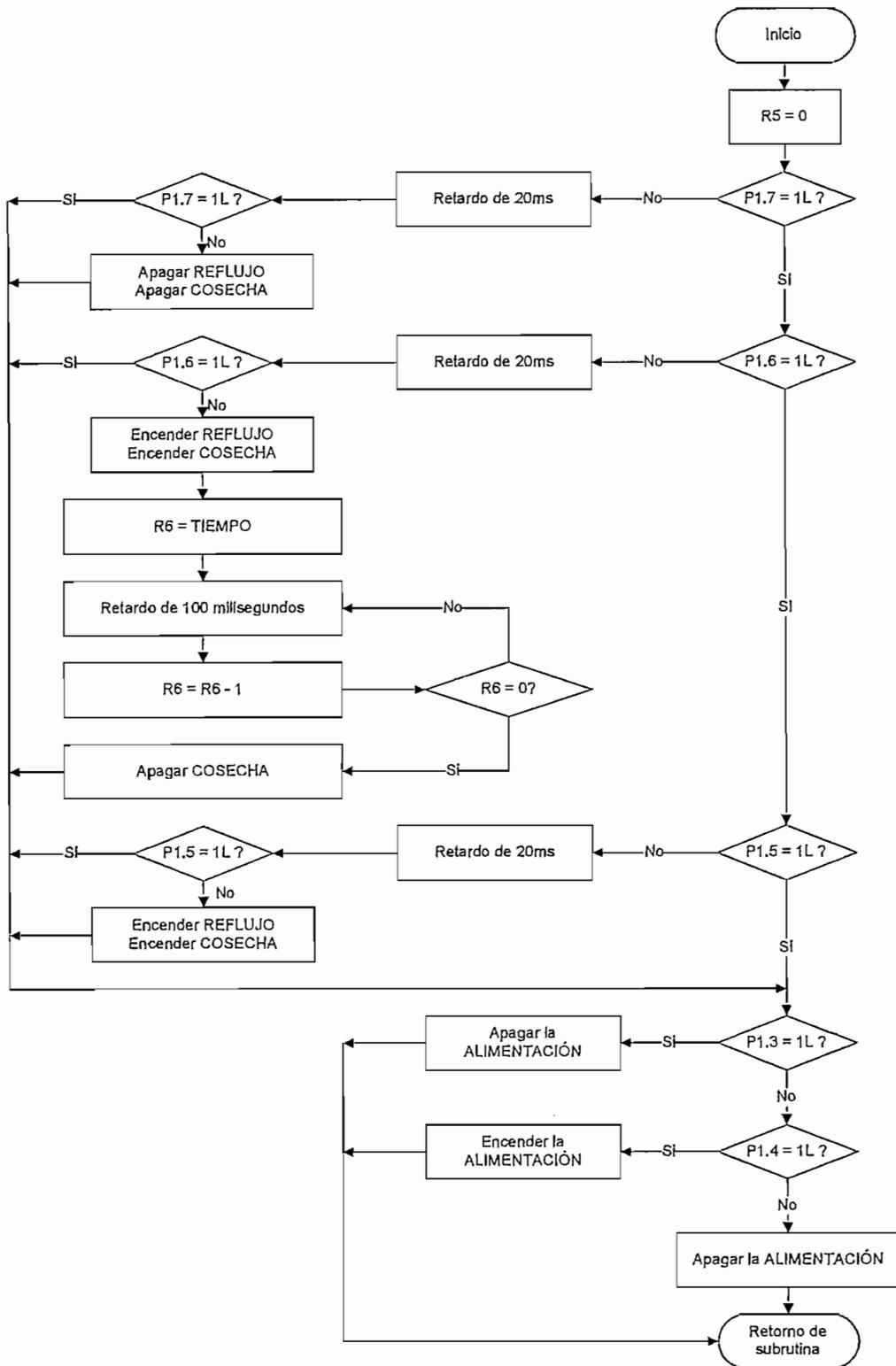


Figura 3.55

### SUBROUTINA MANUAL

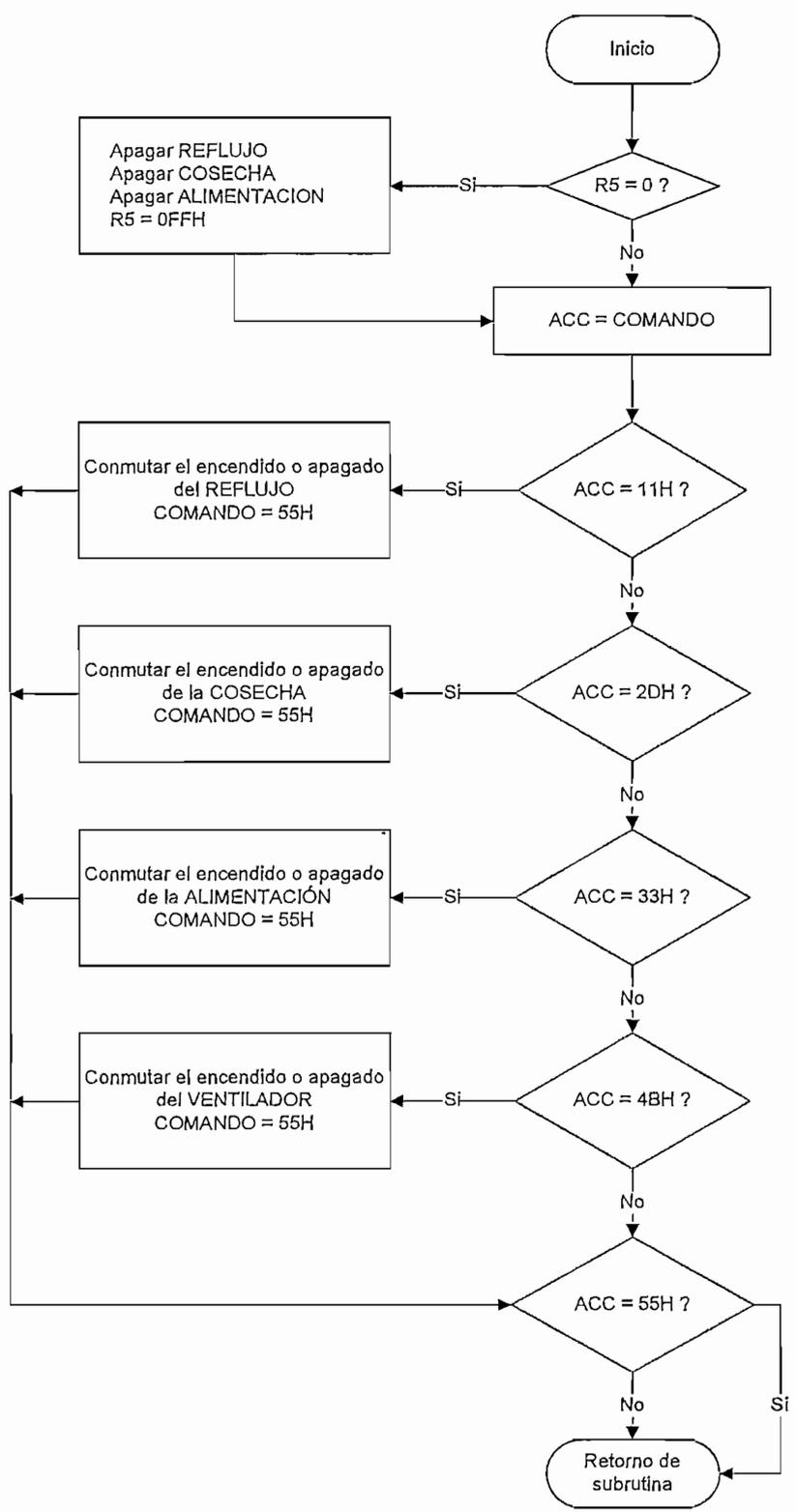


Figura 3.56

## SUBROUTINA INT\_SERIAL

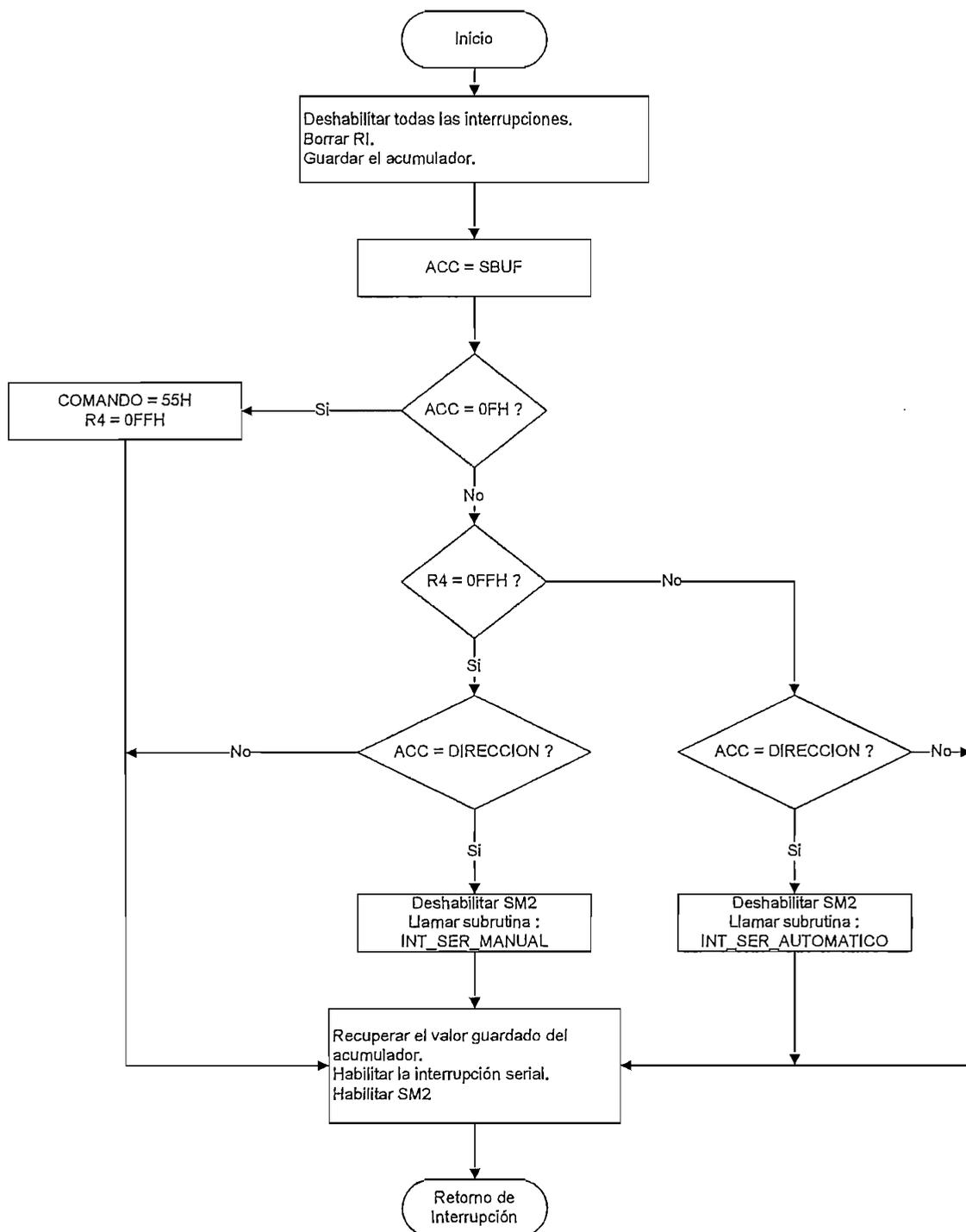


Figura 3.57

## SUBROUTINA SERIAL\_AUTOMATICO

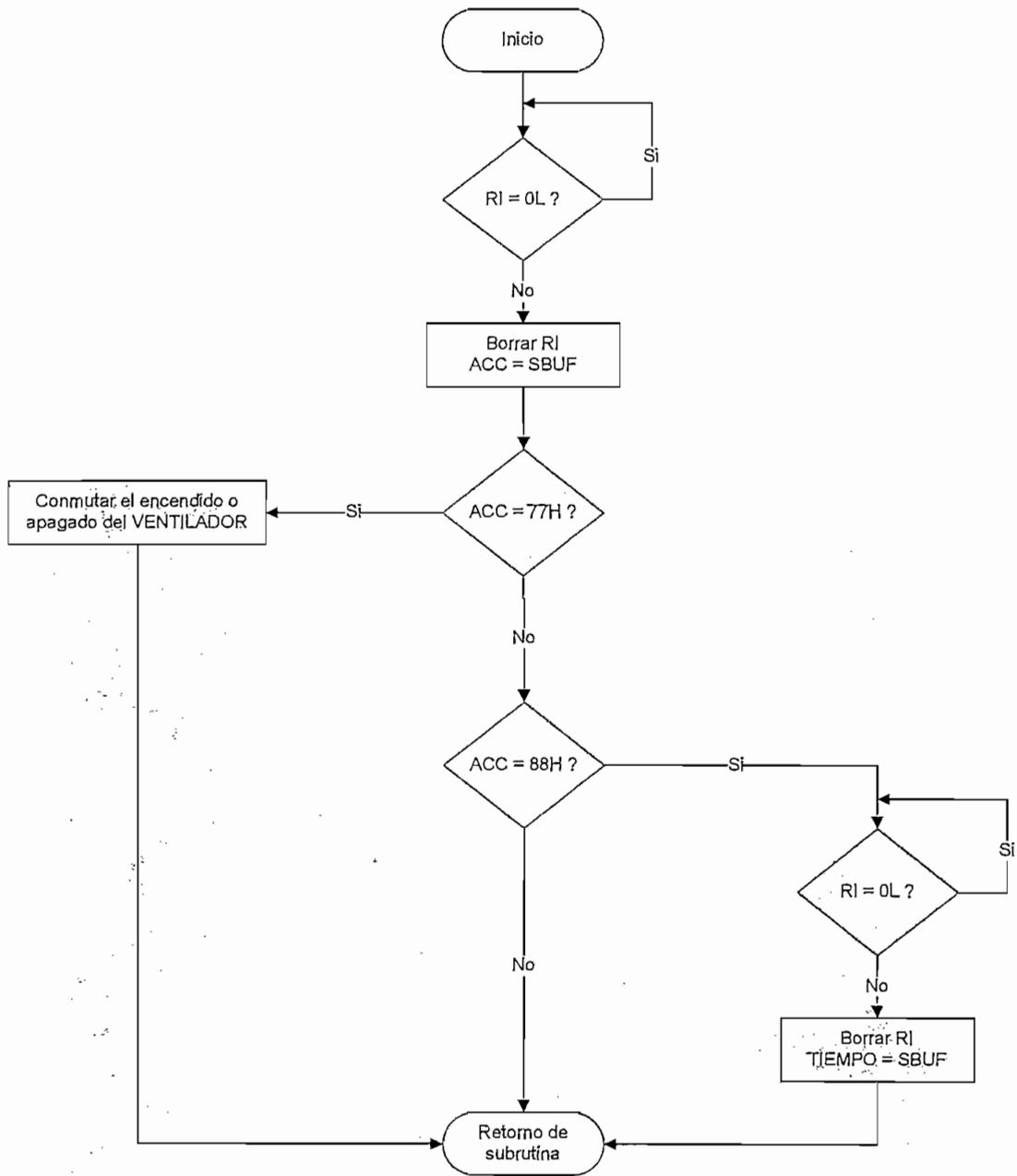


Figura 3.58

## SUBROUTINA SERIAL\_MANUAL

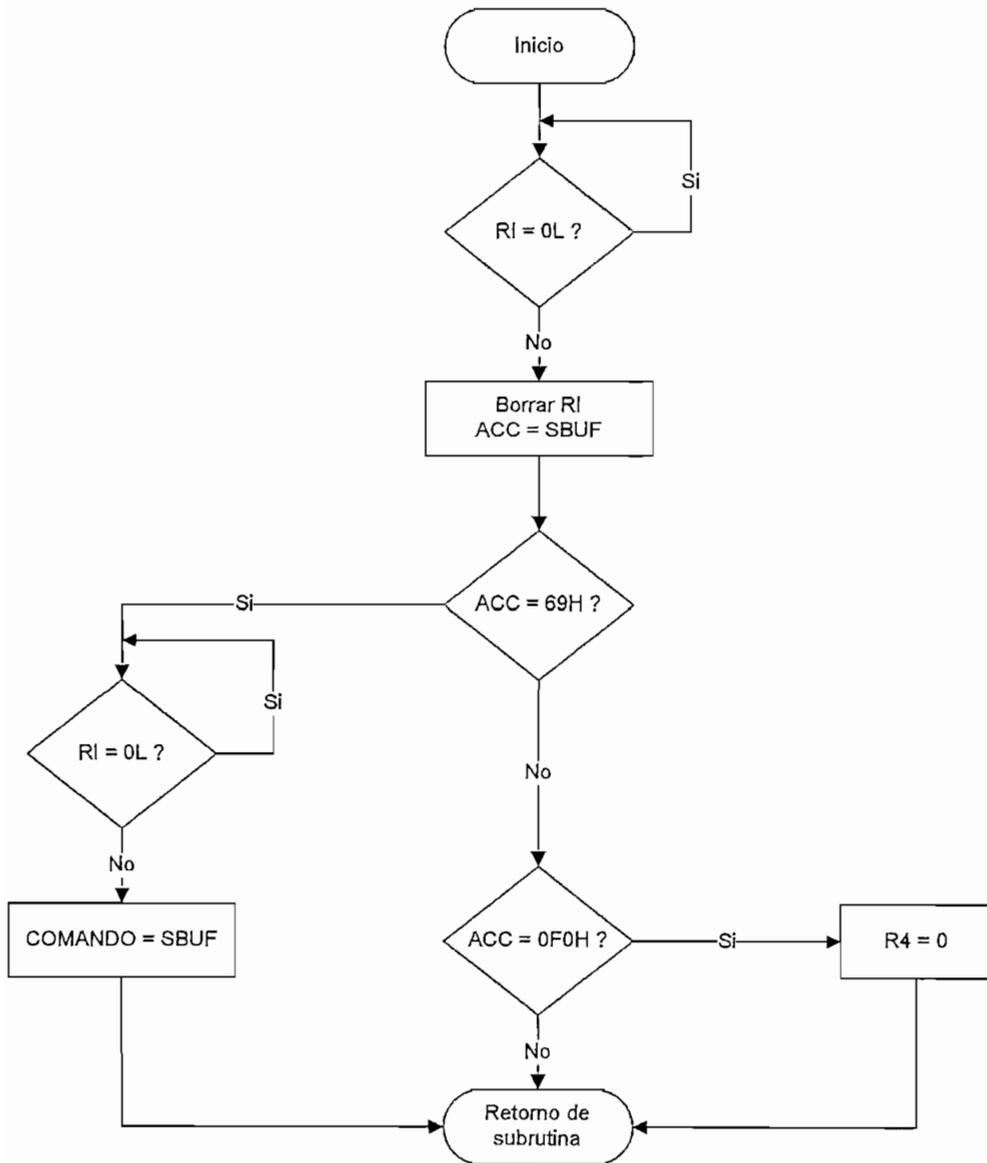


Figura 3.59

## CAPITULO 4: PRUEBAS Y MEDICIONES DEL EQUIPO CONSTRUIDO

### 4.1 PUESTA EN MARCHA DEL SISTEMA EN MODO AUTOMÁTICO

Lo primero que se realiza es el encendido del sistema por medio de una llave tubular, la cual es la encargada de energizar todo el equipo por medio del respectivo conmutador. Al momento que se energiza el sistema, se enciende un indicador luminoso de color rojo que indica que el sistema se encuentra energizado. Además se enciende un indicador luminoso de color amarillo, el cual indica que esta llegando energía al transformador utilizado en el circuito de detección del cruce por cero de la señal senoidal. Si cualquiera de los dos indicadores luminosos no se enciende implica que el equipo tiene alguna falla y que necesita revisión. Al mismo tiempo el motor paso a paso cierra por completo la válvula de fondos encendiendo el LED de color verde, en el circuito de control de nivel del reboiler; indicando así que la válvula de fondos está completamente cerrada. También pasados 4 segundos el control de temperatura del reboiler empieza a tomar medidas de temperatura en tres partes diferentes de la torre y enviarlas al panel principal, pero solo encenderá el dispositivo de calentamiento de la base si en el reboiler hay la suficiente cantidad de líquido.

Una vez encendido el equipo hay que ingresar todos los SET POINTS, de temperatura, tiempo y el número de pasos de rotación del motor paso a paso. Para ello se utiliza las opciones que ofrece el menú en el LCD con la ayuda del teclado.

Las primeras dos opciones que aparecen en el menú se muestran en la figura 4.1:

0	1	P	i	n	i	c	i	o	R	e	b	1
0	2	S	P	T	F	o	n	d	o	s		

Figura 4.1

La primera opción sirve para ingresar un número entre 0 y 100 que representa el porcentaje de potencia con el que trabajará el dispositivo de calentamiento al momento que se detecte la temperatura de disparo en la parte más alta de la torre. A partir de ahí el control de temperatura tomará este valor como el límite inferior al cual puede llegar el actuador.

Para ingresar a esta opción se debe digitar el número cero y después el uno ya que corresponden a la combinación mostrada en el menú para ingresar a dicha opción.

Se puede cancelar cualquier pulsación de una tecla con la tecla C, antes de ingresar el segundo dígito. Una vez ingresado el segundo dígito se pasa a un menú diferente, que en el caso de la opción 1 es el que se muestra en la figura 4.2.

P		i	n	i	c	i	a	l		R	e	b	o	i	1
(	0	%	-	1	0	0	%	)	=						

Figura 4.2

En la figura 4.2 el cursor se ubica una posición después del signo igual esperando el ingreso de un número de 3 dígitos. En caso de que el número a ingresarse sea menor a 100 se debe ingresar primero el número cero seguido de dos dígitos. El programa realiza una validación que es un filtro que impide el ingreso de cualquier número que no esté comprendido entre 000 y 100. Después de ingresados los tres números el programa espera que se presione una cualquiera de las tres teclas de función que son:

1. A    Aceptar.
2. B    Borrar.
3. C    Cancelar.

Si se presiona la tecla "A" el dato es enviado en este caso para el control de temperatura del reboiler y se retorna al menú principal. La tecla "A" no funciona si no se han ingresado los 3 números que conforman el dato.

Si se ha cometido un error al ingresar el dato, en cualquier momento el usuario puede hacer uso de la tecla B para borrar todos los números que se hayan ingresado.

Además, si el usuario considera que no es conveniente cambiar el SET POINT y ya ha ingresado a la plantilla de entrada de datos, en cualquier momento puede cancelar la operación y retornar al menú principal con la presión de la tecla C.

Cabe decir que el menú de entrada de datos es prácticamente el mismo para todos los ingresos de SET POINTS, sólo difiere en los mensajes que se muestran en la primera fila y a veces en la segunda. En todas las plantillas de entrada siempre se va a permitir el ingreso de un número comprendido entre 0 y 100. Por lo tanto para el ingreso de tiempo o número de pasos de rotación del motor paso a paso será necesario que el usuario multiplique el valor que ingresó por un factor que se indica en el menú para que sepa realmente como influirá el dato ingresado en el respectivo control.

La opción 2 del menú principal sirve para ingresar el SET POINT de temperatura de los fondos que servirá para realizar el control de temperatura del reboiler. Después de todas las pruebas efectuadas se ha logrado comprobar que cuando en el sensor de temperatura dentro del reboiler, marca una temperatura de 81 grados centígrados, la concentración de alcohol de la mezcla dentro del mismo es de 5 grados BRIX. Esta medida fue tomada por medio de un BRIXÓMETRO disponible en el laboratorio. Cuando el medidor de temperatura dentro del reboiler marca 82 grados centígrados paulatinamente se tiene un decremento de 5 a 3 grados BRIX.

Los grados BRIX son una medida indirecta del porcentaje de alcohol en la mezcla. Para saber el verdadero porcentaje de alcohol en la mezcla, se recurre a la figura

4.15, en donde se muestra una curva de los grados BRIX versus el porcentaje de alcohol. Entonces teniendo la medida en grados BRIX fácilmente podemos determinar que porcentaje de alcohol se tiene en la mezcla.

Es muy importante la selección apropiada del SET POINT de temperatura de fondos, ya que uno de los objetivos es que la concentración de alcohol que se saca por los fondos y se elimina sea la más baja.

La combinación de dos dígitos para ingresar al menú de entrada de datos del SET POINT de temperatura de fondos es 02; y una vez ingresada esta combinación de dos números estando en el menú principal, se presentará el menú de entrada de datos que se muestra en la figura 4.3.

S	P		T		F	o	n	d	o	s				
(	0		-		1	0	0	)	=					

Figura 4.3

El menú de la figura 4.3 es muy similar a la de entrada del porcentaje de potencia de inicio del reboiler lo que cambia es la leyenda en la primera fila del LCD. De ahí se aplican las mismas reglas que se señaló para la plantilla de entrada del SET POINT de potencia de inicio del reboiler.

Para avanzar en el menú se presiona la tecla "\*" y aparecerán las opciones que se muestran en la figura 4.4:

0	3		S	P		T		P	r	o	d	u	c	t	o
0	4		T	e	m	p	e		D	i	s	p	a	r	o

Figura 4.4

La opción 03 del menú principal sirve para ingresar el SET POINT de temperatura de trabajo y que se mide en la parte más alta de la torre. Se logró comprobar que cuando la temperatura en el tope de la torre es menor o igual a 71 grados se obtienen buenos productos, alrededor de 20 grados BRIX, cuya equivalencia en la

figura 4.15 es de 93% de alcohol. Debido a ello la principal meta del control de temperatura es mantener el SET POINT de temperatura de productos, y una vez logrado esto se dedica a alcanzar el SET POINT de temperatura de fondos.

Para ingresar al menú de entrada del SET POINT de temperatura de productos, se ingresa la combinación 03 en el menú principal y aparecerá la plantilla de entrada de datos de la figura 4.5:

S	P		T	P	r	o	d	u	c	t	o	s		
(	0		-	1	0	0	)	=						

Figura 4.5

En este caso, el dato que se ingresa es la temperatura en grados centígrados a la cual el usuario desea que opere la parte alta de la torre. Este dato también es analizado para realizar el control de temperatura del reboiler.

La opción 4 del menú principal sirve para ingresar la temperatura de disparo. Esta opción se relaciona con la opción 01 del menú principal, porcentaje de potencia de trabajo para el calentamiento, porque al momento que se alcanza la temperatura de disparo en el tope de la torre, automáticamente el control reduce la potencia al valor ingresado en la opción uno.

Lo que se sugiere es que el SET POINT de temperatura de disparo sea 2 grados más que el SET POINT de temperatura de los productos, porque de esa forma se le da la oportunidad al control de temperatura de actuar cuando la temperatura en la parte alta de la torre sea un grado más que el SET POINT de temperatura de productos; pero si el control no baja logra bajar la temperatura y se llega a la temperatura de disparo; se reduce la potencia al valor ingresado en la opción 1. De igual forma que los tres datos anteriores, este dato también es considerado para el control de temperatura del reboiler.

Luego se ingresa el tiempo de retardo de encendido de la bomba de cosecha, que es la opción 08 del menú principal. Para esto se presiona la tecla "\*" el número de

veces que sea necesario hasta visualizar la opción 07 y 08 que se presenta en la figura 4.6.

0	7		V	e	r		T		D	a	L	l	a	s	3
0	8		D	e	l	a	y		A	c	u	m	u	l	a

Figura 4.6

La opción 08 sirve para ingresar un valor entre 0 y 100, este valor se debe multiplicar por 100ms para saber que retardo es el que va a provocar con dicho dato, en el tiempo de encendido de la bomba de cosecha. Así por ejemplo, si se desea que la bomba de cosecha se encienda durante 4.7 segundos al momento que detecta el nivel 2 en el acumulador; hay que ingresar el dato 047. Este dato es la referencia para el control del acumulador.

La plantilla de datos que se visualizará al momento que se escoge esta opción será la mostrada en la figura 4.7.

D	e	l	a	y		A	C	C		*	1	0	0	m	s
(	0		-		1	0	0	)	=						

Figura 4.7

Las opciones 09 y 10 sirven para ingresar el máximo y mínimo número de pasos que el motor paso a paso abrirá la válvula de fondos. En ambos casos al número ingresado hay que multiplicarlo por 10 para saber el número de pasos que realmente se abrirá o cerrará la válvula. Para poder ver estas opciones se tiene que presionar la tecla "\*" o "#" hasta que aparezcan las opciones mostradas en la figura 4.8.

0	9		M	i	n		A	c	T		M	o	t	o	r
1	0		M	a	x		A	c	T		M	o	t	o	r

Figura 4.8

Con la opción 09 se ingresa la mínima apertura de la válvula de fondos. La válvula de fondos se abrirá ha este valor al momento que se alcance el SET POINT de temperatura de fondos. De acuerdo a varias pruebas realizadas, los valores que se han venido utilizando y que se recomiendan están entre 000 y 003. Este valor es enviado al control de nivel del reboiler. El caudal de salida de los fondos tiene que ser aproximadamente igual al de la alimentación para que el reboiler no se quede sin líquido. Es importante aclarar que hay que iniciar la alimentación cuando se haya llegado al SP de temperatura de fondos.

Al momento que se seleccione la opción 09 aparecerá la plantilla de entrada de datos mostrada en la figura 4.9.

M	A	X		A	p	e		M	o	t	o	r	*	1	0
(	0		-		1	0	0	)	=						

Figura 4.9

Con la opción 10 se ingresa la máxima apertura de la válvula de fondos. El control hará uso de este valor al momento que se sobrepase el nivel 6 dentro del reboiler, entonces la válvula de fondos se abrirá al valor máximo ingresado para desalojar el exceso de líquido dentro del reboiler. El dato que se ha utilizado para esta variable en la mayoría de pruebas es 010 que equivale a media vuelta de la válvula de fondos. Este dato también es enviado al control de nivel del reboiler.

De acuerdo con lo explicado se ha finalizado el ingreso de los SET POINTS, de ahí en adelante se procede a alimentar la torre hasta que el reboiler llegue al nivel 6 y esperar que el control actúe hasta alcanzar todos los SET POINTS y entrar en un régimen de operación automática.

Se podrá ver el progreso de la operación del sistema por medio de la visualización de las temperaturas y porcentaje de potencia con el que está funcionando el dispositivo de calentamiento. Las opciones 05, 06 y 07 del menú principal permiten visualizar la temperatura en tres diferentes partes de la torre y además en cada

una de ellas se puede observar en el LCD, el porcentaje de potencia con el que está funcionando el dispositivo de calentamiento.

Una vez que el sistema se encuentra en régimen de operación automática y por tanto en control continuo, se puede pasar a observar los datos de trabajo con la ayuda de las teclas "\*" y "#" desplazándose a través del menú principal hasta llegar a las opciones 05 y 06 como se muestra en figura 4.10.

0	5	V	e	r	T	R	e	b	o	i	l	1
0	6	V	e	r	T	D	a	l	i	a	s	2

Figura 4.10

Digitando la opción 05 en el menú principal se podrá visualizar el porcentaje de potencia actual con el que está trabajando el dispositivo de calentamiento en la primera fila del LCD, y en la segunda fila se muestra la temperatura dentro del reboiler, como se muestra a continuación en la figura 4.11.

P	%	R	B	a	s	e	=	5	7	%	
R	e	b	o	i	l	e	r	1	=	7	2

Figura 4.11

En la figura 4.11 se muestra, como un ejemplo, el porcentaje de potencia del dispositivo de calentamiento de la base que es 57% y la temperatura dentro del reboiler que para el caso es de 72 grados centígrados. La plantilla de visualización de datos se actualiza cada 4 segundos y sólo se puede retornar al menú principal presionando la tecla "C".

Si se ingresa la combinación 06 en cambio se visualizará la temperatura que existe en la parte media de la torre. Cabe destacar que el valor de temperatura utilizado para encender o apagar el ventilador es de 50 grados. La plantilla que se muestra cuando se digita 06 es la de la figura 4.12.

P	%		R		B	a	s	e	=			6	0	%	
T			D	a	l	l	a	s	2	=			7	0	

Figura 4.12

Y la última opción es la 07 que se muestra en el menú principal conjuntamente con la opción 08 y que se muestra en la figura 4.13.

0	7		V	e	r		T		D	a	l	l	a	s	3
0	8		D	e	l	a	y		A	c	u	m	u	l	a

Figura 4.13

La lectura del sensor de temperatura ubicado en la parte más alta de la torre sirve para dar una idea al usuario de que porcentaje de alcohol se está obteniendo en los productos, al respecto es importante tener en cuenta que el valor de esta temperatura es la referencia para realizar el control de calentamiento en la base. La plantilla que se visualizará al seleccionar esta opción es la de la figura 4.14.

P	%		R		B	a	s	e	=			5	5	%	
T			D	a	l	l	a	s	3	=			6	9	

Figura 4.14

A continuación se presenta la figura 4.15 en la que se muestra la curva de grados BRIX versus el porcentaje de alcohol; dicha curva, se ha determinado, ya que constituye una buena referencia para conocer el porcentaje de alcohol que tiene la mezcla en un momento determinado.

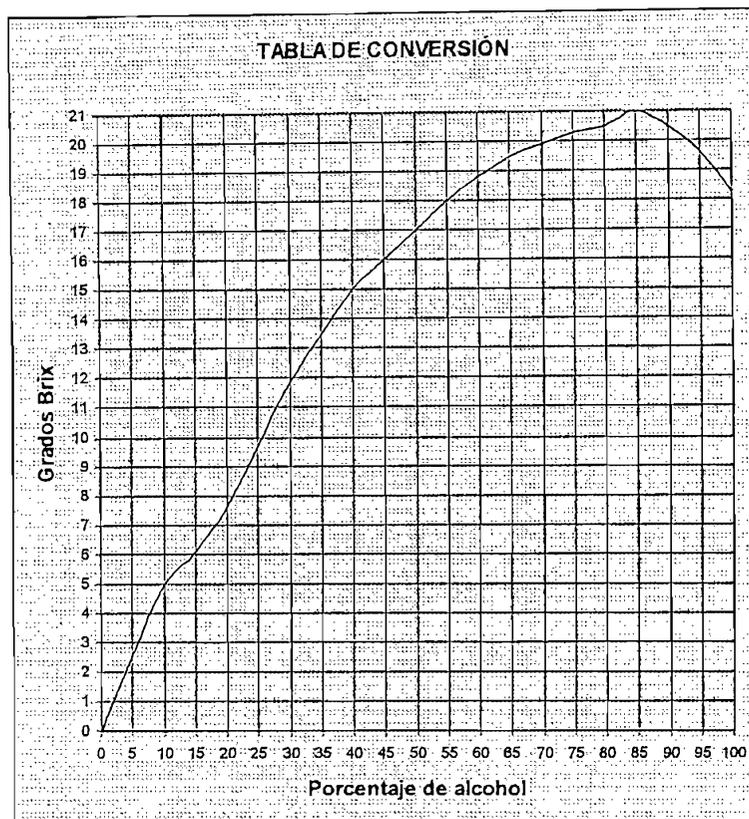


Figura 4.15

## 4.2 PUESTA EN MARCHA DEL SISTEMA EN MODO MANUAL

En el modo de operación manual, son muy pocas las acciones de control que se puede realizar, como por ejemplo apagar el dispositivo de calentamiento si no se encuentra sumergido en un suficiente nivel de líquido; por otra parte hay que tener mucho cuidado en cuanto a los niveles en el tanque de alimentación, en el acumulador y en el reboiler porque se puede o inundar la torre o derramar líquidos sobre el equipo.

Para pasar de modo de control automático a manual se debe presionar la tecla "D" e ingresar una clave de cuatro dígitos. Si se ingresan correctamente los dígitos de la clave se podrá visualizar un nuevo menú el cual controla todos los actuadores del sistema.

Las dos primeras opciones en visualizarse para realizar el control manual, son las que se muestran en la figura 4.16.

0	1		%	P		R	e	b	o	i	l	e	r		1
0	2		V	e	r		T		R	e	b	o	i	l	1

Figura 4.16

La primera opción sirve para manejar la potencia del dispositivo de calentamiento de la base. Si se ingresa a dicha opción se presentará una plantilla de datos muy similar a las anteriores con la que se podrá controlar directamente la potencia que llega al dispositivo de calentamiento de la base. La plantilla que se visualizará se muestra en la figura 4.17.

A	c	t	u	a	d	o	r		R	e	b	o	i	l	1
(	0	%	-	1	0	0	%	)	=						

Figura 4.17

En estas condiciones la resistencia estará trabajando a la potencia que se ingreso, a menos que el usuario ingrese otro porcentaje de potencia o que se retorne al control automático.

La opción 02 del menú del control manual es idéntica a la opción 05 del menú del control automático. De igual forma las opciones 03 y 04 del menú del control manual con las opciones 06 y 07 del menú del control automático; es decir permiten visualizar la temperatura en tres puntos diferentes de la torre así como también la potencia con la que está trabajando el dispositivo de calentamiento. Las opciones 03 y 04 del menú de control manual se presentan a continuación en la figura 4.18.

0	3		V	e	r		T		D	a	l	l	a	s	2
0	4		V	e	r		T		D	a	l	l	a	s	3

Figura 4.18

Las opciones 05 y 06 sirven para controlar el encendido o apagado de las bombas de reflujo y de cosecha respectivamente. No se visualizará ninguna otra opción de trabajo; simplemente a cada combinación de dos dígitos se envía un comando al respectivo control para que realice la acción de encender o de apagar dicho dispositivo. Las dos opciones que aparecerán en el menú serán las de la figura 4.19.

0	5			/	O		R	e	f	l	u	j	o		
0	6			/	O		C	o	s	e	c	h	a		

Figura 4.19

Las dos siguientes opciones son 07 y 08 sirven para el encendido o apagado de la bomba de alimentación y del ventilador. De igual forma que las dos opciones anteriores no se visualiza ninguna plantilla adicional; sino que se ejecuta un comando de acuerdo a la combinación de dos dígitos que se haya ingresado. Las dos opciones se presentan en la figura 4.20.

0	7			/	O		A	l	i	m	e	n	t	a	r
0	8			/	O		V	e	n	t	i	l	a	r	

Figura 4.20

Las opciones 09 y 10 sirven para abrir de una manera muy fina la válvula de fondos. Con la opción 09 se logra que la válvula se abra 10 pasos y con la opción 10 por el contrario se logra que la válvula de fondos se cierre 10 pasos. Ambas opciones se muestran en la figura 4.21.

0	9		A	b	r	i	r		1	0	p	a	s	o	s
1	0		C	e	r	r	a		1	0	p	a	s	o	s

Figura 4.21

Pero también se dispone de opciones para que la válvula de fondos se abra por completo, se cierre por completo o se detenga mientras se estaban ejecutando las dos anteriores. La opción 11 sirve para abrir la válvula de fondos por completo o

hasta que se seleccione la acción 12. Ambas opciones son mostradas en la figura 4.22.

1	1		A	b	r	i	r		v	a	v	u	l	a
1	2		D	e	t	e	n	e	r					

Figura 4.22

Y la última opción es la 13, la que permite cerrar por completo la válvula de fondos a menos que se seleccione la opción 12 antes de cerrar por completo la válvula de fondos. A continuación se muestra como se visualiza en el LCD esta opción en la figura 4.23.

1	1		C	e	r	r	a		v	a	v	u	l	a
1	4													

Figura 4.23

### 4.3 TABLAS DE MEDIDAS

#### 4.3.1 TABLA DEL CIRCUITO DE DETECCIÓN DE CRUCE POR CERO

Como se indicó en el capítulo 2, para saber que tiempo pasa la señal en el estado 1 lógico a la salida de la compuerta NAND de la figura 2.27 se utilizó el circuito y el programa del anexo 5, por medio de los cuales se obtuvieron 100 muestras, las mismas que son presentadas en la tabla 4.1.

TABLA DE DATOS OBTENIDA DEL DETECTOR DE CRUCE POR CERO							
N°-	Dato Hex	N°-	Dato Hex	N°	Dato Hex	N°-	Dato Hex
1	0676	26	067D	51	0672	76	0675
2	0681	27	0671	52	067E	77	0681
3	0676	28	067F	53	0671	78	0675
4	0682	29	0671	54	067E	79	0681
5	0676	30	067E	55	0673	80	0675

6	0680	31	0672	56	067E	81	0681
7	0675	32	067E	57	0671	82	0675
8	0682	33	0675	58	067F	83	0682
9	0675	34	0681	59	0672	84	0675
10	0680	35	0676	60	067E	85	0683
11	0675	36	0681	61	0671	86	0676
12	0681	37	0675	62	067D	87	0682
13	0677	38	0680	63	0672	88	0676
14	0681	39	0676	64	067E	89	0682
15	0675	40	0681	65	0682	90	0676
16	0680	41	0676	66	0675	91	0681
17	0672	42	0675	67	0681	92	0674
18	067E	43	0675	68	0675	93	0682
19	0672	44	0681	69	0682	94	0675
20	067D	45	0676	70	0675	95	0682
21	0672	46	0681	71	0683	96	0675
22	067E	47	0671	72	0676	97	0680
23	0671	48	0685	73	0682	98	0675
24	067E	49	0671	74	0675	99	0672
25	0672	50	067D	75	0681	100	067F

Tabla 4.1

#### 4.3.2 TABLA DE CONVERSIÓN DE LA SUBROUTINA COD2HEX

La subrutina COD2HEX es la encargada de realizar una conversión de código de la tecla presionada, que es el código que se lee en el pórtilo 2; a un valor hexadecimal entre 00H y 0FH que corresponde al BYTE menos significativo del DPTR que se muestra en la tabla 4.2.

TABLA T_CODIGO		
Tecla presionada	DPTR	Código
0	0E00H	0BEH
1	0E01H	077H
2	0E02H	0B7H
3	0E03H	0D7H
4	0E04H	07BH
5	0E05H	0BBH
6	0E06H	0DBH
7	0E07H	07DH
8	0E08H	0BDH
9	0E09H	0DDH
A	0E0AH	0E7H
B	0E0BH	0EBH
C	0E0CH	0EDH
D	0E0DH	0EEH
*	0E0EH	07EH
#	0E0FH	0DEH

Tabla 4.2

### 4.3.3 TABLA DE CONVERSIÓN DE LA SUBROUTINA HEX2ASCII

En vez de realizar operaciones matemáticas para hacer la conversión de hexadecimal a código ASCII, se optó por trabajar con una tabla que ya contiene los códigos ASCII utilizados. De esa forma el programa realiza la conversión de acuerdo a lo que se muestra en la tabla 4.3

TABLA T_ASCII		
Número HEX	DPTR	Código ASCII
0H	0E30H	30H
1H	0E31H	31H
2H	0E32H	32H
3H	0E33H	33H
4H	0E34H	34H
5H	0E35H	35H
6H	0E36H	36H
7H	0E37H	37H
8H	0E38H	38H
9H	0E39H	39H
AH	0E3AH	41H
BH	0E3BH	42H
CH	0E3CH	43H
DH	0E3DH	44H
EH	0E3EH	45H
FH	0E3FH	46H

Tabla 4.3

#### 4.3.4 TABLA PARA EL MANEJO DEL DISPLAY DE 7 SEGMENTOS

Por facilidad de hardware no se conectaron en orden los pines del pÓrtico 1 del microprocesador AT89C51, al display de 7 segmentos, debido a lo cual se creo la tabla 4.4 para poder determinar que código enviar al pÓrtico 1 para poder tener la visualización requerida.

TABLA PARA UTILIZAR EL DISPLAY DE 7 SEGMENTOS									
N°-	NC	d	c	g	b	a	f	e	Código
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
1	1	1	0	1	0	1	1	1	D7H
2	1	0	1	0	0	0	1	0	A2H
3	1	0	0	0	0	0	1	1	83H

4	1	1	0	0	0	1	0	1	C5H
5	1	0	0	0	1	0	0	1	89H
6	1	0	0	0	1	1	0	0	8CH
7	1	1	0	1	0	0	1	1	D3H
8	1	0	0	0	0	0	0	0	80H

Tabla 4.4

#### 4.3.5 TABLA PARA LA CONVERSIÓN DE PORCENTAJE DE POTENCIA A GRADOS DE DISPARO DEL TRIAC

La tabla 4.5 se utiliza para hacer una conversión de el número que envía el panel principal, el mismo que se encuentra entre 0 y 100, a un ángulo de disparo, que es representado por un número que puede variar entre 0 y 180.

TABLA T_PORCENTAJE2GRADO							
Porcentaje (%)	Ángulo (°)	Porcentaje (%)	Ángulo (°)	Porcentaje (%)	Ángulo (°)	Porcentaje (%)	Ángulo (°)
0	180	30	124	60	71	90	18
1	178	31	122	61	70	91	16
2	176	32	121	62	68	92	14
3	174	33	120	63	66	93	12
4	172	34	118	64	64	94	10
5	171	35	116	65	62	95	8
6	170	36	114	66	60	96	6
7	168	37	112	67	58	97	4
8	166	38	110	68	56	98	2
9	164	39	108	69	54	99	1
10	162	40	106	70	53	100	0
11	160	41	104	71	52		
12	158	42	103	72	50		
13	156	43	102	73	48		
14	154	44	100	74	46		

15	153	45	98	75	44		
16	152	46	96	76	42		
17	150	47	94	77	40		
18	148	48	92	78	38		
19	146	49	90	79	37		
20	144	50	88	80	36		
21	142	51	87	81	34		
22	140	52	86	82	32		
23	138	53	84	83	30		
24	136	54	82	84	28		
25	134	55	80	85	26		
26	132	56	78	86	24		
27	130	57	76	87	22		
28	128	58	74	88	21		
29	126	59	72	89	20		

Tabla 4.5

#### **4.3.5 TABLA PARA LA CONVERSIÓN DE ÁNGULO DE DISPARO DEL TRIAC A PORCENTAJE DE POTENCIA**

Para poder visualizar el porcentaje de potencia en el LCD que maneja el panel principal, el control de temperatura del reboiler hace una conversión del ángulo de disparo del TRIAC que es un número comprendido entre 0 y 180, a un número que simboliza el porcentaje de potencia que es un número comprendido entre 0 y 100 y que se enviará al panel principal para su posterior visualización en el LCD. La relación entre el ángulo de disparo y el porcentaje de potencia es mostrada en la tabla 4.6.

TABLA T\_GRADO2PORCENTAJE

Ángulo (°)	Porcentaje (%)	Ángulo (°)	Porcentaje (%)	Ángulo (°)	Porcentaje (%)	Ángulo (°)	Porcentaje (%)
0	100	50	72	100	44	150	17
1	99	51	71	101	43	151	16
2	98	52	71	102	43	152	16
3	98	53	70	103	42	153	15
4	97	54	69	104	41	154	14
5	97	55	69	105	41	155	13
6	96	56	68	106	40	156	13
7	96	57	68	107	40	157	12
8	95	58	67	108	39	158	12
9	95	59	67	109	39	159	11
10	94	60	66	110	38	160	11
11	93	61	66	111	37	161	10
12	93	62	65	112	37	162	10
13	92	63	64	113	36	163	9
14	92	64	64	114	36	164	9
15	91	65	63	115	35	165	8
16	91	66	63	116	35	166	8
17	90	67	62	117	34	167	7
18	90	68	62	118	34	168	7
19	89	69	61	119	33	169	6
20	89	70	61	120	33	170	6
21	88	71	60	121	32	171	5
22	87	72	59	122	31	172	4
23	87	73	59	123	31	173	3
24	86	74	58	124	30	174	3
25	86	75	58	125	30	175	2
26	85	76	57	126	29	176	2
27	85	77	57	127	29	177	1
28	84	78	56	128	28	178	1

29	83	79	55	129	27	179	0
30	83	80	55	130	27	180	0
31	82	81	54	131	26		
32	82	82	54	132	26		
33	81	83	53	133	25		
34	81	84	53	134	25		
35	80	85	52	135	24		
36	80	86	52	136	24		
37	79	87	51	137	23		
38	78	88	50	138	23		
39	78	89	50	139	22		
40	77	90	49	140	22		
41	77	91	49	141	21		
42	76	92	48	142	21		
43	76	93	48	143	20		
44	75	94	47	144	20		
45	75	95	46	145	19		
46	74	96	46	146	19		
47	73	97	45	147	18		
48	73	98	45	148	18		
49	72	99	44	149	17		

Tabla 4.6

## CAPÍTULO 5: ANÁLISIS FINANCIERO DE LA SOLUCIÓN IMPLEMENTADA

El presente capítulo tiene por objetivo realizar un análisis financiero de la solución implementada. Se analizará el costo de los materiales así como también el costo del trabajo de ingeniería realizado y un posible costo de venta al público de equipos similares.

### 5.1 DETALLE DE COSTO DEL CIRCUITO DE CONTROL DE TEMPERATURA DEL REBOILER

Cantidad	Código	Valor	Precio Unitario	Precio total
2	R1,10	330ohm	0,03	0,06
5	R4,5,6,7,9	560ohm	0,03	0,15
5	R12,13,14,15,16	4.7K	0,03	0,15
1	R3	8.2K	0,03	0,03
3	R2,8,11	10K	0,03	0,09
1	C0	0.1uF	0,12	0,12
1	C1	10uF	0,12	0,12
2	C2,C3	33pF	0,12	0,24
4	1N4003		0,10	0,40
1	Transformador		5,00	5,00
1	Zócalo 6 pines		0,12	0,12
1	ECG 3083		0,50	0,50
1	Zócalo 14 pines		0,12	0,12
1	ECG 4093		1,50	1,50
1	Zócalo 20 pines		0,40	0,40
1	AT89C2051		5,00	5,00
7	Conector 2 pines		0,40	2,80
5	Conector 4 pines		0,60	3,00
2	Bornera 3 pines		0,60	1,20
1	Pulsante		0,25	0,25
	Cristal			
1	11.0592MHz		0,90	0,90
3	DS1821		8,00	24,00
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
2	LED Rojo		0,10	0,20
1	LED Amarillo		0,10	0,10
1	LED Verde		0,10	0,10

1	Caja para proyectos	4,00	4,00
4	Postes	0,20	0,80
<b>Total</b>			<b>52,58</b>

## 5.2 DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DE TEMPERATURA

Cantidad	Código	Valor	Precio Unitario	Precio Total
1	R1	180ohm	0,25	0,25
1	R2	10Kohm	0,25	0,25
1	Zócalo 6 pines		0,12	0,12
1	ECG 3047		0,50	0,50
2	Borneras de 2 pines		0,60	1,20
1	Conector de 2 pines		0,40	0,40
1	TRIAC 56029		1,65	1,65
1	Disipador		5,00	5,00
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
4	Postes		0,20	0,80
1	Caja metálica 15x15		7,00	7,00
<b>Total</b>				<b>18,40</b>

## 5.3 DETALLE DE COSTO DEL CIRCUITO DE CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS

Cantidad	Código	Valor	Precio Unitario	Precio total
1	R1	8.2Kohm	0,03	0,03
1	R2	10Kohm	0,03	0,03
1	R3	10Kohm	0,03	0,03
1	R4	200Kohm	0,03	0,03
1	R5	1Kohm	0,03	0,03
1	R6	200Kohm	0,03	0,03

1	R7	1Kohm	0,03	0,03
1	R8	560ohm	0,03	0,03
1	R9	560ohm	0,03	0,03
1	R10	560ohm	0,03	0,03
1	R11	560ohm	0,03	0,03
1	R12	560ohm	0,03	0,03
1	R13	560ohm	0,03	0,03
1	R14	560ohm	0,03	0,03
1	R15	560ohm	0,03	0,03
1	C0	0,1uF	0,12	0,12
1	C1	10uF	0,12	0,12
1	C2	33pF	0,12	0,12
1	C3	33pF	0,12	0,12
1	C4	10uF	0,12	0,12
1	C5	10uF	0,12	0,12
2	1N4003		0,10	0,20
1	Cristal 11.0592MHz		0,90	0,90
1	Pull up 8 * 10K		0,20	0,20
1	Zócalo 14 pines		0,12	0,12
1	ECG 4093		0,60	0,60
1	Zócalo 20 pines		0,40	0,40
1	AT89C2051		5,00	5,00
4	Led Verde		0,10	0,40
10	Conector de 2 pines		0,40	4,00
1	Conector de 4 pines		0,60	0,60
1	Pulsante		0,25	0,25
2	Borneras de 3 pines		0,60	1,20
4	Postes		0,20	0,80
1	Caja para proyectos		4,00	4,00
3	Switch magnéticos		0,25	0,75
	Sensor de nivel			
1	blanco		22,36	22,36
1	Sensor de nivel gris		10,00	10,00
1	Sensor nivel brazo		10,00	10,00
1	Baquelita		0,8	0,80
1	Percloruro		0,43	0,43
	<b>Total</b>			<b>64,18</b>

#### 5.4 DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS

Cantidad	Código	Valor	Precio Unitario	Precio Total
1	R1	10Kohm	0,03	0,03
1	R2	1Kohm	0,03	0,03
1	R3	10Kohm	0,03	0,03
1	R4	1Kohm	0,03	0,03
1	R5	10Kohm	0,03	0,03
1	R6	1Kohm	0,03	0,03
1	R7	10Kohm	0,03	0,03
1	R8	1Kohm	0,03	0,03
4	1N4003		0,10	0,40
4	ECG 159		0,10	0,40
5	Relay 5V		1,00	5,00
8	Borneras 3 pines		0,60	4,80
4	Conector 2 pines		0,40	1,60
8	Postes		0,20	1,60
2	Cajas para proyectos		7,00	14,00
2	Tomacorrientes		3,00	6,00
1	Jack hembra		1,00	1,00
1	Jack macho		1,00	1,00
2	Bombas de pecera		12,00	24,00
1	Electroválvula Bomba de carro		25,00	25,00
1	12V		25,00	25,00
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
<b>Total</b>				<b>111,27</b>

#### 5.5 DETALLE DE COSTO DEL CIRCUITO DE CONTROL DE NIVEL DEL REBOILER

Cantidad	Código	Valor	Precio Unitario	Precio total
1	R1	8.2Kohm	0,03	0,03

1	R2	560ohm	0,03	0,03
1	R3	560ohm	0,03	0,03
1	R4	560ohm	0,03	0,03
1	R5	560ohm	0,03	0,03
1	R6	560ohm	0,03	0,03
1	R7	560ohm	0,03	0,03
1	R8	560ohm	0,03	0,03
1	R9	560ohm	0,03	0,03
1	R10	560ohm	0,03	0,03
1	R11	330ohm	0,03	0,03
1	R12	330ohm	0,03	0,03
1	R13	10Kohm	0,03	0,03
1	R14	10Kohm	0,03	0,03
1	R15	560ohm	0,03	0,03
1	R16	560ohm	0,03	0,03
1	R17	10Kohm	0,03	0,03
1	R18	10Kohm	0,03	0,03
1	C0	0,01uF	0,12	0,12
1	C1	10uF	0,12	0,12
1	C2	33pF	0,12	0,12
1	C3	33pF	0,12	0,12
1	Resistencia Pull up	10K*8	0,20	0,20
1	Zócalo de 40 pines		0,50	0,50
1	AT89C51		8,00	8,00
7	Zócalo de 14 pines		0,12	0,84
	Display ánodo			
1	común		2,00	2,00
1	Pulsante		0,25	0,25
1	Cristal 11.0592MHz		0,90	0,90
1	Optoacoplador		0,50	0,50
1	Zócalo 6 pines		0,12	0,12
2	Optoswitch		1,50	3,00
1	Led Rojo		0,10	0,10
1	Led Amarillo		0,10	0,10
1	Led Verde		0,10	0,10
5	Conector 2 pines		0,40	2,00
4	Conector 4 pines		0,60	2,40
1	Conector 8 pines		0,85	0,85
1	Caja para proyectos		4,00	4,00
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
4	Postes		0,20	0,80
	Sensores			
8	magnéticos		0,90	7,20
	<b>Total</b>			<b>36,11</b>

## 5.6 DETALLE DE COSTO DEL CIRCUITO DE POTENCIA DEL CONTROL DE NIVEL DEL REBOILER

Cantidad	Código	Valor	Precio Unit	Precio total
1	R1	560ohm	0,03	0,03
1	R2	560ohm	0,03	0,03
1	R3	560ohm	0,03	0,03
1	R4	560ohm	0,03	0,03
1	R5	10Kohm	0,03	0,03
1	R6	10Kohm	0,03	0,03
1	R7	10Kohm	0,03	0,03
1	R8	10Kohm	0,03	0,03
1	R9	1Kohm	0,03	0,03
1	R10	1Kohm	0,03	0,03
1	R11	1Kohm	0,03	0,03
1	R12	1Kohm	0,03	0,03
4	Diodos 1A 1N4003		0,10	0,40
4	transistores ECG 153		0,85	3,40
1	Motor paso a paso mediano		7,00	7,00
1	Bornera 3 pines		0,60	0,60
1	Conector 8 pines		0,85	0,85
1	Conector 4 pines		0,60	0,60
1	Caja para proyectos		7,00	7,00
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
4	Postes		0,20	0,80
<b>Total</b>				<b>22,24</b>

## 5.7 DETALLE DE COSTO DEL CIRCUITO DEL PANEL DE CONTROL

Cantidad	Código	Valor	Precio Unit	Precio total
1	R1	8,2Kohm	0,03	0,03
1	R2	1Kohm	0,03	0,03
1	R3	10Kohm	0,03	0,03
1	R4	10Kohm	0,03	0,03
1	R5	10Kohm	0,03	0,03
1	R6	10Kohm	0,03	0,03
1	R7	10Kohm	0,03	0,03

1	R8	10Kohm	0,03	0,03
1	C1	10uF	0,12	0,12
1	C2	33pF	0,12	0,12
1	C3	33pF	0,12	0,12
1	C4	10uF	0,12	0,12
1	C5	10uF	0,12	0,12
1	C6	10uF	0,12	0,12
1	C7	10uF	0,12	0,12
1	AT89C51		8,00	8,00
1	Zócalo con palanca		11,50	11,50
1	ECG 4093		0,60	0,60
1	Zócalo de 14 pines		0,12	0,12
1	MAX 232		3,20	3,20
1	Zócalo de 16 pines		0,12	0,12
1	ECG 4082		0,60	0,60
1	Zócalo de 14 pines		0,12	0,12
1	Cristal 11.0592 MHz		0,90	0,90
4	Bornera 3 pines		0,60	2,40
1	Pulsante		0,25	0,25
1	Llave tubular		5,50	5,50
1	Teclado		10,00	10,00
1	LCD		20,00	20,00
1	Bocina		2,50	2,50
1	ECG 159		0,10	0,10
2	Resisten Pull Up	10 Kohm	0,20	0,40
2	Conector de 8 pines		0,85	1,70
1	Conector de 4 pines		0,60	0,60
1	Conector de 2 pines		0,40	0,40
1	Caja para proyectos		12,00	12,00
1	Fuente de voltaje			0,00
1	Transformador		5,00	5,00
2	Indicadores 110V		0,60	1,20
1	Baquelita		0,80	0,80
1	Percloruro		0,43	0,43
4	Postes		0,20	0,80

**Total**

**90,32**

Todos los costos antes señalados, son los que efectivamente se realizaron para poder llevar adelante el proyecto y corresponden a costos de mercado local.

## 5.8 COSTO DE INGENIERÍA E IMPLEMENTACIÓN

El tiempo de diseño de los sistemas de control tomo alrededor de 160 horas si la hora ingeniero en este tipo de proyectos esta alrededor de 10 dólares se tendrá:

$$\begin{aligned} \text{Costo}_{\text{INGENIERIA}} &= \text{Costo}_{\text{HORA}} \times \text{NúmeroDeHoras} \\ \text{Costo}_{\text{INGENIERIA}} &= 10 \frac{\text{dólares}}{\text{hora}} * 160 \text{horas} \\ \text{Costo}_{\text{INGENIERIA}} &= 1600 \text{dólares} \end{aligned}$$

Por otra parte, también tomó un tiempo de 160 horas las pruebas y ajustes requeridos para conseguir que el equipo trabaje de acuerdo con lo previsto, por lo que bajo la misma consideración anterior, para esta parte se tendría:

$$\begin{aligned} \text{Costo}_{\text{IMPLEMENTACIÓN}} &= \text{Costo}_{\text{HORA}} \times \text{NúmeroDeHoras} \\ \text{Costo}_{\text{IMPLEMENTACIÓN}} &= 10 \frac{\text{dólares}}{\text{hora}} * 160 \text{horas} \\ \text{Costo}_{\text{IMPLEMENTACIÓN}} &= 1600 \text{dólares} \end{aligned}$$

## 5.9 CÁLCULO DEL PRECIO DE VENTA AL PÚBLICO

Haciendo una suma de todos los costos involucrados se tendrá:

### Costo total del proyecto

Circuito de control de temperatura	52,58
Circuito de potencia del control de temperatura	18,40
Circuito de control del acumulador	64,18
Circuito de potencia del control del acumulador	111,27
Circuito de control de nivel del reboiler	36,11
Circuito de potencia de control de nivel del reboiler	22,24
Circuito del panel principal	90,32
Imprevistos	150,00
Costo de ingeniería	1600,00
Costo de implementación	1600,00

$$\text{Precio}_{\text{EQUIPO}} = 1.3 \times \text{InversiónFija}_{\text{EQUIPO}}$$

$$\text{Precio}_{\text{EQUIPO}} = 1.3 \times 545.10 \text{dólares}$$

$$\text{Precio}_{\text{EQUIPO}} = 708.63 \text{dólares}$$

Ahora tomando en consideración que se prevé vender 40 equipos en 2 años, se dividirá la suma de los costos de ingeniería e implementación entre el número de equipos para de esa forma el resultado de esta operación se sume al precio del equipo y obtener un precio de venta al público final. El resultado de la división será un precio de recuperación del costo de ingeniería e implementación por equipo.

$$\text{Precio}_{\text{RECUPERACIÓN}} = \frac{\text{Costo}_{\text{INGENIERÍA}} + \text{Costo}_{\text{IMPLEMENTACIÓN}}}{\text{NúmeroEquipos}}$$

$$\text{Precio}_{\text{RECUPERACIÓN}} = \frac{1600 \text{dólares} + 1600 \text{dólares}}{40}$$

$$\text{Precio}_{\text{RECUPERACIÓN}} = 80 \text{dólares}$$

Con lo que el precio de venta al público será:

$$PVP = \text{Precio}_{\text{EQUIPO}} + \text{Precio}_{\text{RECUPERACIÓN}}$$

$$PVP = 708.63 \text{dólares} + 80 \text{dólares}$$

$$PVP = 788.63 \text{dólares}$$

## CAPÍTULO 6: COMENTARIOS Y CONCLUSIONES

### 6.1 COMENTARIOS

1. Una primera recomendación importante, es continuar trabajando en la temática abordada; en este sentido, se puede mejorar el funcionamiento de la torre si se coloca una electroválvula en la tubería que alimenta la torre, para cuando no se consigue la concentración deseada de los fondos en el reboiler se pueda controlar el nivel dentro del reboiler por medio de la apertura o cierre de la alimentación.
2. Otra recomendación sería que se desarrolle un programa para un computador personal, dicho programa permitirá contar con una interfaz hombre-máquina amigable al usuario y se podría contar con información en tablas y gráficos sobre el comportamiento de la operación de la torre de destilación. Además, se puede aprovechar las facilidades matemáticas de las que dispone un computador para poder realizar controles mas complicados como PID.
3. En el diseño circuital, se utilizó resistencias de 560 ohmios para limitar la corriente que circula a través de los LEDs en polarización directa, pero la luminosidad de los LEDs no es muy buena, debido a ello se recomienda que una mejora que se puede realizar es primero aislar los pines de salida del pòrtico por medio de una compuerta CMOS para después realizar una amplificación de corriente por medio del transistor.
4. Otra recomendación es que a los pines libres del microcontrolador que maneja el panel principal se le coloque resistencias de 4.7Kohmios hacia la fuente de polarización, a fin de evitar problemas de ruido y posteriormente aprovechar estas líneas libres para en futuras aplicaciones poder conectar otros medidores de temperatura en base al circuito DS1821.

5. En la construcción del acumulador de la torre de destilación por medio de un tubo de vidrio y tapas de plástico, resultó no ser un buen sellador el silicón debido a que siempre se producen fugas en las uniones. Se recomienda construir el acumulador no en vidrio, si no en acero inoxidable de la misma manera que el reboiler.
  
6. Si en un futuro para ir mejorando la operación de la torre, se necesitara realizar el control de la válvula de alimentación y de la válvula de reflujo, se recomienda usar válvulas del tipo aguja ya que permiten flujos muy pequeños de líquido y por ende un control más exacto del proceso.

## 6.2 CONCLUSIONES

1. El microcontrolador AT89020C51 presenta muchas limitaciones de corriente en comparación a los microcontroladores PIC, debido a ello se necesita conectar buffers a la salida de cada pin, y si se necesita una amplificación de corriente mayor por lo que se utilizaron transistores que trabajen en corte y saturación y cuya base es controlada por el buffer que está conectado al pin del microcontrolador. Por lo que a futuro se podría pensar en una solución circuital que utilice microcontroladores que tengan la posibilidad de manejar los niveles de corriente esperados.
  
2. También se logró comprobar que el microcontrolador AT89C2051 presenta problemas al manejar cargas inductivas, inclusive si se utiliza un buffer y un transistor; debido a ello fue necesario aislar ópticamente la salida del microcontrolador por medio de un optoacoplador y a la salida del optoacoplador se realizó una amplificación adicional por medio de un transistor en corte y saturación para controlar las cargas inductivas como son los relays.

3. Una ventaja de los microcontroladores AT89C51 y AT89C2051 con respecto a los microcontroladores PIC que podría ser una alternativa a considerarse es que tienen más instrucciones de programación y por ende la programación es mucho más sencilla.
4. El control descentralizado pensado desde el inicio dio buen resultado en la realización de este proyecto. Facilita mucho el diseño el hacer que no todas las señales lleguen a un mismo microcontrolador y también facilita el mantenimiento del equipo.
5. La inclusión del circuito integrado DS1821 en este proyecto representó una forma innovadora y poco tradicional de sensar la temperatura y convertir esa lectura analógica a en formato digital. Además, las subrutinas desarrolladas en este proyecto para el protocolo ONE WIRE pueden ser utilizadas para lograr la comunicación con otros dispositivos ONE WIRE de la fábrica DALLAS.
6. Otra cosa importante que resaltar son los códigos que se utilizaron tanto en los comandos como en las direcciones en la comunicación serial de los microcontroladores. Estos códigos permiten una detección de errores muy buena pero si se deseará mejorar el protocolo de comunicaciones se necesitaría incluir acuses de recibo e intervalos de tiempo. Pero esto no es necesario debido a que el interfaz RS232 soporta una velocidad de transmisión máxima de 19200bps a 15m, y la máxima distancia a la que se encuentran los dispositivos más lejanos en este proyecto es escasamente 3 metros.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] SHREVE, Norris. The Chemical Process Industries. 2<sup>da</sup> edición. McGraw-Hill. London. 1956.
- [2] KISTER, Henry. Distillation Operation. 1<sup>ra</sup> edición. McGraw-Hill. London. 1990.
- [3] RASHID, Muhammad. Electrónica de Potencia: circuitos, dispositivos y aplicaciones. 2<sup>da</sup> edición. Prentice Hall. México. 1995.
- [4] WILLIAMS, Arthur. Manual de Circuitos Integrados: selección, diseño y aplicaciones. 1<sup>ra</sup> edición. MacGraw-Hill. México. 1994.
- [5] TANENBAUM, Andrew. Redes de computadoras. 4<sup>ta</sup> edición. Prentice Hall. México. 2003.
- [6] ATMEL. Data Book, AT89 Series Flash MCUs. ATMEL. San José. 1997.

## DIRECCIONES ELECTRÓNICAS

- <http://www.todorobot.com>
- <http://www.hobbiepic.com>
- <http://www.maxim-ic.com/1-Wire.cfm>

# ANEXO 1

# ANEXO 1: PROGRAMA DEL PANEL PRINCIPAL

CPU "E:\SIDES51\0051.TBL"

HOF "INT8"

INCL "E:\SIDES51\SFR"

```
.....  
;TEMA: PANEL PRINCIPAL  
.....
```

```
.....  
;REALIZADO POR: Marco Antonio Chávez  
.....
```

```
.....  
;CONSTANTES Y VARIABLES  
.....
```

```
DIREC_1: EQU 11H  
DIREC_2: EQU 2DH  
DIREC_3: EQU 33H  
  
DIRECCION: EQU 50H  
DATO_INGRESADO: EQU 51H  
AUTO_MANUAL: EQU 52H  
PARTE_A: EQU 53H  
PARTE_B: EQU 54H  
OPCION: EQU 55H  
  
MEDIDOR_1: EQU 56H  
MEDIDOR_2: EQU 57H  
MEDIDOR_3: EQU 58H  
MEDIDOR_4: EQU 59H  
  
ACT_REBOILER1: EQU 5AH  
POTEN_INICIAL: EQU 5BH  
SP_T_FONDOS: EQU 5CH  
SP_T_PRODUC: EQU 5DH  
T_DISPARO: EQU 5EH  
MIN_ACT_MOTOR: EQU 5FH  
MAX_ACT_MOTOR: EQU 60H  
  
DELAY_ACC: EQU 61H  
LOW_SP_MOTOR: EQU 62H  
HIGH_SP_MOTOR: EQU 63H  
  
BAN: EQU 64H  
BANDERA_2: EQU 65H  
BANDERA_1: EQU 66H  
  
VAR1: EQU 67H  
VAR2: EQU 68H  
VAR3: EQU 69H  
  
AUXILIAR_DPL: EQU 6AH  
AUXILIAR_DPH: EQU 6BH  
CODIGO_HEX: EQU 6CH  
FILTRO: EQU 6DH  
COMANDOS: EQU 6EH  
  
CLAVE1: EQU 6FH  
CLAVE2: EQU 70H  
CLAVE3: EQU 71H  
CLAVE4: EQU 72H  
  
NUMERO1: EQU 73H  
NUMERO2: EQU 74H  
NUMERO3: EQU 75H  
NUMERO4: EQU 76H  
  
CENTENA: EQU 77H  
DECENA: EQU 78H  
UNIDAD: EQU 79H  
DATO: EQU 7AH  
COMPARADOR: EQU 7BH
```

ACTUADOR: EQU 7CH  
AUX\_DPL: EQU 7DH  
AUX\_DPH: EQU 7EH

DB7: EQU P1.7  
DB6: EQU P1.6  
DB5: EQU P1.5  
DB4: EQU P1.4  
E: EQU P1.3  
RS: EQU P1.2  
BEEP: EQU P1.1

;Matriz de caracter de 5\*7, activacion de 2 linea  
;bus de datos de 4 bits.

SET4\_BITS: EQU 28H

;Apagado y encendido de la pantalla

DISPLAY\_OFF: EQU 08H  
DISPLAY\_ON: EQU 0CH  
CURSOR\_ON: EQU 0EH  
BLINK\_ON: EQU 0FH

;Modo de entrada de los caracteres

CURSOR\_INC\_TF: EQU 06H

;Desplazamiento del cursor

CURSOR\_LEFT: EQU 10H  
CURSOR\_RIGHT: EQU 14H

;Para desplazar el cursor a la segunda linea se utiliza el  
;comando LowerLine

LOWER\_LINE: EQU 0C0H  
UPPER\_LINE: EQU 80H

MANUAL\_1: EQU 0A00H  
MANUAL\_2: EQU 0A10H  
MANUAL\_3: EQU 0A20H  
MANUAL\_4: EQU 0A30H  
MANUAL\_5: EQU 0A40H  
MANUAL\_6: EQU 0A50H  
MANUAL\_7: EQU 0A60H  
MANUAL\_8: EQU 0A70H  
MANUAL\_9: EQU 0A80H  
MANUAL\_A: EQU 0A90H  
MANUAL\_B: EQU 0AA0H  
MANUAL\_C: EQU 0AB0H  
MANUAL\_D: EQU 0AC0H  
MANUAL\_E: EQU 0AD0H  
MANUAL\_F: EQU 0AE0H

AUTOMATICO\_1: EQU 0C00H  
AUTOMATICO\_2: EQU 0C10H  
AUTOMATICO\_3: EQU 0C20H  
AUTOMATICO\_4: EQU 0C30H  
AUTOMATICO\_5: EQU 0C40H  
AUTOMATICO\_6: EQU 0C50H  
AUTOMATICO\_7: EQU 0C60H  
AUTOMATICO\_8: EQU 0C70H  
AUTOMATICO\_9: EQU 0C80H  
AUTOMATICO\_A: EQU 0C90H  
AUTOMATICO\_B: EQU 0CA0H  
AUTOMATICO\_C: EQU 0CB0H  
AUTOMATICO\_D: EQU 0CC0H  
AUTOMATICO\_E: EQU 0CD0H  
AUTOMATICO\_F: EQU 0CDEH

T\_CODIGO: EQU 0E00H  
T\_FILTRO\_1: EQU 0E10H  
T\_FILTRO\_2: EQU 0E20H  
T\_ASCII: EQU 0E30H

```

INGRESE_INI_ACT:      EQU    0E40H
INGRESE_INI_ACT_B:    EQU    0E50H

INGRESE_ACT_REB1:     EQU    0E60H
INGRESE_ACT_REB1_B:  EQU    0E70H

INGRESE_SP_T_FONDOS: EQU    0E80H
INGRESE_SP_T_FONDOS_B: EQU    0E90H

INGRESE_SP_T_PRODUC: EQU    0EA0H
INGRESE_SP_T_PRODUC_B: EQU    0EB0H

INGRESE_T_DISPARO:    EQU    0EC0H
INGRESE_T_DISPARO_B: EQU    0ED0H

INGRESE_DELAY_ACC:    EQU    0EE0H
INGRESE_DELAY_ACC_B: EQU    0EF0H

INGRESE_MIN_APER:     EQU    0F00H
INGRESE_MIN_APER_B:  EQU    0F10H

INGRESE_MAX_APER:     EQU    0F20H
INGRESE_MAX_APER_B:  EQU    0F30H

V_T_BASE:             EQU    0F40H
V_T_BASE_B:           EQU    0F50H

V_T_DALLAS_2:         EQU    0F60H
V_T_DALLAS_2_B:      EQU    0F70H

V_T_DALLAS_3:         EQU    0F80H
V_T_DALLAS_3_B:      EQU    0F90H

INGRESE_CLAVE:       EQU    0FA0H

.....
;SOFTWARE
.....

RESET: ORG    0000H
        LJM   RESET1

.....
;INTERRUPCIONES
.....

EXTI0:  ORG    0003H  ;INTERRUPCION EXTERNA 0
TIMER0: ORG    000BH  ;DESBORDAMIENTO TIMER 0
EXTI1:  ORG    0013H  ;INTERRUPCION EXTERNA 1
TIMER1: ORG    001BH  ;DESBORDAMIENTO TIMER 1
SINT:   ORG    0023H  ;PORTICO SERIAL
        LJM   INT_SERIAL
EXTI2:  ORG    002BH  ;INT.EXT 2.o DESB.TIM.2

.....
;VARIABLES
.....

;ETIQUETA      OPCODE OPERANDOS      COMENTARIO

.....
;INICIALIZACION DE MEMORIA RAM
.....
        ORG    0030H  ;INICIO DEL PROGRAMA

RESET1: MOV    P0,#0FFH      ;INICIALIZO COMO ENTRADAS

```

```

MOV P1,#0FFH ;A LOS PUERTOS
MOV P2,#0FFH ;P0, P1, P2, P3
MOV P3,#0FFH ;

CLR A
MOV IE,A
MOV IP,A
MOV TCON,A
MOV PCON,A
MOV PSW,A
MOV R0,#7FH
RESET2: MOV @R0,A
DJNZ R0,RESET2
MOV SP,#2FH

```

```

;.....
;INICIO DE PROGRAMA PRINCIPAL
;.....

```

PROGRAMA: ;Inicializamos el LCD despues enviamos el comando para que el LCD funcione con un bus de 4 líneas.

```

PROGRAMA: MOV SP_T_FONDOS,#81
MOV T_DISPARO,#73
MOV SP_T_PRODUC,#71
MOV CLAVE1,#1
MOV CLAVE2,#9
MOV CLAVE3,#7
MOV CLAVE4,#3
SETB EA
SETB ES
MOV TMOD,#20H
MOV PCON,#80H
MOV TH1,#0FDH
MOV SCON,#0D0H
SETB TR1
CLR E
MOV VAR1,#216
MOV VAR2,#36
LCALL RETARDO_2
LCALL INI4_BITS
MOV A,#SET4_BITS
LCALL COMANDO

```

;Enviamos comando para que parpadee el cursor

```

MOV A,#BLINK_ON
LCALL COMANDO

```

;Enviamos el comando para que cada vez que se ingrese un caracter se incremente la posicion del cursor y el texto quede fijo.

```

MOV A,#CURSOR_INC_TF
LCALL COMANDO

```

```

MOV AUXILIAR_DPL,#0
MOV AUXILIAR_DPH,#0CH

```

;Llamos a la subrutina CLS que es la encargada de borrar los caracteres anteriormente impresos en el LCD

```

OTRA_VEZ: LCALL CLS

```

;Para visualizar un texto completo se utiliza la subrutina LCD, necesita que se coloque el numero de caracteres del texto en el registro R3, y la base de la tabla en el DPTR; El AUXILIAR\_DPL funcionará conjuntamente con las teclas A y B para desplazarse a traves del menu. Desplazarse para abajo implica un doble incremento en Nibble; mas significativo del AUXILIAR\_DPL. Cuando llega a sus valores maximo o minimo se resetean sus valores

;Por DEFAULT se mostrara desde el primer mensaje

```

MOV     DPL,AUXILIAR_DPL
MOV     DPH,AUXILIAR_DPH
MOV     R3,#16
LCALL   LCD
MOV     A,#LOWER_LINE
LCALL   COMANDO
MOV     R3,#16
LCALL   LCD

```

;Llamamos a la subrutina teclado, la cual detecta la presion  
;de una de las teclas, y envia el codigo de la tecla presionada

```

REPITA: LCALL   CONTROL
        LCALL   TECLADO
        CJNE   A,#80H,TRANSFOR
        SJMP   REPITA

```

;Si se produjo la presion de una tecla valida, se llama a la  
;subrutina COD2HEX la cual toma el codigo de la tecla presionada  
;y segun los datos colocados anteriormente en una tabla  
;devuelve el valor del valor que esta en la parte superior de cada  
;tecla

```

TRANSFOR: LCALL   COD2HEX
          CJNE   A,#80H,DISER
          SJMP   REPITA

DISER:    CJNE   A,#0AH,QUIZA_B
          LJMP   ES_A
QUIZA_B:  CJNE   A,#0BH,QUIZA_C
          LJMP   ES_B
QUIZA_C:  CJNE   A,#0CH,QUIZA_D
          LJMP   ES_C
QUIZA_D:  CJNE   A,#0DH,QUIZA_E
          LJMP   ES_D
QUIZA_E:  CJNE   A,#0EH,QUIZA_F
          LJMP   ES_E
QUIZA_F:  CJNE   A,#0FH,UN_NUMERO
          LJMP   ES_F
UN_NUMERO: SWAP   A
          MOV    PARTE_A,A

```

```

B_REPITA: LCALL   CONTROL
          LCALL   TECLADO
          CJNE   A,#80H,B_TRANSFOR
          SJMP   B_REPITA

```

```

B_TRANSFOR: LCALL   COD2HEX
           CJNE   A,#80H,B_DISER
           LJMP   B_REPITA

```

```

B_DISER:  CJNE   A,#0AH,MAYBE_B
          LJMP   B_REPITA
MAYBE_B:  CJNE   A,#0BH,MAYBE_C
          LJMP   B_REPITA
MAYBE_C:  CJNE   A,#0CH,MAYBE_D
          LJMP   REPITA
MAYBE_D:  CJNE   A,#0DH,MAYBE_E
          LJMP   B_REPITA
MAYBE_E:  CJNE   A,#0EH,MAYBE_F
          LJMP   B_REPITA
MAYBE_F:  CJNE   A,#0FH,DOS_NUMERO
          LJMP   B_REPITA
DOS_NUMERO: MOV    PARTE_B,A
          ORL   A,PORTE_A
          MOV   OPCION,A

```

```

ES_01:   CJNE   A,#01H,ES_02

```

;Verificamos si esta en manual o en automatico.

```

MOV     A,AUTO_MANUAL
CJNE   A,#0,ES_01_MANUAL

```

;La opcion 1 en modo automatico sirve para ingresar el valor con el que  
;va iniciar la resistencia de calentamiento despues de que detecte  
;la ebullición

```

MOV     COMANDOS,#2DH
MOV     DIRECCION,#DIREC_1
MOV     DPTR,#INGRESE_INI_ACT
LCALL  INGRESE_DATO
MOV     POTEN_INICIAL,DATO_INGRESADO
LJMP   OTRA_VEZ

```

;La opción 1 en el modo manual sirve para ingresar el % de potencia que  
;el usuario desee para que trabaje la resistencia.

```

ES_01_MANUAL: MOV     COMANDOS,#11H
MOV     DIRECCION,#DIREC_1
MOV     DPTR,#INGRESE_ACT_REB1
LCALL  INGRESE_DATO
MOV     ACT_REBOILER1,DATO_INGRESADO
LJMP   OTRA_VEZ

```

```
ES_02:  CJNE   A,#02H,ES_03
```

;Verificamos si esta en manual o automatico

```

MOV     A,AUTO_MANUAL
CJNE   A,#0,ES_02_MANUAL

```

;La opción 2 en el modo automático sirve para ingresar el valor del SP  
;de temperatura de FONDOS

```

MOV     COMANDOS,#33H
MOV     DIRECCION,#DIREC_1
MOV     DPTR,#INGRESE_SP_T_FONDOS
LCALL  INGRESE_DATO
MOV     SP_T_FONDOS,DATO_INGRESADO
LJMP   OTRA_VEZ

```

;La opción 2 en el modo manual sirve para visualizar la temperatura  
;dentro del REBOILER1.

```

ES_02_MANUAL: MOV     DIRECCION,#DIREC_1
MOV     DPTR,#V_T_BASE
MOV     AUX_DPL,DPL
MOV     AUX_DPH,DPH
LCALL  VER_TEMPERATURA
LJMP   OTRA_VEZ

```

```
ES_03:  CJNE   A,#03H,ES_04
```

;Verificamos si esta en manual o en automatico

```

MOV     A,AUTO_MANUAL
CJNE   A,#0,ES_03_MANUAL

```

;La opción 3 en el modo automático sirve para ingresar el set point  
;de temperatura de los productos

```

MOV     COMANDOS,#4BH
MOV     DIRECCION,#DIREC_1
MOV     DPTR,#INGRESE_SP_T_PRODUC
LCALL  INGRESE_DATO
MOV     SP_T_PRODUC,DATO_INGRESADO
LJMP   OTRA_VEZ

```

;La opción 3 en el modo manual sirve para ver la temperatura del Dallas2

```

ES_03_MANUAL: MOV     DIRECCION,#DIREC_1
MOV     DPTR,#V_T_DALLAS_2
MOV     AUX_DPL,DPL
MOV     AUX_DPH,DPH
LCALL  VER_TEMPERATURA
LJMP   OTRA_VEZ

```

ES\_04: CJNE A,#04H,ES\_05

;Verificamos si esta en manual o automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_04_MANUAL
```

;La opción 4 en el modo automático sirve para ingresar  
;de temperatura de disparo. Es decir cuando la temperatura en el  
;Dallas 2 alcanza el valor que en esta opción se ha colocado.  
;Automáticamente coloca la potencia de la resistencia al valor  
;que se ingreso en la variable POTEN\_INICIAL.

```
MOV COMANDOS,#55H
MOV DIRECCION,#DIREC_1
MOV DPTR,#INGRESE_T_DISPARO
LCALL INGRESE_DATO
MOV T_DISPARO,DATO_INGRESADO

LJMP OTRA_VEZ
```

;La opción 4 en el modo manual sirve para ver la temperatura en el Dallas 3

```
ES_04_MANUAL: MOV DIRECCION,#DIREC_1
MOV DPTR,#V_T_DALLAS_3
MOV AUX_DPL,DPL
MOV AUX_DPH,DPH
LCALL VER_TEMPERATURA
LJMP OTRA_VEZ
```

;La opción 5 servira para ver la lectura del Dallas1  
;tanto en manual como en automático.

ES\_05: CJNE A,#05H,ES\_06

;Verificamos si esta en manual o en automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_05_MANUAL
```

;La opción 5 en el modo automático sirve para ver la temperatura dentro del  
;reboiler 1.

```
MOV DIRECCION,#DIREC_1
MOV DPTR,#V_T_BASE
MOV AUX_DPL,DPL
MOV AUX_DPH,DPH
LCALL VER_TEMPERATURA

LJMP OTRA_VEZ
```

;La opción 5 en modo manual sirve para encender o apagar el REFLUJO

```
ES_05_MANUAL: SETB TB8
MOV A,#DIREC_2
LCALL ENVIAR
CLR TB8
MOV A,#69H
LCALL ENVIAR
MOV A,#11H
LCALL ENVIAR
LJMP OTRA_VEZ
```

ES\_06: CJNE A,#06H,ES\_07

;Verificamos si esta en manual o en automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_06_MANUAL
```

;La opción 6 en modo automatico sirve para ver la temperatura del DALLAS 2

```
MOV DIRECCION,#DIREC_1
```

```

MOV    DPTR,#V_T_DALLAS_2
MOV    AUX_DPL,DPL
MOV    AUX_DPH,DPH
LCALL  VER_TEMPERATURA

LJMP   OTRA_VEZ

```

;La opcion 6 en el modo manual sirve para encender o apagar la COSECHA

```

ES_06_MANUAL: SETB  TB8
MOV     A,#DIREC_2
LCALL  ENVIAR
CLR     TB8
MOV     A,#69H
LCALL  ENVIAR
MOV     A,#2DH
LCALL  ENVIAR
LJMP   OTRA_VEZ

```

```
ES_07:  CJNE  A,#07H,ES_08
```

;Verificamos si esta en manual o en automatico

```

MOV     A,AUTO_MANUAL
CJNE    A,#0,ES_07_MANUAL

```

;La opción 7 en el modo automático sirve para ver la temperatura del DALLAS 3

```

MOV     DIRECCION,#DIREC_1
MOV     DPTR,#V_T_DALLAS_3
MOV     AUX_DPL,DPL
MOV     AUX_DPH,DPH
LCALL  VER_TEMPERATURA

LJMP   OTRA_VEZ

```

;La opción 7 en el modo manual sirve para encender o apagar la ALIMENTACION

```

ES_07_MANUAL: SETB  TB8
MOV     A,#DIREC_2
LCALL  ENVIAR
CLR     TB8
MOV     A,#69H
LCALL  ENVIAR
MOV     A,#33H
LCALL  ENVIAR
LJMP   OTRA_VEZ

```

```
ES_08:  CJNE  A,#08H,ES_09
```

;Verificamos si esta en manual o en automatico

```

MOV     A,AUTO_MANUAL
CJNE    A,#0,ES_08_MANUAL

```

;La opcion 8 en el modo automático sirve para ingresar el tiempo de retardo  
;del encendido de la bomba de cosecha.

```

MOV     COMANDOS,#88H
MOV     DIRECCION,#DIREC_2
MOV     DPTR,#INGRESE_DELAY_ACC
LCALL  INGRESE_DATO
MOV     DELAY_ACC,DATO_INGRESADO

LJMP   OTRA_VEZ

```

;La opción 8 en el modo manual sirve para encender o apagar el VENTILADOR

```

ES_08_MANUAL: SETB  TB8
MOV     A,#DIREC_2
LCALL  ENVIAR
CLR     TB8
MOV     A,#69H

```

```
LCALL ENVIAR
MOV A,#4BH
LCALL ENVIAR
LJMP OTRA_VEZ
```

```
ES_09: CJNE A,#09H,ES_10
```

;Verificamos si esta en manual o en automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_09_MANUAL
```

;La opción 9 en el modo automático sirve para ingresar el minimo valor que se debe  
;ABRIR la válvula.

```
MOV COMANDOS,#77H
MOV DIRECCION,#DIREC_3
MOV DPTR,#INGRESE_MIN_APER
LCALL INGRESE_DATO
MOV MIN_ACT_MOTOR,DATO_INGRESADO
LJMP OTRA_VEZ
```

;La opción 9 en el modo manual sirve para abrir la válvula 10 grados

```
ES_09_MANUAL: SETB TB8
MOV A,#DIREC_3
LCALL ENVIAR
CLR TB8
MOV A,#69H
LCALL ENVIAR
MOV A,#11H
LCALL ENVIAR
LJMP OTRA_VEZ
```

```
ES_10: CJNE A,#10H,ES_11
```

;Verificamos si esta en manual o en automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_10_MANUAL
```

;La opción 10 en el modo automático sirve para ingresar el valor maximo  
;de apertura de la valvula.

```
MOV COMANDOS,#88H
MOV DIRECCION,#DIREC_3
MOV DPTR,#INGRESE_MAX_APER
LCALL INGRESE_DATO
MOV MAX_ACT_MOTOR,DATO_INGRESADO
```

```
LJMP OTRA_VEZ
```

;En el modo manual la opción 10 sirve para CERRAR la valvula 10 grados

```
ES_10_MANUAL: SETB TB8
MOV A,#DIREC_3
LCALL ENVIAR
CLR TB8
MOV A,#69H
LCALL ENVIAR
MOV A,#2DH
LCALL ENVIAR
LJMP OTRA_VEZ
```

```
ES_11: CJNE A,#11H,ES_12
```

;Verificamos si esta en manual o en automatico

```
MOV A,AUTO_MANUAL
CJNE A,#0,ES_11_MANUAL
```

;La opción 11 en el modo automatico sirve para encender o apagar  
;el VENTILADOR.

```

SETB  TB8
MOV    A,#DIREC_2
LCALL ENVIAR
CLR    TB8
MOV    A,#77H
LCALL ENVIAR
LJMP  OTRA_VEZ

```

;La opcion 11 en el modo manual sirve para abrir por completo la valvula

```

ES_11_MANUAL: SETB  TB8
MOV    A,#DIREC_3
LCALL ENVIAR
CLR    TB8
MOV    A,#69H
LCALL ENVIAR
MOV    A,#33H
LCALL ENVIAR
LJMP  OTRA_VEZ

```

```
ES_12: CJNE  A,#12H,ES_13
```

;Verificamos si esta en manual o en automatico

```

MOV    A,AUTO_MANUAL
CJNE  A,#0,ES_12_MANUAL

LJMP  OTRA_VEZ

```

;La opcion 12 en el modo MANUAL sirve para detener el cierre o apertura de la  
;válvula.

```

ES_12_MANUAL: SETB  TB8
MOV    A,#DIREC_3
LCALL ENVIAR
CLR    TB8
MOV    A,#69H
LCALL ENVIAR
MOV    A,#55H
LCALL ENVIAR
LJMP  OTRA_VEZ

```

```
ES_13: CJNE  A,#13H,ES_14
```

;Verificamos si esta en manual o en automatico

```

MOV    A,AUTO_MANUAL
CJNE  A,#0,ES_13_MANUAL
LJMP  OTRA_VEZ

```

;La opcion 13 en el modo MANUAL sirve para CERRAR por completo  
;la válvula.

```

ES_13_MANUAL: SETB  TB8
MOV    A,#DIREC_3
LCALL ENVIAR
CLR    TB8
MOV    A,#69H
LCALL ENVIAR
MOV    A,#4BH
LCALL ENVIAR
LJMP  OTRA_VEZ

```

```
ES_14: CJNE  A,#14H,ES_15
```

;Verificamos si esta en manual o en automatico

```

MOV    A,AUTO_MANUAL
CJNE  A,#0,ES_14_MANUAL
LJMP  OTRA_VEZ

```

```
ES_14_MANUAL: LJMP  OTRA_VEZ
```

```
ES_15: CJNE  A,#15H,ES_A
```

;Verificamos si esta en manual o en automatico

```

MOV     A,AUTO_MANUAL
CJNE   A,#0,ES_15_MANUAL
LJMP   OTRA_VEZ

ES_15_MANUAL: LJMP   OTRA_VEZ

ES_A:   CJNE   A,#0AH,ES_B
ES_B:   CJNE   A,#0BH,ES_C

ES_C:   CJNE   A,#0CH,ES_D
        LJMP   REPITA

ES_D:   CJNE   A,#0DH,ES_E
;Verificamos si esta en manual o en automatico
MOV     A,AUTO_MANUAL
CJNE   A,#0,ES_D_MANUAL
LCALL  CLAVE
LJMP   OTRA_VEZ

;Si se selecciona la tecla D cuando el menu esta en MANUAL
;se regresa automaticamente a AUTOMATICO.

ES_D_MANUAL: MOV     AUTO_MANUAL,#0
              MOV     AUXILIAR_DPL,#0
              MOV     AUXILIAR_DPH,#0CH
;Enviamos la direccion del acumulador y enviamos el numero
;0FFh se tiene hacer esto con todos los dispositivos.

        SETB    TB8
        MOV     A,#DIREC_1
        LCALL  ENVIAR
        CLR     TB8
        MOV     A,#0F0H
        LCALL  ENVIAR

        SETB    TB8
        MOV     A,#DIREC_2
        LCALL  ENVIAR
        CLR     TB8
        MOV     A,#0F0H
        LCALL  ENVIAR

        SETB    TB8
        MOV     A,#DIREC_3
        LCALL  ENVIAR
        CLR     TB8
        MOV     A,#0F0H
        LCALL  ENVIAR
        LJMP   OTRA_VEZ

ES_E:   CJNE   A,#0EH,ES_F
        LCALL  SUBIR_MENU
        LJMP   REPITA

ES_F:   CJNE   A,#0FH,BUCLE
        LCALL  BAJAR_MENU
        LJMP   REPITA

BUCLE:  LJMP   REPITA

CLAVE:  LCALL  CLS
        MOV     DPTR,#INGRESE_CLAVE
        MOV     R3,#16
        LCALL  LCD

;Bajamos una linea para ingresar la clave

        MOV     A,#LOWER_LINE
        LCALL  COMANDO

```

;Vamos a permitir el ingreso de 4 numeros

```
MOV R7,#4
MOV R0,#NUMERO1
CLAVE_REPITA: LCALL TECLADO
CJNE A,#80H,CLAVE_TRANSFOR
LJMP CLAVE_REPITA

CLAVE_TRANSFOR: LCALL COD2HEX
CJNE A,#80H,CLAVE_DISER
LJMP CLAVE_REPITA

CLAVE_DISER: CJNE A,#0CH,CLAVE_GUARDAR
LJMP CLAVE_SALIR

CLAVE_GUARDAR: MOV @R0,A
MOV A,#02AH
LCALL CARACTER
INC R0
DJNZ R7,CLAVE_REPITA

MOV A,NUMERO1
CJNE A,CLAVE1,CLAVE
MOV A,NUMERO2
CJNE A,CLAVE2,CLAVE
MOV A,NUMERO3
CJNE A,CLAVE3,CLAVE
MOV A,NUMERO4
CJNE A,CLAVE4,CLAVE
MOV AUXILIAR_DPH,#0AH
MOV AUXILIAR_DPL,#0
MOV AUTO_MANUAL,#0FFH
```

;Debemos informar a todos los esclavos que estamos en modo  
;MANUAL y eso se lo realiza enviando el comando 00 a todos

```
SETB TB8
MOV A,#0FH
LCALL ENVIAR

CLAVE_SALIR: RET
```

VER\_TEMPERATURA:

;Borramos el LCD y mostramos el SetPoint y la temperatura  
;actual.

```
LCALL CLS
MOV DPL,AUX_DPL
MOV DPH,AUX_DPH
MOV R3,#12
LCALL LCD

;Convertimos de HEX a BCD
MOV DATO,ACTUADOR
LCALL HEX2BCD
;Visualizamos las unidades y las decenas

MOV A,CENTENA
CJNE A,#0,V_T_ESUNO
MOV A,#20H
LCALL CARACTER
SJMP V_T_SEGUIR

V_T_ESUNO: LCALL HEX2ASCII
LCALL CARACTER
V_T_SEGUIR: MOV A,DECENA
LCALL HEX2ASCII
LCALL CARACTER

MOV A,UNIDAD
LCALL HEX2ASCII
LCALL CARACTER
```

;Visualizamos el signo porcentaje

```
MOV    A,#37
LCALL  CHARACTER

MOV    A,#LOWER_LINE
LCALL  COMANDO

MOV    DPL,AUX_DPL
MOV    DPH,AUX_DPH
MOV    A,DPL
SWAP   A
INC    ACC
SWAP   A
MOV    DPL,A

MOV    R3,#12
LCALL  LCD
```

V\_T\_ANOTHER: MOV BAN,#0

```
MOV    A,AUTO_MANUAL
CJNE   A,#0,V_T_MANUAL
```

```
MOV    A,OPCION
CJNE   A,#05H,V_T_A_DS2
MOV    DATO,MEDIDOR_1
LJMP   V_T_VISUA
```

V\_T\_A\_DS2: CJNE A,#06H,V\_T\_A\_DS3  
MOV DATO,MEDIDOR\_2  
LJMP V\_T\_VISUA

V\_T\_A\_DS3: CJNE A,#07H,V\_T\_VISUA  
MOV DATO,MEDIDOR\_3  
LJMP V\_T\_VISUA

```
V_T_MANUAL: MOV    A,OPCION
CJNE   A,#02H,V_T_M_DS2
MOV    DATO,MEDIDOR_1
LJMP   V_T_VISUA
```

V\_T\_M\_DS2: CJNE A,#03H,V\_T\_M\_DS3  
MOV DATO,MEDIDOR\_2  
LJMP V\_T\_VISUA

V\_T\_M\_DS3: CJNE A,#04H,V\_T\_VISUA  
MOV DATO,MEDIDOR\_3

;Nos regresamos 3 espacios a la izquierda y borramos enviando a  
;imprimir espacios. Despues nos volvemos a regresar 2 espacios para  
;poder imprimir los datos que recibimos del dispositivo esclavo.  
;Visualizamos el valor que recibimos

V\_T\_VISUA: LCALL HEX2BCD

```
MOV    A,CENTENA
CJNE   A,#1,DEJAR_ESPACIO
LCALL  HEX2ASCII
LCALL  CHARACTER
SJMP   IMPRIME_CENTENA
DEJAR_ESPACIO: MOV    A,#20H
LCALL  CHARACTER
IMPRIME_CENTENA: MOV    A,DECENA
LCALL  HEX2ASCII
LCALL  CHARACTER
```

```
MOV    A,UNIDAD
LCALL  HEX2ASCII
LCALL  CHARACTER
```

;Estamos sensando continuamente el teclado para detectar la  
;presión de la tecla "C" de no suceder esto, se producen  
;múltiples interrupciones debido al dispositivo esclavo.

;En la subrutina de interrupción lo que se hace es visualizar  
;el dato enviado por el dispositivo esclavo.

```
V_T_TECLADO:  LCALL  TECLADO
              CJNE   A,#80H,V_T_ESCANEAR
              LJMP   V_T_SALTO
```

```
V_T_ESCANEAR: CJNE   A,#0EDH,V_T_SALTO
              LJMP   V_T_SALIR
```

```
V_T_SALTO:   MOV    A,AUTO_MANUAL
              LCALL  CONTROL
```

```
V_T_BAN:     MOV    A,BAN
              JZ     V_T_TECLADO
              LJMP  VER_TEMPERATURA
```

```
V_T_SALIR:   MOV    BAN,#0
              RET
```

;ooooooooo el programa se cuelga porque que tenemos que POPear al  
;inverso de lo que se PUSHea

```
INT_SERIAL:  CLR    EA
              CLR    RI
              PUSH   ACC
              PUSH   PSW

              MOV    A,SBUF
              CJNE   A,#0EEH,INT_SER_SALIR
```

```
EPOH:       JNB    RI,EPOH
              CLR    RI
              MOV    MEDIDOR_1,SBUF
```

```
TIAW:       JNB    RI,TIAW
              CLR    RI
              MOV    MEDIDOR_2,SBUF
```

```
RUDE:       JNB    RI,RUDE
              CLR    RI
              MOV    MEDIDOR_3,SBUF
```

```
WASTE:     JNB    RI,WASTE
              CLR    RI
              MOV    ACTUADOR,SBUF
```

```
INT_SER_SALIR: MOV    BAN,#0FFH
               POP    PSW
               POP    ACC
               SETB   EA
```

RETI

```
CONTROL:    MOV    A,AUTO_MANUAL
              CJNE   A,#0,CON_SALIR_1
              LJMP   CON_COMPARAR
```

```
CON_SALIR_1: LJMP   CON_SALIR
```

```
CON_COMPARAR:MOV    A,MEDIDOR_1
              CJNE   A,SP_T_FONDOS,CON_FON_MAYMEN
```

```
CON_FON_MAYOR: MOV    A,BANDERA_1
```

```
              CJNE   A,#0,CON_VENTI
```

```
              SETB   TB8
```

```
              MOV    A,#DIREC_3
```

```
              LCALL  ENVIAR
```

```
              CLR    TB8
```

```
              MOV    A,#96H
```

```
              LCALL  ENVIAR
```

```
              MOV    BANDERA_1,#0FFH
```

```
              LJMP   CON_VENTI
```

```
CON_FON_MAYMEN: JNC    CON_FON_MAYOR
                MOV    A,BANDERA_1
```

```

CJNE  A,#0FFH,CON_VENTI
SETB  TB8
MOV   A,#DIREC_3
LCALL ENVIAR
CLR   TB8
MOV   A,#0AAH
LCALL ENVIAR
MOV   BANDERA_1,#0

```

```

CON_VENTI:  MOV   A,MEDIDOR_2
CJNE  A,#50,CON_VENTI_MAYMEN
CON_VENTI_MAYOR:  MOV   A,BANDERA_2
CJNE  A,#0,CON_SALIR
SETB  TB8
MOV   A,#DIREC_2
LCALL ENVIAR
CLR   TB8
MOV   A,#77H
LCALL ENVIAR
MOV   BANDERA_2,#0FFH
LJMP  CON_SALIR

```

```

CON_VENTI_MAYMEN:  JNC   CON_VENTI_MAYOR
MOV   A,BANDERA_2
CJNE  A,#0FFH,CON_SALIR
SETB  TB8
MOV   A,#DIREC_2
LCALL ENVIAR
CLR   TB8
MOV   A,#77H
LCALL ENVIAR
MOV   BANDERA_2,#0
CON_SALIR:  RET

```

#### BAJAR\_MENU:

;Cada vez que se presiona la tecla abajo lo primero que se hace es  
;borrar el LCD. Movemos el AUXILIAR\_DPL al R7 pues de esa forma lo  
;podemos comparar ya que el CJNE solo funciona con registros y el ACC.

```

LCALL CLS
MOV   R7,AUXILIAR_DPL
CJNE  R7,#0E0H,B_M_ROTATE

```

;Si llega al límite superior que es 0E0H lo primero que se hace es  
;cargarle con el valor inferior de la tabla que es 00h.

```

MOV   AUXILIAR_DPL,#00H
LJMP  B_M_MUESTRE

```

;Si no esta en el valor límite superior lo que se hace es mover el  
;AUXILIAR\_DPL al ACC. Esto para poder aumentar en 2 el nibble mas  
;significativo.

```

B_M_ROTATE:  MOV   A,AUXILIAR_DPL
SWAP  A
INC   A
INC   A
SWAP  A
MOV   AUXILIAR_DPL,A

```

;Aqui es donde se envia al LCD los datos, es importante aclarar que  
;la base del mensaje va en el DPTR y el numero de caracteres va en R3

```

B_M_MUESTRE:  MOV   DPL,AUXILIAR_DPL
MOV   DPH,AUXILIAR_DPH
MOV   R3,#16
LCALL LCD
MOV   A,#LOWER_LINE
LCALL COMANDO
MOV   R3,#16
LCALL LCD

```

```

RET

```

SUBIR\_MENU:

;Cada vez que se presiona la tecla arriba lo primero que se hace es  
;borrar el LCD. Movemos el AUXILIAR\_DPL al R7 pues de esa forma lo  
;podemos comparar ya que el CJNE solo funciona con registros y el ACC.

```
LCALL CLS
MOV R7,AUXILIAR_DPL
CJNE R7,#00H,S_M_ROTAR
```

;Si llega al limite inferior que es cero lo primero que se hace es  
;cargarle con el valor superior de la tabla menos 16 lo que nos da  
;0E0h, es decir cargandole con el valor del penultimo mensaje.

```
MOV AUXILIAR_DPL,#0E0H
LJMP S_M_MUESTRE
```

;Si no esta en el valor limite inferior lo que se hace es mover el  
;AUXILIAR\_DPL al ACC. Esto para poder decrementar en 2 el nibble mas  
;significativo.

```
S_M_ROTAR: MOV A,AUXILIAR_DPL
SWAP A
DEC A
DEC A
SWAP A
MOV AUXILIAR_DPL,A
```

;Aqui es donde se envia al LCD los datos, es importante aclarar que  
;la base del mensaje va en el DPTR y el numero de caracteres va en R3

```
S_M_MUESTRE: MOV DPL,AUXILIAR_DPL
MOV DPH,AUXILIAR_DPH
MOV R3,#16
LCALL LCD
MOV A,#LOWER_LINE
LCALL COMANDO
MOV R3,#16
LCALL LCD
RET
```

INGRESE\_DATO: LCALL CLS

;Enviamos el mensaje al LCD "P% R BASE "

```
MOV R3,#16
LCALL LCD
MOV A,#LOWER_LINE
LCALL COMANDO
```

;Enviamos el mensaje al LCD "(0%-100%)="

```
MOV R3,#11
LCALL LCD
```

;Permitimos el ingreso de 3 numeros y una tecla de funcion.  
;Los dos primeros pueden estar en  
;rango entre 0 y 9 y la tercero es una tecla de función. Como puede ser  
; A.- Aceptar  
; B.- Borrar  
; C.- Cancelar

I\_D\_BORRADO: MOV R7,#04

;Tener cuidado el valor que se le ingresa porque solo estamos  
;usando decenas y unidades.

```
MOV R0,#CENTENA
```

;Primero esperamos que el usuario halla ingresado el primer numero  
;para de esa forma poder utilizar el uno u el otro filtro.  
;Sabemos que si el primer numero en ingresarse es el 1 por obvias razones  
;los siguientes dos numeros tendran que ser 0 0.

```
I_D_FILTRAR: CJNE R7,#04H,I_D_NORMAL
SJMP I_D_PRIMERO
```

;Comparamos si el valor ingresado por el usuario es 1 o 0  
;segun eso ocupamos el uno u el otro filtro.

```
I_D_NORMAL:    MOV     A,CENTENA
               CJNE   A,#1,I_D_PRIMERO
               MOV    DPTR,#T_FILTRO_2
               SJMP  I_D_GO_AHEAD

I_D_PRIMERO:   MOV    DPTR,#T_FILTRO_1
I_D_GO_AHEAD: MOV    A,R7
               MOVC  A,@A+DPTR
               MOV   FILTRO,A

I_D_ANOTHER:  LCALL  TECLADO
               CJNE  A,#80H,I_D_CONVERTIR
               SJMP  I_D_ANOTHER
I_D_CONVERTIR: LCALL  COD2HEX
               CJNE  A,#80H,I_D_ULTIMA_TECLA
               SJMP  I_D_ANOTHER
I_D_ULTIMA_TECLA: CJNE  R7,#01H,I_D_ESCANEO
               SJMP  I_D_ES_LA_A
I_D_ESCANEO:  CJNE  A,#00H,I_D_ES_UNO
               LJMP  I_D_FILTRAJE
I_D_ES_UNO:   CJNE  A,#01H,I_D_ES_DOS
               LJMP  I_D_FILTRAJE
I_D_ES_DOS:   CJNE  A,#02H,I_D_ES_TRES
               LJMP  I_D_FILTRAJE
I_D_ES_TRES:  CJNE  A,#03H,I_D_ES_CUATRO
               LJMP  I_D_FILTRAJE
I_D_ES_CUATRO: CJNE  A,#04H,I_D_ES_CINCO
               LJMP  I_D_FILTRAJE
I_D_ES_CINCO: CJNE  A,#05H,I_D_ES_SEIS
               LJMP  I_D_FILTRAJE
I_D_ES_SEIS:  CJNE  A,#06H,I_D_ES_SIETE
               LJMP  I_D_FILTRAJE
I_D_ES_SIETE: CJNE  A,#07H,I_D_ES_OCHO
               LJMP  I_D_FILTRAJE
I_D_ES_OCHO:  CJNE  A,#08H,I_D_ES_NUEVE
               LJMP  I_D_FILTRAJE
I_D_ES_NUEVE: CJNE  A,#09H,I_D_ES_LA_A
               LJMP  I_D_FILTRAJE
I_D_ES_LA_A:  CJNE  A,#0AH,I_D_ES_LA_B
```

;En caso de que no se hallan ingresado los 3 numeros, cancela  
;la pulsacion de la tecla aceptar y va de nuevo a escanear el  
;teclado

```
               CJNE  R7,#01,I_D_ANOTHER
               LCALL  ENVIAR_DATO
               LJMP  I_D_SALIDAS
```

;Si la tecla presionada es borrar, borra todos los numeros ingresados

```
I_D_ES_LA_B:  CJNE  A,#0BH,I_D_ES_LA_C

               LCALL  BORRAR
               LJMP  I_D_BORRADO
```

;Si la tecla presionada es la C se sale de la subrutina

```
I_D_ES_LA_C:  CJNE  A,#0CH,I_D_ES_LA_D
               SJMP  I_D_SALIDAS
I_D_ES_LA_D:  CJNE  A,#0DH,I_D_ES_LA_E
I_D_ES_LA_E:  CJNE  A,#0EH,I_D_ES_LA_F
I_D_ES_LA_F:  CJNE  A,#0FH,I_D_ANOTHER

I_D_FILTRAJE: MOV    R6,FILTRO
               MOV   COMPARADOR,#00H
I_D_COMPARALO: CJNE  A,COMPARADOR,I_D_OTRO_NUM
               SJMP  I_D_GUARDALO
I_D_OTRO_NUM: INC    COMPARADOR
               DJNZ  R6,I_D_COMPARALO
```

```
LJMP I_D_ANOTHER
```

;Una vez que es guardado el numero cambiamos el valor del filtro

```
I_D_GUARDALO: MOV @R0,A
               LCALL HEX2ASCII
               LCALL CARACTER
               INC R0
               DJNZ R7,I_D_FIL_TRAR
               SJMP I_D_SALIDAS
I_D_FIL_TRAR: LJMP I_D_FILTRAR
I_D_SALIDAS: RET
```

```
ENVIAR_DATO: LCALL BCD2HEX
             MOV DATO_INGRESADO,DATO
```

;Enviamos la direccion del dispositivo

```
SETB TB8
MOV A,DIRECCION
LCALL ENVIAR
```

;Despues enviamos el SET POINT

```
CLR TB8
MOV A,COMANDOS
LCALL ENVIAR
```

```
CLR TB8
MOV A,DATO_INGRESADO
LCALL ENVIAR
RET
```

```
ENVIAR: CLR EA
        CLR TI
        MOV SBUF,A
```

```
ENV_ESPERAR: JNB TI,ENV_ESPERAR
             CLR TI
             SETB EA
             RET
```

```
BCD2HEX: MOV A,CENTENA
         MOV B,#100
         MUL AB
         MOV DATO,A
         MOV A,DECENA
         MOV B,#10
         MUL AB
         ADD A,DATO
         MOV DATO,A
         MOV A,UNIDAD
         ADD A,DATO
         MOV DATO,A
```

```
RET
```

```
HEX2ASCII: MOV DPTR,#T_ASCII
           MOVC A,@A+DPTR
           RET
```

;Si ingreso un solo numero tenemos que borrarlo e incrementar  
;R7 en una unidad.

```
BORRAR: CJNE R7,#04,BOR_INGRESO_UNO
        LJMP BOR_QUITER
BOR_INGRESO_UNO: CJNE R7,#03,BOR_INGRESO_DOS
               MOV A,#CURSOR_LEFT
               LCALL COMANDO
               MOV A,#20H
               LCALL CARACTER
```

```

MOV     A,#CURSOR_LEFT
LCALL  COMANDO
LJMP   BOR_QUITER

BOR_INGRESO_DOS:    CJNE  R7,#02,BOR_INGRESO_TRES
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#20H
LCALL  CHARACTER
MOV     A,#20H
LCALL  CHARACTER
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
LJMP   BOR_QUITER

BOR_INGRESO_TRES:  CJNE  R7,#01,BOR_QUITER
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#20H
LCALL  CHARACTER
MOV     A,#20H
LCALL  CHARACTER
MOV     A,#20H
LCALL  CHARACTER
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO
MOV     A,#CURSOR_LEFT
LCALL  COMANDO

BOR_QUITER:      RET

TECLADO:         MOV     R2,#04
MOV     A,#7FH
TEC_OTRA:        MOV     P2,A
CJNE   A,P2,TEC_TECLA
SJMP   TEC_SIGUE
TEC_TECLA:       MOV     VAR1,#72
MOV     VAR2,#180
LCALL  RETARDO_2
CJNE   A,P2,TEC_GUAR
SJMP   TEC_SIGUE
TEC_GUAR:        MOV     A,P2
LCALL  BEEPER
SJMP   TEC_SALIR
TEC_SIGUE:       RR      A
DJNZ   R2,TEC_OTRA
MOV     A,#80H
TEC_SALIR:      RET

BEEPER:CLR       BEEP
;Retardo de 35ms
MOV     VAR1,#189
MOV     VAR2,#63

LCALL  RETARDO_2
SETB   BEEP
RET

```

```

COD2HEX:    MOV    DPTR,#T_CODIGO
            MOV    R2,#16
            MOV    CODIGO_HEX,A
COD2_OTRO:  MOV    A,#00H
            MOVC   A,@A+DPTR
            CJNE   A,CODIGO_HEX,COD2_SIGUE
            MOV    A,DPL
            MOV    CODIGO_HEX,A
            SJMP   COD2_SALIR
COD2_SIGUE: INC    DPTR
            DJNZ   R2,COD2_OTRO
            MOV    A,#80H
COD2_SALIR: RET

```

```

RETARDO_1:  DJNZ   VAR1,RETARDO_1
RET

```

```

RETARDO_2:  DJNZ   VAR1,RETARDO_2
            DJNZ   VAR2,RETARDO_2
RET

```

```

RETARDO_3:  DJNZ   VAR1,RETARDO_3
            DJNZ   VAR2,RETARDO_3
            DJNZ   VAR3,RETARDO_3
RET

```

;Ojo quite dos nOP que van entre la activacion y desactivacion del enable.

```

INI4_BITS:  MOV    R2,#3
            CLR    RS
INI4_REPITA: CLR    DB7
            CLR    DB6
            SETB   DB5
            SETB   DB4
            SETB   E
            CLR    E
            MOV    VAR1,#189
            MOV    VAR2,#11
            LCALL  RETARDO_2
            DJNZ   R2,INI4_REPITA
            CLR    DB7
            CLR    DB6
            SETB   DB5
            CLR    DB4
            SETB   E
            CLR    E
            MOV    VAR1,#189
            MOV    VAR2,#11
            LCALL  RETARDO_2
RET

```

```

CARACTER:   SETB   RS
            MOV    R2,#2
CAR_REPITA: RLC    A
            MOV    DB7,C
            RLC    A
            MOV    DB6,C
            RLC    A
            MOV    DB5,C
            RLC    A
            MOV    DB4,C
            SETB   E
            CLR    E
            MOV    VAR1,#61
            LCALL  RETARDO_1
            DJNZ   R2,CAR_REPITA
RET

```

```

COMANDO:    CLR    RS
            MOV    R2,#2
COM_REPITA: RLC    A
            MOV    DB7,C
            RLC    A
            MOV    DB6,C
            RLC    A
            MOV    DB5,C
            RLC    A
            MOV    DB4,C
            SETB   E
            CLR    E
            MOV    VAR1,#24
            LCALL  RETARDO_1
            DJNZ   R2,COM_REPITA
RET

```

```

LCD:    MOV    A,#00H
        MOVC   A,@A+DPTR
        LCALL  CHARACTER
        INC    DPTR
        DJNZ   R3,LCD
RET

```

```

CLS:    MOV    A,#1
        CLR    RS
        MOV    R2,#2
CLS_REPITA: RLC    A
        MOV    DB7,C
        RLC    A
        MOV    DB6,C
        RLC    A
        MOV    DB5,C
        RLC    A
        MOV    DB4,C
        SETB   E
        CLR    E
        MOV    VAR1,#145
        MOV    VAR2,#4
        LCALL  RETARDO_2
        DJNZ   R2,CLS_REPITA
RET

```

```

HEX2BCD: MOV    A,DATO
        MOV    B,#0AH
        DIV    AB
        MOV    UNIDAD,B
        MOV    B,#0AH
        DIV    AB
        MOV    DECENA,B
        MOV    CENTENA,A
RET

```

```

HOME:    MOV    A,#02H
        CLR    RS
        MOV    R2,#2
HOME_REPITA: RLC    A
        MOV    DB7,C
        RLC    A
        MOV    DB6,C
        RLC    A
        MOV    DB5,C
        RLC    A
        MOV    DB4,C
        SETB   E
        CLR    E
        MOV    VAR1,#145

```

```

MOV    VAR2,#4
LCALL  RETARDO_2
DJNZ   R2,HOME_REPITA
RET

```

;Tabla que utiliza la subrutina COD2HEX para el codigo de  
;la tecla presionada.

```

ORG    MANUAL_1
DFB    "01 %P Reboiler 1"
DFB    "02 Ver T Reboil1"
DFB    "03 Ver T Dallas2"
DFB    "04 Ver T Dallas3"
DFB    "05 I/O Reflujo  "
DFB    "06 I/O Cosecha  "
DFB    "07 I/O Alimentar"
DFB    "08 I/O Ventilar "
DFB    "09 Abrir 10 grad"
DFB    "10 Cerra 10 grad"
DFB    "11 Abrir valvula"
DFB    "12 Detener  "
DFB    "13 Cerra valvula"
DFB    "14      "
DFB    "15      "
DFB    "16      "

ORG    AUTOMATICO_1
DFB    "01 P inicio Reb1"
DFB    "02 SP T Fondos  "
DFB    "03 SP T Producto"
DFB    "04 Tempe Disparo"
DFB    "05 Ver T Reboil1"
DFB    "06 Ver T Dallas2"
DFB    "07 Ver T Dallas3"
DFB    "08 Delay Acumula"
DFB    "09 Min Act Motor"
DFB    "10 Max Act Motor"
DFB    "11 I/O Ventilar "
DFB    "12      "
DFB    "13      "
DFB    "14      "
DFB    "15      "
DFB    "16      "

ORG    T_CODIGO
DFB    0BEH,077H,0B7H,0D7H,07BH,0BBH,0DBH,07DH
DFB    0BDH,0DDH,0E7H,0EBH,0EDH,0EEH,07EH,0DEH

ORG    T_FILTRO_1
DFB    00H,00H,0AH,0AH,02H

ORG    T_FILTRO_2
DFB    00H,00H,01H,01H,02H

ORG    T_ASCII
DFB    30H,31H,32H,33H,34H,35H,36H,37H,38H,39H
DFB    41H,42H,43H,44H,45H,46H

ORG    INGRESE_ACT_REB1
DFB    "Actuador Reboil1"
DFB    "(0%-100%)="

ORG    INGRESE_INI_ACT
DFB    "P Inicial Reboil1"
DFB    "(0%-100%)="

```

ORG INGRESE\_SP\_T\_FONDOS  
DFB "SP T Fondos "  
DFB "(0 - 100)="

ORG INGRESE\_SP\_T\_PRODUC  
DFB "SP T Productos "  
DFB "(0 - 100)="

ORG INGRESE\_T\_DISPARO  
DFB "Tempe de Disparo"  
DFB "(0 - 100)="

ORG V\_T\_BASE  
DFB "P% R Base = "  
DFB "Reboiler1 = "

ORG V\_T\_DALLAS\_2  
DFB "P% R Base = "  
DFB "T Dallas 2= "

ORG V\_T\_DALLAS\_3  
DFB "P% R Base = "  
DFB "T Dallas 3= "

ORG INGRESE\_DELAY\_ACC  
DFB "Ingreso SP ACC "  
DFB "(0s-100s)="

ORG INGRESE\_CLAVE  
DFB "Ingreso clave "

ORG INGRESE\_MAX\_APER  
DFB "MAX Ape Motor\*10"  
DFB "(0 - 100)="

ORG INGRESE\_MIN\_APER  
DFB "MIN Ape Motor\*10"  
DFB "(0 - 100)="

End□

# **ANEXO 2**

## ANEXO 2: PROGRAMA DEL CONTROL DE TEMPERATURA DEL REBOILER

CPU "E:\SIDES51\8051.TBL"  
HOF "INT8"  
INCL "E:\SIDES51\SFR"

.....  
;TEMA: Control de temperatura del REBOILER  
.....

.....  
;REALIZADO POR: Marco Antonio Chávez  
.....

.....  
;CONSTANTES Y VARIABLES  
.....

DIRECCION: EQU 11H

SP\_T\_FONDOS: EQU 50H  
SP\_T\_PRODUC: EQU 51H  
T\_DISPARO: EQU 52H  
SER\_ACTUADOR: EQU 53H  
ACTUADOR: EQU 54H  
COMANDO: EQU 55H  
BANDELA: EQU 56H

MEDIDOR\_1: EQU 57H  
MEDIDOR\_2: EQU 58H  
MEDIDOR\_3: EQU 59H  
MEDIDOR\_4: EQU 5AH  
MEDIDOR\_5: EQU 5BH

CONTADOR: EQU 5CH

VAR1: EQU 70H  
VAR2: EQU 71H  
VAR3: EQU 72H

ONE\_WIRE\_1: EQU P3.2  
ONE\_WIRE\_2: EQU P1.4  
ONE\_WIRE\_3: EQU P1.5  
ONE\_WIRE\_4: EQU P1.6  
ONE\_WIRE\_5: EQU P1.7

DISPARO: EQU P3.3  
CRUCE\_CERO: EQU P3.4  
ERROR: EQU P3.5  
ENABLE: EQU P3.7

ESTABLE: EQU P1.0  
ACCION: EQU P1.1  
MUESTREO: EQU P1.2  
ERR\_ENABLE: EQU P1.3  
T\_ANGULO2PORCENTAJE: EQU 500H  
T\_PORCENTAJE2ANGULO: EQU 600H

.....  
;SOFTWARE  
.....

RESET: ORG 0000H  
LJMP RESET1

.....  
;INTERRUPCIONES  
.....

EXTIO: ORG 0003H ;INTERRUPCION EXTERNA 0  
TIMER0: ORG 000BH ;DESBORDAMIENTO TIMER 0

```

EXTI1:  ORG    0013H    ;INTERRUPCION EXTERNA 1
TIMER1: ORG    001BH    ;DESBORDAMIENTO TIMER 1
SINT:   ORG    0023H    ;PORTICO SERIAL
        LJMPL INT_SERIAL
EXTI2:  ORG    002BH    ;INT.EXT 2.o DESB.TIM.2

```

```

;VARIABLES

```

```

;ETIQUETA      OPCODE OPERANDOS      COMENTARIO

```

```

;INICIALIZACION DE MEMORIA RAM

```

```

        ORG    0030H    ;INICIO DEL PROGRAMA

RESET1: MOV    P0,#0FFH    ;INICIALIZO COMO ENTRADAS
        MOV    P1,#0FFH    ;A LOS PUERTOS
        MOV    P2,#0FFH    ;P0, P1, P2, P3
        MOV    P3,#0FFH    ;

        CLR    A
        MOV    IE,A
        MOV    IP,A
        MOV    TCON,A
        MOV    PCON,A
        MOV    PSW,A
        MOV    R0,#7FH
RESET2: MOV    @R0,A
        DJNZ   R0,RESET2
        MOV    SP,#2FH

```

```

;INICIO DE PROGRAMA PRINCIPAL

```

```

        MOV    TMOD,#20H
        MOV    PCON,#80H
        MOV    TH1,#0FDH
        MOV    SCON,#0F0H
        SETB   TR1
        MOV    IE,#10010000B

        MOV    ACTUADOR,#0
        MOV    SP_T_FONDOS,#81
        MOV    SP_T_PRODUC,#71
        MOV    T_DISPARO,#73

        MOV    VAR1,#206
        MOV    VAR2,#24
        MOV    VAR3,#22
        LCALL  RETARDO_3

TIMIN_1: LCALL  DALLAS_RESET_1
        JNB   ERROR,TIMIN_1
        MOV   A,#0CH
        LCALL DALLAS_WRITE_1
        MOV   A,#40H
        LCALL DALLAS_WRITE_1
        LCALL DALLAS_RESET_1
        MOV   A,#0EEH
        LCALL DALLAS_WRITE_1

TIMIN_2: LCALL  DALLAS_RESET_2
        JNB   ERROR,TIMIN_2
        MOV   A,#0CH
        LCALL DALLAS_WRITE_2
        MOV   A,#40H
        LCALL DALLAS_WRITE_2
        LCALL DALLAS_RESET_2
        MOV   A,#0EEH
        LCALL DALLAS_WRITE_2

```

```

TIMIN_3: LCALL DALLAS_RESET_3
        JNB  ERROR,TIMIN_3
        MOV  A,#0CH
        LCALL DALLAS_WRITE_3
        MOV  A,#40H
        LCALL DALLAS_WRITE_3
        LCALL DALLAS_RESET_3
        MOV  A,#0EEH
        LCALL DALLAS_WRITE_3

```

```

PROGRAMA: CJNE  R4,#0FFH,CONTROL
          LCALL MANUAL
          LJMP  PROGRAMA
CONTROL:  LCALL AUTOMATICO
          LJMP  PROGRAMA

```

```

MANUAL:  MOV   R7,#2
          MOV   R6,#0
MANU_ESPE1: JNB  CRUCE_CERO,MANU_ESPE1
MANU_ESPE2: JB   CRUCE_CERO,MANU_ESPE2
MANU_ESPE3: JNB  CRUCE_CERO,MANU_ESPE3
          LCALL ACT_ENABLE
          JC   MANU_SIGUE
          MOV  R3,ACTUADOR
          CJNE R3,#00H,MANU_RETAR
          LJMP MANU_DIREC
MANU_RETAR: LCALL RETAR_37US
          DJNZ R3,MANU_RETAR
MANU_DIREC: CLR  DISPARO
          LCALL RETAR_3US
          SETB DISPARO
MANU_SIGUE: DJNZ  R6,MANU_ESPE4
          DJNZ R7,MANU_ESPE4
          CPL  MUESTREO
          LCALL MEDIR
          LCALL ENVIAR_DATOS
          LJMP MANU_SALIR
MANU_ESPE4: JB   CRUCE_CERO,MANU_ESPE4
          LJMP MANU_ESPE3
MANU_SALIR: RET

```

```

AUTOMATICO: MOV   R7,#2
            MOV   R6,#0
AUTO_ESPE1: JNB  CRUCE_CERO,AUTO_ESPE1
AUTO_ESPE2: JB   CRUCE_CERO,AUTO_ESPE2
AUTO_ESPE3: JNB  CRUCE_CERO,AUTO_ESPE3
            LCALL ACT_ENABLE
            JC   AUTO_SIGUE
            MOV  R3,ACTUADOR
            CJNE R3,#00H,AUTO_RETAR
            LJMP AUTO_DIREC
AUTO_RETAR: LCALL RETAR_37US
            DJNZ R3,AUTO_RETAR
AUTO_DIREC: CLR  DISPARO
            LCALL RETAR_3US
            SETB DISPARO
AUTO_SIGUE: DJNZ  R6,AUTO_ESPE4
            DJNZ R7,AUTO_ESPE4
            CPL  MUESTREO
            LCALL MEDIR
            LCALL ENVIAR_DATOS
            LCALL ACT_ENABLE
            JC   AUTO_SALIR
            LCALL CONTROL_TEMPE
            LJMP AUTO_SALIR
AUTO_ESPE4: JB   CRUCE_CERO,AUTO_ESPE4
            LJMP AUTO_ESPE3
AUTO_SALIR: RET

```

```

CONTROL_TEMPE:      MOV    A,MEDIDOR_3
                   CJNE   A,T_DISPARO,CON_1_MAYMEN
CON_1_MAYOR:      MOV    ACTUADOR,SER_ACTUADOR
                   SETB   ESTABLE
                   LJMPL CON_SALIR
CON_1_MAYMEN:     JNC    CON_1_MAYOR
                   INC    CONTADOR
                   MOV    A,CONTADOR
                   CJNE   A,#15,CON_SALIR
                   MOV    CONTADOR,#0
                   CPL    ACCION
                   MOV    A,MEDIDOR_3
                   CJNE   A,SP_T_PRODUC,CON_2_MAYMEN
CON_2_MENOR:     LJMP   CON_FONDOS
CON_2_MAYMEN:     JC     CON_2_MENOR
                   LCALL  INCRE_ANGULO
                   SETB   ESTABLE
                   LJMPL CON_SALIR
CON_FONDOS:      MOV    A,MEDIDOR_1
                   CJNE   A,SP_T_FONDOS,CON_3_MAYMEN
CON_3_MAYOR:     CLR    ESTABLE
                   LJMPL CON_SALIR
CON_3_MAYMEN:     JNC    CON_3_MAYOR
                   LCALL  DECRE_ANGULO
                   SETB   ESTABLE
CON_SALIR:       RET

```

```

INCRE_ANGULO:    MOV    A,ACTUADOR
                   CJNE   A,SER_ACTUADOR,INC_MAYMEN
INC_MAYOR:      LJMP   INC_SALIR
INC_MAYMEN:     JNC    INC_MAYOR
                   INC    ACTUADOR
INC_SALIR:      RET

```

```

DECRE_ANGULO:   MOV    A,ACTUADOR
                   CJNE   A,#0,DEC_MAYMEN
                   LJMPL DEC_SALIR
DEC_MAYMEN:     DEC    ACTUADOR
DEC_SALIR:     RET

```

```

INT_SERIAL:     CLR    EA
                   SETB  DISPARO
                   CLR    R1
                   PUSH  ACC
                   MOV    A,SBUF
                   CJNE   A,#0FH,INT_SER_MANU_AUTO
                   MOV    R4,#0FFH
                   LJMPL INT_SER_SALIR
INT_SER_MANU_AUTO: CJNE  R4,#0FFH,INT_SER_AUTO
                   CJNE   A,#DIRECCION,INT_SER_SALIR
                   CLR    SM2
                   LCALL  SERIAL_MANUAL
                   LJMPL INT_SER_SALIR
INT_SER_AUTO:   CJNE   A,#DIRECCION,INT_SER_SALIR
                   CLR    SM2
                   LCALL  SERIAL_AUTOMATICO
INT_SER_SALIR:  POP    ACC
                   SETB  SM2
                   SETB  EA
RETI

```

```

SERIAL_MANUAL:  JNB    R1,SERIAL_MANUAL
                   CLR    R1
                   MOV    A,SBUF
                   CJNE   A,#11H,SER_MANU_OP2
SER_MANU_ESPO:  JNB    R1,SER_MANU_ESPO

```

```

        CLR    RI
        MOV    A,SBUF
        MOV    DPTR,#T_PORCENTAJE2ANGULO
        MOVC  A,@A+DPTR
        MOV    ACTUADOR,A
        LJMP  SER_MANU_SALIR
SER_MANU_OP2: CJNE  A,#0FOH,SER_MANU_SALIR
        MOV    R4,#0
SER_MANU_SALIR:    RET

```

```

SERIAL_AUTOMATICO:  JNB    RI,SERIAL_AUTOMATICO
        CLR    RI
        MOV    A,SBUF
        CJNE  A,#2DH,SER_AUTO_OP2
SER_AUTO_WAIT: JNB    RI,SER_AUTO_WAIT
        CLR    RI
        MOV    A,SBUF
        MOV    DPTR,#T_PORCENTAJE2ANGULO
        MOVC  A,@A+DPTR
        MOV    SER_ACTUADOR,A
        LJMP  SER_AUTO_SALIR
SER_AUTO_OP2: CJNE  A,#33H,SER_AUTO_OP3
SER_AUTO_TIAW: JNB    RI,SER_AUTO_TIAW
        CLR    RI
        MOV    SP_T_FONDOS,SBUF
        LJMP  SER_AUTO_SALIR
SER_AUTO_OP3: CJNE  A,#4BH,SER_AUTO_OP4
SER_AUTO_ESPE:  JNB    RI,SER_AUTO_ESPE
        CLR    RI
        MOV    SP_T_PRODUC,SBUF
        LJMP  SER_AUTO_SALIR
SER_AUTO_OP4: CJNE  A,#55H,SER_AUTO_SALIR
SER_AUTO_WATE:  JNB    RI,SER_AUTO_WATE
        MOV    T_DISPARO,SBUF
SER_AUTO_SALIR:  RET

```

```

ENVIAR_DATOS: CLR    EA
        MOV    A,#0EEH
        LCALL ENVIAR
        MOV    A,MEDIDOR_1
        LCALL ENVIAR
        MOV    A,MEDIDOR_2
        LCALL ENVIAR
        MOV    A,MEDIDOR_3
        LCALL ENVIAR
        MOV    A,ACTUADOR
        MOV    DPTR,#T_ANGULO2PORCENTAJE
        MOVC  A,@A+DPTR
        LCALL ENVIAR
        SETB  EA

```

RET

```

ENVIAR: MOV    SBUF,A
ENV_ESPE: JNB    TI,ENV_ESPE
        CLR    TI
RET

```

```

MEDIR: LCALL  DALLAS_RESET_1
        JNB  ERROR,MEDIR
        MOV  A,#0AAH
        LCALL DALLAS_WRITE_1
        LCALL DALLAS_READ_1
        MOV  MEDIDOR_1,A
MEDIR_2: LCALL  DALLAS_RESET_2
        JNB  ERROR,MEDIR_2
        MOV  A,#0AAH

```

```

        LCALL DALLAS_WRITE_2
        LCALL DALLAS_READ_2
        MOV   MEDIDOR_2,A
MEDIR_3: LCALL DALLAS_RESET_3
        JNB  ERROR,MEDIR_3
        MOV  A,#0AAH
        LCALL DALLAS_WRITE_3
        LCALL DALLAS_READ_3
        MOV  MEDIDOR_3,A
RET

```

```

ACT_ENABLE: MOV   C,ENABLE
            CPL   C
            MOV  ERR_ENABLE,C

            MOV  C,ENABLE
            JNC  ACT_SEGUIR

            MOV  A,MEDIDOR_1
            CJNE A,#70,ACT_MAYMEN
ACT_MAYOR:  MOV  ACTUADOR,SER_ACTUADOR
            LJMP ACT_SEGUIR
ACT_MAYMEN: JNC  ACT_MAYOR
ACT_SEGUIR: MOV  C,ENABLE
RET

```

```

RETARDO_3: DJNZ  VAR1,RETARDO_3
            DJNZ  VAR2,RETARDO_3
            DJNZ  VAR3,RETARDO_3
RET

```

```

RETAR_37US: MOV  VAR1,#9
            MOV  VAR2,#1
            MOV  VAR3,#1
RE_37US:   DJNZ  VAR1,RE_37US
            DJNZ  VAR2,RE_37US
            DJNZ  VAR3,RE_37US
RET

```

```

RETAR_3US:  MOV  VAR1,#1
            MOV  VAR2,#1
            MOV  VAR3,#1
RE_3US:    DJNZ  VAR1,RE_3US
            DJNZ  VAR2,RE_3US
            DJNZ  VAR3,RE_3US
RET

```

```

MICRO:     DJNZ  VAR1,MICRO
RET

```

```

DALLAS_READ_1: MOV  R2,#8
D_R_REPITA_1:  CLR  ONE_WIRE_1
            SETB ONE_WIRE_1
            MOV  VAR1,#1
            LCALL MICRO
            MOV  C,ONE_WIRE_1
            MOV  VAR1,#24
            LCALL MICRO
            RRC  A
            DJNZ R2,D_R_REPITA_1
RET

```

```

DALLAS_READ_2: MOV  R2,#8
D_R_REPITA_2:  CLR  ONE_WIRE_2

```

```

    SETB  ONE_WIRE_2
    MOV   VAR1,#1
    LCALL MICRO
    MOV   C,ONE_WIRE_2
    MOV   VAR1,#24
    LCALL MICRO
    RRC   A
    DJNZ  R2,D_R_REPITA_2
RET

```

```

DALLAS_READ_3: MOV   R2,#8
D_R_REPITA_3:  CLR   ONE_WIRE_3
    SETB  ONE_WIRE_3
    MOV   VAR1,#1
    LCALL MICRO
    MOV   C,ONE_WIRE_3
    MOV   VAR1,#24
    LCALL MICRO
    RRC   A
    DJNZ  R2,D_R_REPITA_3
RET

```

```

DALLAS_RESET_1: CLR   ONE_WIRE_1
    MOV   A,#151
D_R_RETA_1:    DEC   A
    JNZ  D_R_RETA_1
    SETB ONE_WIRE_1
    MOV   A,#13
D_R_ESPERAR_1: DEC   A
    JNZ  D_R_VERIFICAR_1
    SJMP D_R_ERROR_1
D_R_VERIFICAR_1: JB   ONE_WIRE_1,D_R_ESPERAR_1
    MOV   A,#46
D_R_WAIT_1:   DEC   A
    JNZ  D_R_ANALIZAR_1
    SJMP D_R_ERROR_1
D_R_ANALIZAR_1: JNB  ONE_WIRE_1,D_R_WAIT_1
    MOV   A,#128
D_R_REGUE_1:  DEC   A
    JNZ  D_R_REGUE_1
    SETB ERROR
    SJMP D_R_SALIR_1
D_R_ERROR_1:  CLR   ERROR
D_R_SALIR_1:  RET

```

```

DALLAS_RESET_2: CLR   ONE_WIRE_2
    MOV   A,#151
D_R_RETA_2:    DEC   A
    JNZ  D_R_RETA_2
    SETB ONE_WIRE_2
    MOV   A,#13
D_R_ESPERAR_2: DEC   A
    JNZ  D_R_VERIFICAR_2
    SJMP D_R_ERROR_2
D_R_VERIFICAR_2: JB   ONE_WIRE_2,D_R_ESPERAR_2
    MOV   A,#46
D_R_WAIT_2:   DEC   A
    JNZ  D_R_ANALIZAR_2
    SJMP D_R_ERROR_2
D_R_ANALIZAR_2: JNB  ONE_WIRE_2,D_R_WAIT_2
    MOV   A,#128
D_R_REGUE_2:  DEC   A
    JNZ  D_R_REGUE_2
    SETB ERROR
    SJMP D_R_SALIR_2
D_R_ERROR_2:  CLR   ERROR
D_R_SALIR_2:  RET

```

```

DALLAS_RESET_3:      CLR      ONE_WIRE_3
                    MOV      A,#151
D_R_RETA_3:         DEC      A
                    JNZ      D_R_RETA_3
                    SETB     ONE_WIRE_3
                    MOV      A,#13
D_R_ESPERAR_3:      DEC      A
                    JNZ      D_R_VERIFICAR_3
                    SJMP     D_R_ERROR_3
D_R_VERIFICAR_3:    JB      ONE_WIRE_3,D_R_ESPERAR_3
                    MOV      A,#46
D_R_WAIT_3:         DEC      A
                    JNZ      D_R_ANALIZAR_3
                    SJMP     D_R_ERROR_3
D_R_ANALIZAR_3:     JNB      ONE_WIRE_3,D_R_WAIT_3
                    MOV      A,#128
D_R_REGUE_3:        DEC      A
                    JNZ      D_R_REGUE_3
                    SETB     ERROR
                    SJMP     D_R_SALIR_3
D_R_ERROR_3:        CLR      ERROR
D_R_SALIR_3:        RET

```

```

DALLAS_WRITE_1:     MOV      R2,#8
D_W_REPITA_1:       RRC      A
                    JC      D_W_ESUNO_1
                    CLR      ONE_WIRE_1
                    MOV      VAR1,#28
                    LCALL    MICRO
                    SETB     ONE_WIRE_1
                    SJMP     D_W_OTRO_1
D_W_ESUNO_1:        CLR      ONE_WIRE_1
                    SETB     ONE_WIRE_1
                    MOV      VAR1,#28
                    LCALL    MICRO
D_W_OTRO_1:         DJNZ    R2,D_W_REPITA_1
                    RET

```

```

DALLAS_WRITE_2:     MOV      R2,#8
D_W_REPITA_2:       RRC      A
                    JC      D_W_ESUNO_2
                    CLR      ONE_WIRE_2
                    MOV      VAR1,#28
                    LCALL    MICRO
                    SETB     ONE_WIRE_2
                    SJMP     D_W_OTRO_2
D_W_ESUNO_2:        CLR      ONE_WIRE_2
                    SETB     ONE_WIRE_2
                    MOV      VAR1,#28
                    LCALL    MICRO
D_W_OTRO_2:         DJNZ    R2,D_W_REPITA_2
                    RET

```

```

DALLAS_WRITE_3:     MOV      R2,#8
D_W_REPITA_3:       RRC      A
                    JC      D_W_ESUNO_3
                    CLR      ONE_WIRE_3
                    MOV      VAR1,#28
                    LCALL    MICRO
                    SETB     ONE_WIRE_3
                    SJMP     D_W_OTRO_3
D_W_ESUNO_3:        CLR      ONE_WIRE_3
                    SETB     ONE_WIRE_3
                    MOV      VAR1,#28
                    LCALL    MICRO
D_W_OTRO_3:         DJNZ    R2,D_W_REPITA_3

```

RET

ORG T\_ANGULO2PORCENTAJE

DFB 100,99,98,98,97,97,96,96,95,95,94,93,93,92,92,91,91,90,90 ;19  
DFB 89,89,88,87,87,86,86,85,85,84,83,83,82,82,81,81,80,80 ;18  
DFB 79,78,78,77,77,76,76,75,75,74,73,73,72,72,71,71,70 ;17  
DFB 69,69,68,68,67,67,66,66,65,64,64,63,63,62,62,61,61,60 ;18  
DFB 59,59,58,58,57,57,56,55,55,54,54,53,53,52,52,51,50,50 ;18  
DFB 49,49,48,48,47,46,46,45,45,44,44,43,43,42,41,41,40,40 ;18  
DFB 39,39,38,37,37,36,36,35,35,34,34,33,33,32,31,31,30,30 ;18  
DFB 29,29,28,27,27,26,26,25,25,24,24,23,23,22,22,21,21,20,20 ;19  
DFB 19,19,18,18,17,17,16,16,15,14,13,13,12,12,11,11,10,10 ;18  
DFB 9,9,8,8,7,7,6,6,5,4,3,3,2,2,1,1,0,0 ;18

ORG T\_PORCENTAJE2ANGULO

;Son 101 valores que hay que colocar porque el ingreso es de 00 a 100  
;el cero tambien cuenta.

DFB 180,178,176,174,172,171,170,168,166,164,162  
DFB 160,158,156,154,153,152,150,148,146,144,142  
DFB 140,138,136,134,132,130,128,126,124,122,121  
DFB 120,118,116,114,112,110,108,106,104,103,102  
DFB 100,98,96,94,92,90,88,87,86,84,82  
DFB 80,78,76,74,72,71,70,68,66,64,62  
DFB 60,58,56,54,53,52,50,48,46,44,42  
DFB 40,38,37,36,34,32,30,28,26,24,22  
DFB 21,20,18,16,14,12,10,8,6,4,2,1,0

ENDD

# **ANEXO 3**

# ANEXO 3: PROGRAMA DEL CONTROL DE NIVEL DEL REBOILER

```
CPU "E:\SIDES51\8051.TBL"  
HOF "INT8"  
INCL "E:\SIDES51\SFR"
```

```
.....  
;TEMA: Control de nivel del REBOILER  
.....
```

```
.....  
;REALIZADO POR: Marco Antonio Chávez  
.....
```

```
.....  
;CONSTANTES Y VARIABLES  
.....
```

```
DIRECCION: EQU 33H
```

```
VAR1: EQU 40H  
VAR2: EQU 41H  
VAR3: EQU 42H
```

```
AUX_VAR1: EQU 43H  
AUX_VAR2: EQU 44H  
AUX_VAR3: EQU 45H
```

```
COMANDO: EQU 46H  
MAX_ACT: EQU 47H  
MIN_ACT: EQU 48H  
SECUENCIA: EQU 49H  
AUX: EQU 4AH  
NIVEL: EQU 4BH
```

```
CONTADOR_1: EQU 4CH  
CONTADOR_2: EQU 4DH
```

```
AUX_MAX_ACT: EQU 4EH  
AUX_MIN_ACT: EQU 4FH
```

```
BANDERA: EQU 50H  
BANDERA_1:
```

```
SENSOR_1: EQU P2.4  
SENSOR_2: EQU P2.5  
LED_SENSOR_1: EQU P2.6  
LED_SENSOR_2: EQU P2.7
```

```
ENABLE: EQU P3.2  
ERROR: EQU P3.3
```

```
.....  
;SOFTWARE  
.....
```

```
RESET: ORG 0000H  
LJMP RESET1
```

```
.....  
;INTERRUPCIONES  
.....
```

```
EXTI0: ORG 0003H ;INTERRUPCION EXTERNA 0  
TIMER0: ORG 000BH ;DESBORDAMIENTO TIMER 0  
EXTI1: ORG 0013H ;INTERRUPCION EXTERNA 1  
TIMER1: ORG 001BH ;DESBORDAMIENTO TIMER 1  
SINT: ORG 0023H ;PORTICO SERIAL  
LJMP INT_SERIAL
```

EXTI2: ORG 002BH ;INT.EXT 2.o DESB.TIM.2

.....  
;VARIABLES  
.....

;ETIQUETA            OPCODE   OPERANDOS            COMENTARIO

.....  
;INICIALIZACION DE MEMORIA RAM  
.....

ORG 0030H ;INICIO DEL PROGRAMA

RESET1: MOV P0,#0FFH ;INICIALIZO COMO ENTRADAS  
MOV P1,#0FFH ;A LOS PUERTOS  
MOV P2,#0FFH ;P0, P1, P2, P3  
MOV P3,#0FFH ;

CLR A  
MOV IE,A  
MOV IP,A  
MOV TCON,A  
MOV PCON,A  
MOV PSW,A  
MOV R0,#7FH

RESET2: MOV @R0,A  
DJNZ R0,RESET2  
MOV SP,#2FH

.....  
;INICIO DE PROGRAMA PRINCIPAL  
.....

MOV TMOD,#21H  
MOV PCON,#80H  
MOV TH1,#0FDH  
MOV SCON,#0F0H  
SETB TR1  
MOV IE,#10010000B

MOV NIVEL,#0  
SETB ENABLE  
MOV SECUENCIA,#33H  
MOV AUX\_VAR1,#216  
MOV AUX\_VAR2,#36  
MOV AUX\_VAR3,#1  
MOV MAX\_ACT,#60  
MOV MIN\_ACT,#0

LCALL CALIBRAR  
MOV AUX\_MAX\_ACT,#10  
MOV AUX\_MIN\_ACT,#3

PROGRAMA: CJNE R4,#0FFH,CONTROL  
LCALL MANUAL  
LJMP PROGRAMA  
CONTROL: LCALL AUTOMATICO  
LJMP PROGRAMA

AUTOMATICO: LCALL ACT\_SENSORES  
LCALL SEN\_NIVEL  
MOV A,BANDERA  
CJNE A,#0,AUTO\_FONDOS  
MOV MIN\_ACT,#0

AUTO\_REPE: LCALL CERRAR  
MOV A,CONTADOR\_1  
CJNE A,#0,AUTO\_REPE  
MOV A,CONTADOR\_2  
CJNE A,#0,AUTO\_REPE  
MOV BANDERA\_1,#0  
LJMP AUTO\_SALIR

AUTO\_FONDOS: MOV A,BANDERA\_1  
CJNE A,#0,AUTO\_SEGUIR

```

MOV     MAX_ACT,AUX_MAX_ACT
MOV     MIN_ACT,AUX_MIN_ACT
LCALL  ACT_MOTOR
AUTO_SEGUIR: MOV     BANDERA_1,#0FFH
;MOV     MAX_ACT,AUX_MAX_ACT
;MOV     MIN_ACT,AUX_MIN_ACT
MOV     A,NIVEL
CJNE   A,#6,AUTO_MAYMEN
LCALL  CERRAR
LJMP   AUTO_SALIR
AUTO_MAYMEN: JNC     AUTO_MAYOR
LCALL  CERRAR
LJMP   AUTO_SALIR
AUTO_MAYOR: LCALL  ABRIR
AUTO_SALIR: RET

MANUAL:  MOV     MAX_ACT,#60
MOV     MIN_ACT,#0
LCALL  ACT_SENSORES
LCALL  SEN_NIVEL
MOV     A,COMANDO
CJNE   A,#11H,MANU_OP2
MOV     R6,#10
MANU_REP: LCALL  ABRIR
DJNZ   R6,MANU_REP
MOV     COMANDO,#55H
LJMP   MANUAL_SALIR
MANU_OP2: CJNE   A,#2DH,MANU_OP3
MOV     R6,#10
MANU_PER: LCALL  CERRAR
DJNZ   R6,MANU_PER
MOV     COMANDO,#55H
LJMP   MANUAL_SALIR
MANU_OP3: CJNE   A,#33H,MANU_OP4
LCALL  ABRIR
MOV     A,CONTADOR_2
CJNE   A,MAX_ACT,MANUAL_SALIR
MOV     COMANDO,#55H
LJMP   MANUAL_SALIR
MANU_OP4: CJNE   A,#4BH,MANU_OP5
LCALL  CERRAR
MOV     A,CONTADOR_1
CJNE   A,#0,MANUAL_SALIR
MOV     A,CONTADOR_2
CJNE   A,MIN_ACT,MANUAL_SALIR
MOV     COMANDO,#55H
LJMP   MANUAL_SALIR
MANU_OP5: CJNE   A,#55H,MANUAL_SALIR
MANUAL_SALIR: RET

INCREMENTAR: INC     CONTADOR_1
MOV     A,CONTADOR_1
CJNE   A,#10,INCRE_SALIR
MOV     CONTADOR_1,#0
INC     CONTADOR_2
INCRE_SALIR: RET

DECREMENTAR: DEC     CONTADOR_1
MOV     A,CONTADOR_1
CJNE   A,#0FFH,DECRE_SALIR
MOV     CONTADOR_1,#9
DEC     CONTADOR_2
DECRE_SALIR: RET

CALIBRAR:  MOV     AUX_VAR1,#116
MOV     AUX_VAR2,#132
MOV     AUX_VAR3,#4
MOV     R7,#20
CAL_OTRA:  MOV     CONTADOR_1,#0
MOV     CONTADOR_2,#0
LCALL  ABRIR

```

```

        LCALL ACT_SENTORES
        LCALL SEN_NIVEL
        DJNZ R7,CALI_OTRA
        MOV AUX_VAR1,#216
        MOV AUX_VAR2,#36
        MOV AUX_VAR3,#1
CALI_REPITA: MOV CONTADOR_1,#0
        MOV CONTADOR_2,MAX_ACT
        LCALL CERRAR
        LCALL ACT_SENTORES
        LCALL SEN_NIVEL
        JB SENSOR_1,CALI_REPITA
        MOV R7,#20
CALI_AGAIN: MOV CONTADOR_1,#0
        MOV CONTADOR_2,MAX_ACT
        LCALL CERRAR
        LCALL ACT_SENTORES
        LCALL SEN_NIVEL
        JNB SENSOR_2,CALI_REPITA
        DJNZ R7,CALI_AGAIN
        MOV AUX_VAR1,#116
        MOV AUX_VAR2,#132
        MOV AUX_VAR3,#4
        MOV R7,#20
CALI_ONCE:  MOV CONTADOR_1,#0
        MOV CONTADOR_2,#0
        LCALL ABRIR
        LCALL ACT_SENTORES
        LCALL SEN_NIVEL
        DJNZ R7,CALI_ONCE
        MOV AUX_VAR1,#216
        MOV AUX_VAR2,#36
        MOV AUX_VAR3,#1
CALI_OTHER: LCALL CERRAR
        MOV CONTADOR_1,#0
        MOV CONTADOR_2,MAX_ACT
        LCALL ACT_SENTORES
        LCALL SEN_NIVEL
        JB SENSOR_1,CALI_OTHER
        MOV CONTADOR_1,#0
        MOV CONTADOR_2,#0
RET

ACT_SENTORES: MOV C,SENSOR_1
        MOV LED_SENSOR_1,C
        MOV C,SENSOR_2
        MOV LED_SENSOR_2,C
RET

SEN_NIVEL:  JB P0.0,ES_1
        LCALL RETARDO_1
        JB P0.0,ES_1
        MOV P1,#0D7H
        MOV NIVEL,#1
        SETB ENABLE
        LJMP SEN_NIV_FIN
ES_1:      JB P0.1,ES_2
        LCALL RETARDO_1
        JB P0.1,ES_2
        MOV P1,#0A2H
        MOV NIVEL,#2
        SETB ENABLE
        LJMP SEN_NIV_FIN
ES_2:      JB P0.2,ES_3
        LCALL RETARDO_1
        JB P0.2,ES_3
        MOV P1,#83H
        MOV NIVEL,#3
        SETB ENABLE
        LJMP SEN_NIV_FIN
ES_3:      JB P0.3,ES_4
        LCALL RETARDO_1
        JB P0.3,ES_4
        MOV P1,#0C5H
        MOV NIVEL,#4

```

```

SETB  ENABLE
LJMP  SEN_NIV_FIN
ES_4: JB   P0.4,ES_5
      LCALL RETARDO_1
      JB   P0.4,ES_5
      MOV  P1,#89H
      MOV  NIVEL,#5
      CLR  ENABLE
      LJMP  SEN_NIV_FIN
ES_5: JB   P0.5,ES_6
      LCALL RETARDO_1
      JB   P0.5,ES_6
      MOV  P1,#8CH
      MOV  NIVEL,#6
      CLR  ENABLE
      LJMP  SEN_NIV_FIN
ES_6: JB   P0.6,ES_7
      LCALL RETARDO_1
      JB   P0.6,ES_7
      MOV  P1,#0D3H
      MOV  NIVEL,#7
      CLR  ENABLE
      LJMP  SEN_NIV_FIN
ES_7: JB   P0.7,SEN_NIV_FIN
      LCALL RETARDO_1
      JB   P0.7,SEN_NIV_FIN
      MOV  P1,#80H
      MOV  NIVEL,#8
      CLR  ENABLE
SEN_NIV_FIN: RET

```

```

ACT_MOTOR:  MOV  A,CONTADOR_2
            CJNE A,MIN_ACT,ACT_MOTOR_MAYMEN
            LJMP  ACT_MOTOR_SALIR
ACT_MOTOR_MAYMEN: JNC  ACT_MOTOR_MAYOR
ACT_MOTOR_REPITA: LCALL ABRIR
            MOV  A,CONTADOR_2
            CJNE A,MIN_ACT,ACT_MOTOR_REPITA
            LJMP  ACT_MOTOR_SALIR
ACT_MOTOR_MAYOR:  LCALL CERRAR
            MOV  A,CONTADOR_1
            CJNE A,#0,ACT_MOTOR_MAYOR
            MOV  A,CONTADOR_2
            CJNE A,MIN_ACT,ACT_MOTOR_MAYOR
ACT_MOTOR_SALIR:  RET

```

```

ABRIR:  MOV  A,CONTADOR_2
        CJNE A,MAX_ACT,ABRIR_MAYMEN
        LJMP  ABRIR_SALIR
ABRIR_MAYMEN: JNC  ABRIR_SALIR
ABRIR_SEGUIR: MOV  A,SECUENCIA
            RL
            MOV  SECUENCIA,A
            ORL  A,#0F0H
            MOV  AUX,A
            MOV  A,P2
            ORL  A,#0FH
            ANL  A,AUX
            MOV  P2,A
            SETB SENSOR_1
            SETB SENSOR_2
            MOV  VAR1,AUX_VAR1
            MOV  VAR2,AUX_VAR2
            MOV  VAR3,AUX_VAR3
            LCALL RETARDO
            LCALL INCREMENTAR
ABRIR_SALIR:  RET

```

```

CERRAR:  MOV  A,CONTADOR_1
        CJNE A,#0,CERRAR_SEGUIR
        MOV  A,CONTADOR_2

```

```

CJNE A,MIN_ACT,CERRAR_MAYMEN
LJMP CERRAR_SALIR
CERRAR_MAYMEN: JC CERRAR_SALIR
CERRAR_SEGUIR: MOV A,SECUENCIA
RR A
MOV SECUENCIA,A
ORL A,#0F0H
MOV AUX,A
MOV A,P2
ORL A,#0FH
ANL A,AUX
MOV P2,A
SETB SENSOR_1
SETB SENSOR_2
MOV VAR1,AUX_VAR1
MOV VAR2,AUX_VAR2
MOV VAR3,AUX_VAR3
LCALL RETARDO
LCALL DECREMENTAR
MOV A,CONTADOR_1
CJNE A,#0,CERRAR_SALIR
MOV A,CONTADOR_2
CJNE A,#0,CERRAR_SALIR
JNB SENSOR_1,CERRAR_SALIR
CLR ERROR
CERRAR_SALIR: RET

RETARDO: DJNZ VAR1,RETARDO
DJNZ VAR2,RETARDO
DJNZ VAR3,RETARDO
RET

RETARDO_1: MOV VAR1,#234
MOV VAR2,#18
RET_1_OTRA: DJNZ VAR1,RET_1_OTRA
DJNZ VAR2,RET_1_OTRA
RET

INT_SERIAL: CLR EA
CLR RI
PUSH ACC
MOV A,SBUF
CJNE A,#0FH,INT_SER_MANU_AUTO
MOV COMANDO,#55H
MOV R4,#0FFH
MOV AUX_MAX_ACT,MAX_ACT
MOV AUX_MIN_ACT,MIN_ACT
LJMP INT_SER_SALIR
INT_SER_MANU_AUTO: CJNE R4,#0FFH,INT_SER_AUTO
CJNE A,#DIRECCION,INT_SER_SALIR
CLR SM2
LCALL SERIAL_MANUAL
LJMP INT_SER_SALIR
INT_SER_AUTO: CJNE A,#DIRECCION,INT_SER_SALIR
CLR SM2
LCALL SERIAL_AUTOMATICO
INT_SER_SALIR: POP ACC
SETB SM2
SETB EA
RETI

SERIAL_MANUAL: JNB RI,SERIAL_MANUAL
CLR RI
MOV A,SBUF
CJNE A,#69H,SER_MANU_OP2
SER_MANU_ESPO: JNB RI,SER_MANU_ESPO
CLR RI
MOV COMANDO,SBUF
LJMP SER_MANU_SALIR
SER_MANU_OP2: CJNE A,#0F0H,SER_MANU_SALIR

```

```

MOV R4,#0
MOV MAX_ACT,AUX_MAX_ACT
MOV MIN_ACT,AUX_MIN_ACT
SER_MANU_SALIR: RET

SERIAL_AUTOMATICO: JNB RI,SERIAL_AUTOMATICO
CLR RI
MOV A,SBUF
SER_AUTO_OP1: CJNE A,#77H,SER_AUTO_OP2
SER_AUTO_WAIT: JNB RI,SER_AUTO_WAIT
CLR RI
MOV MIN_ACT,SBUF
MOV AUX_MIN_ACT,SBUF
MOV BANDERA_1,#0
LJMP SER_AUTO_SALIR
SER_AUTO_OP2: CJNE A,#88H,SER_AUTO_OP3
SER_AUTO_TIAW: JNB RI,SER_AUTO_TIAW
CLR RI
MOV MAX_ACT,SBUF
MOV AUX_MAX_ACT,SBUF
LJMP SER_AUTO_SALIR
SER_AUTO_OP3: CJNE A,#96H,SER_AUTO_OP4
MOV BANDERA,#0FFH
LJMP SER_AUTO_SALIR
SER_AUTO_OP4: CJNE A,#0AAH,SER_AUTO_SALIR
MOV BANDERA,#0
SER_AUTO_SALIR: RET

```

END

# ANEXO 4

# ANEXO 4: PROGRAMA DEL CONTROL DEL ACUMULADOR, VENTILADOR Y BOMBAS

```
CPU "E:\SIDES51\8051.TBL"  
HOF "INT8"  
INCL "E:\SIDES51\SFR"
```

```
.....  
;TEMA: Control del acumulador, ventilador y bombas  
.....
```

```
.....  
;REALIZADO POR: Marco Antonio Chávez  
.....
```

```
.....  
;CONSTANTES Y VARIABLES  
.....
```

```
REFLUJO: EQU P3.2  
COSECHA: EQU P3.3  
ALIMENTAR: EQU P3.4  
VENTILAR: EQU P3.5
```

```
L_REFLUJO: EQU P1.2  
L_COSECHA: EQU P1.1  
L_ALIMENTAR: EQU P1.0  
L_VENTILAR: EQU P3.7
```

```
VAR1: EQU 40H  
VAR2: EQU 41H  
VAR3: EQU 42H  
TIEMPO: EQU 43H  
COMANDO: EQU 44H
```

```
DIRECCION: EQU 2DH
```

```
.....  
;SOFTWARE  
.....
```

```
RESET: ORG 0000H  
LJMP RESET1
```

```
.....  
;INTERRUPCIONES  
.....
```

```
EXTI0: ORG 0003H ;INTERRUPCION EXTERNA 0  
TIMER0: ORG 000BH ;DESBORDAMIENTO TIMER 0  
; LJMP INT_TIMER0  
EXTI1: ORG 0013H ;INTERRUPCION EXTERNA 1  
TIMER1: ORG 001BH ;DESBORDAMIENTO TIMER 1  
SINT: ORG 0023H ;PORTICO SERIAL  
LJMP INT_SERIAL  
EXTI2: ORG 002BH ;INT.EXT 2.o DESB.TIM.2
```

```
.....  
;VARIABLES  
.....
```

```
;ETIQUETA OPCODE OPERANDOS COMENTARIO
```

```
.....  
;INICIALIZACION DE MEMORIA RAM  
.....
```

```
ORG 0030H ;INICIO DEL PROGRAMA
```

```
RESET1: MOV P0,#0FFH ;INICIALIZO COMO ENTRADAS  
MOV P1,#0FFH ;A LOS PUERTOS
```

```

MOV P2,#0FFH ;P0, P1, P2, P3
MOV P3,#0FFH ;

CLR A
MOV IE,A
MOV IP,A
MOV TCON,A
MOV PCON,A
MOV PSW,A
MOV R0,#7FH
RESET2: MOV @R0,A
DJNZ R0,RESET2
MOV SP,#2FH

```

```

.....
;INICIO DE PROGRAMA PRINCIPAL
.....

```

```

MOV TMOD,#21H
MOV PCON,#80H
MOV TH1,#0FDH
MOV SCON,#0F0H
SETB TR1
MOV IE,#10010000B

MOV TIEMPO,#30
MOV R4,#0

PROGRAMA: CJNE R4,#0FFH,CONTROL
LCALL MANUAL
LJMP PROGRAMA
CONTROL: LCALL AUTOMATICO
LJMP PROGRAMA

```

```

MANUAL: CJNE R5,#0,MANU_DISER
SETB COSECHA
SETB ALIMENTAR
SETB REFLUJO
SETB L_COSECHA
SETB L_ALIMENTAR
SETB L_REFLUJO
MOV R5,#0FFH
MANU_DISER: MOV A,COMANDO
MANU_OP1: CJNE A,#11H,MANU_OP2
CPL REFLUJO
CPL L_REFLUJO
MOV COMANDO,#55H
LJMP MANU_SALIR
MANU_OP2: CJNE A,#2DH,MANU_OP3
CPL COSECHA
CPL L_COSECHA
MOV COMANDO,#55H
LJMP MANU_SALIR
MANU_OP3: CJNE A,#33H,MANU_OP4
CPL ALIMENTAR
CPL L_ALIMENTAR
MOV COMANDO,#55H
LJMP SER_MANU_SALIR
MANU_OP4: CJNE A,#4BH,MANU_OP5
CPL VENTILAR
CPL L_VENTILAR
MOV COMANDO,#55H
LJMP SER_MANU_SALIR
MANU_OP5: CJNE A,#55H,MANU_SALIR
MANU_SALIR: RET

```

```

AUTOMATICO: MOV R5,#0
JB P1.7,AUTO_ES_2
LCALL RETARDO_1
JB P1.7,AUTO_ES_2
SETB REFLUJO

```

```

        SETB  L_REFLUJO
        SETB  COSECHA
        SETB  L_COSECHA
AUTO_ES_2:  JB    P1.6,AUTO_ES_3
        LCALL RETARDO_1
        JB    P1.6,AUTO_ES_3
        CLR  REFLUJO
        CLR  L_REFLUJO
        CLR  COSECHA
        CLR  L_COSECHA
        MOV  R6,TIEMPO
AUTO_EVACUAR: LCALL  CIEN_MILI
        DJNZ R6,AUTO_EVACUAR
        SETB COSECHA
        SETB L_COSECHA
AUTO_ES_3:  JB    P1.5,AUTO_RESE_LLENO
        LCALL RETARDO_1
        JB    P1.5,AUTO_RESE_LLENO
        CLR  REFLUJO
        CLR  L_REFLUJO
        CLR  COSECHA
        CLR  L_COSECHA
AUTO_RESE_LLENO: JB    P1.3,AUTO_APAGAR
        JB    P1.4,AUTO_BAJO_NIVEL
        SETB ALIMENTAR
        SETB L_ALIMENTAR
        LJMP AUTO_SALIR
AUTO_BAJO_NIVEL: CLR  ALIMENTAR
        CLR  L_ALIMENTAR
        LJMP AUTO_SALIR
AUTO_APAGAR: SETB  ALIMENTAR
        SETB  L_ALIMENTAR
AUTO_SALIR:  RET

CIEN_MILI:  MOV   VAR1,#72
        MOV   VAR2,#180
CIEN_MILI_ESPE: DJNZ  VAR1,CIEN_MILI_ESPE
        DJNZ  VAR2,CIEN_MILI_ESPE
RET

RETARDO_1:  MOV   VAR1,#216
        MOV   VAR2,#36
RETA_REPITA: DJNZ  VAR1,RETA_REPITA
        DJNZ  VAR2,RETA_REPITA
RET

INT_SERIAL: CLR   EA
        CLR   RI
        PUSH ACC
        MOV  A,SBUF
        CJNE A,#0FH,INT_SER_MANU_AUTO
        MOV  COMANDO,#55H
        MOV  R4,#0FFH
        LJMP INT_SER_SALIR
INT_SER_MANU_AUTO: CJNE  R4,#0FFH,INT_SER_AUTO
        CJNE  A,#DIRECCION,INT_SER_SALIR
        CLR  SM2
        LCALL SERIAL_MANUAL
        LJMP INT_SER_SALIR
INT_SER_AUTO: CJNE  A,#DIRECCION,INT_SER_SALIR
        CLR  SM2
        LCALL SERIAL_AUTOMATICO
INT_SER_SALIR: POP  ACC
        SETB SM2
        SETB EA
RETI

```

```

SERIAL_AUTOMATICO: JNB RI,SERIAL_AUTOMATICO
                   CLR RI
                   MOV A,SBUF
SER_AUTO_OP1: CJNE A,#77H,SER_AUTO_OP2
                   CPL VENTILAR
                   CPL L_VENTILAR
                   LJMP SER_AUTO_SALIR
SER_AUTO_OP2: CJNE A,#88H,SER_AUTO_SALIR
SER_AUTO_WAIT: JNB RI,SER_AUTO_WAIT
                   CLR RI
                   MOV TIEMPO,SBUF
SER_AUTO_SALIR: RET

```

```

SERIAL_MANUAL: JNB RI,SERIAL_MANUAL
                   CLR RI
                   MOV A,SBUF
                   CJNE A,#69H,SER_MANU_OP2
SER_MANU_ESPE: JNB RI,SER_MANU_ESPE
                   CLR RI
                   MOV COMANDO,SBUF
                   LJMP SER_MANU_SALIR
SER_MANU_OP2: CJNE A,#0F0H,SER_MANU_SALIR
                   MOV R4,#0
SER_MANU_SALIR: RET

```

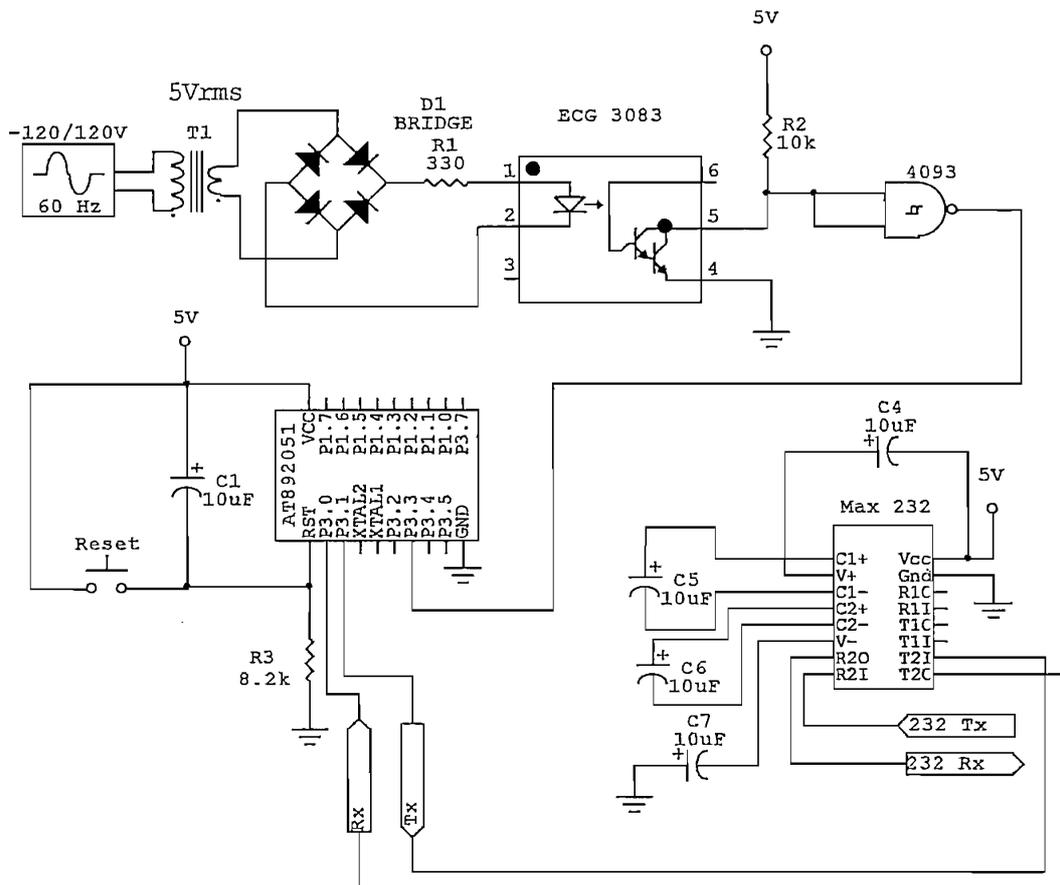
END

# **ANEXO 5**

## ANEXO 5: CIRCUITO Y PROGRAMA PARA MEDIR EL PERIODO DE LA SEÑAL DE CRUCE POR CERO

El siguiente circuito y programa sirven para determinar cual es el tiempo que pasa la señal de la figura 2.30 en su estado 1L, para de esa forma realizar los cálculos para el control de fase de la señal sinusoidal.

El circuito que se presenta a continuación, consta de un microcontrolador 89C2051 al mismo que llega la señal de cruce por cero de la figura 2.30, por medio del circuito detector de cruce por cero de la figura 2.27. Además se ha conectado el circuito integrado MAX 232 a los pines de transmisión y recepción del microcontrolador para de esa forma poder realizar una comunicación serial con el computador y poder visualizar en el monitor los datos obtenidos. El software que se ejecuta en el computador es el HYPERTERMINAL el mismo que forma parte del sistema operativo WINDOWS.



El programa que se ejecuta en el microcontrolador AT89C2051, lo primero que realiza es borrar la RAM. Después se sincroniza con la señal del detector de cruce por cero para a continuación esperar a que se produzca un flanco ascendente. Cuando se produce el flanco ascendente, se ingresa en un bucle en donde se incrementa el DPTR mientras la señal de cruce por cero permanezca en el estado de 1L. Cuando la señal cambie al estado 0L se guarda el valor del DPH y DPL en las localidades de memoria de la RAM interna, a partir de la localidad 40H. Este bucle se lo ejecuta 32 veces. Una vez que se haya ejecutado el bucle 32 veces, ahora se pasa a visualizar los datos en la pantalla del computador por lo tanto se configura los parámetros de la comunicación serial a 19200bps y 8 bits de datos. De ahí se toma uno a uno los datos almacenados desde la localidad 40H hasta la localidad 7FH. Cada vez que se toma un dato se lo separa en dos nibbles para de esa forma hacer una transformación de hexadecimal a código ASCII y poder enviarlo hacia el computador. A continuación se presenta el programa para la obtención de datos para determinar el tiempo que pasa la señal de cruce por cero en el estado 1L.

```

CPU "E:\SIDES51\8051.TBL"
HOF "INT8"
INCL "E:\SIDES51\SFR"
;*****
;TEMA: Crear una tabla que se visualice en el hyperterminal
;de los valores del cruce por cero.
;*****
;REALIZADO POR: Marco Antonio Chávez
;*****
;CONSTANTES Y VARIABLES
;*****

CRUCE_CERO: EQU P3.3
T_ASCII: EQU 0200H

;*****
;SOFTWARE
;*****

RESET: ORG 0000H
LJMP RESET1

;*****
;INTERRUPCIONES
;*****

EXTI0: ORG 0003H ;INTERRUPCION EXTERNA 0
TIMER0: ORG 000BH ;DESBORDAMIENTO TIMER 0
EXTI1: ORG 0013H ;INTERRUPCION EXTERNA 1
TIMER1: ORG 001BH ;DESBORDAMIENTO TIMER 1
SINT: ORG 0023H ;PORTICO SERIAL
EXTI2: ORG 002BH ;INT.EXT 2.o DESB.TIM.2

;*****
;VARIABLES
;*****

;ETIQUETA OPCODE OPERANDOS COMENTARIO

;*****
;INICIALIZACION DE MEMORIA RAM
;*****
ORG 0030H ;INICIO DEL PROGRAMA

```

```

RESET1: MOV  P0,#0FFH    ;INICIALIZO COMO ENTRADAS
        MOV  P1,#0FFH    ;A LOS PUERTOS
        MOV  P2,#0FFH    ;P0, P1, P2, P3
        MOV  P3,#0FFH    ;

        CLR  A
        MOV  IE,A
        MOV  IP,A
        MOV  TCON,A
        MOV  PCON,A
        MOV  PSW,A
        MOV  R0,#7FH
RESET2: MOV  @R0,A
        DJNZ R0,RESET2
        MOV  SP,#2FH

```

```

;.....
;INICIO DE PROGRAMA PRINCIPAL
;.....

```

```

PROGRAMA: MOV  R7,#32
          MOV  R0,#40H
ESPE1: JNB  CRUCE_CERO,ESPE1
ESPE2: JB   CRUCE_CERO,ESPE2
ESPE3: JNB  CRUCE_CERO,ESPE3
          MOV  DPTR,#00H
INCREMENTAR: INC  DPTR
          JB   CRUCE_CERO,INCREMENTAR
          MOV  @R0,DPH
          INC  R0
          MOV  @R0,DPL
          INC  R0
          DJNZ R7,ESPE3

```

```

          MOV  TMOD,#20H
          MOV  SCON,#50H
          MOV  TH1,#0FDH
          MOV  PCON,#80H
          SETB TR1

```

```

          MOV  R7,#64
          MOV  R0,#40H
AGAIN: MOV  R2,#2
OTRO:  MOV  A,@R0
        PUSH ACC
        ANL  A,#0F0H
        SWAP A
        LCALL HEX2ASCII
        MOV  SBUF,A
WAIT1: JNB  TI,WAIT1
        CLR  TI
        POP  ACC
        ANL  A,#0FH
        LCALL HEX2ASCII
        MOV  SBUF,A
WAIT2: JNB  TI,WAIT2
        CLR  TI
        INC  R0
        DJNZ R2,OTRO
        MOV  SBUF,#20H
WAIT3: JNB  TI,WAIT3
        CLR  TI
        DJNZ R7,AGAIN
        SJMP $

```

```

HEX2ASCII: MOV  DPTR,#T_ASCII
           MOVC A,@A+DPTR
           RET

```

```

ORG  T_ASCII
DFB  30H,31H,32H,33H,34H,35H,36H,37H,38H,39H
DFB  41H,42H,43H,44H,45H,46H
END

```

# **ANEXO 6**

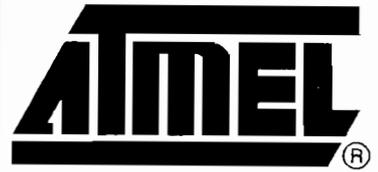
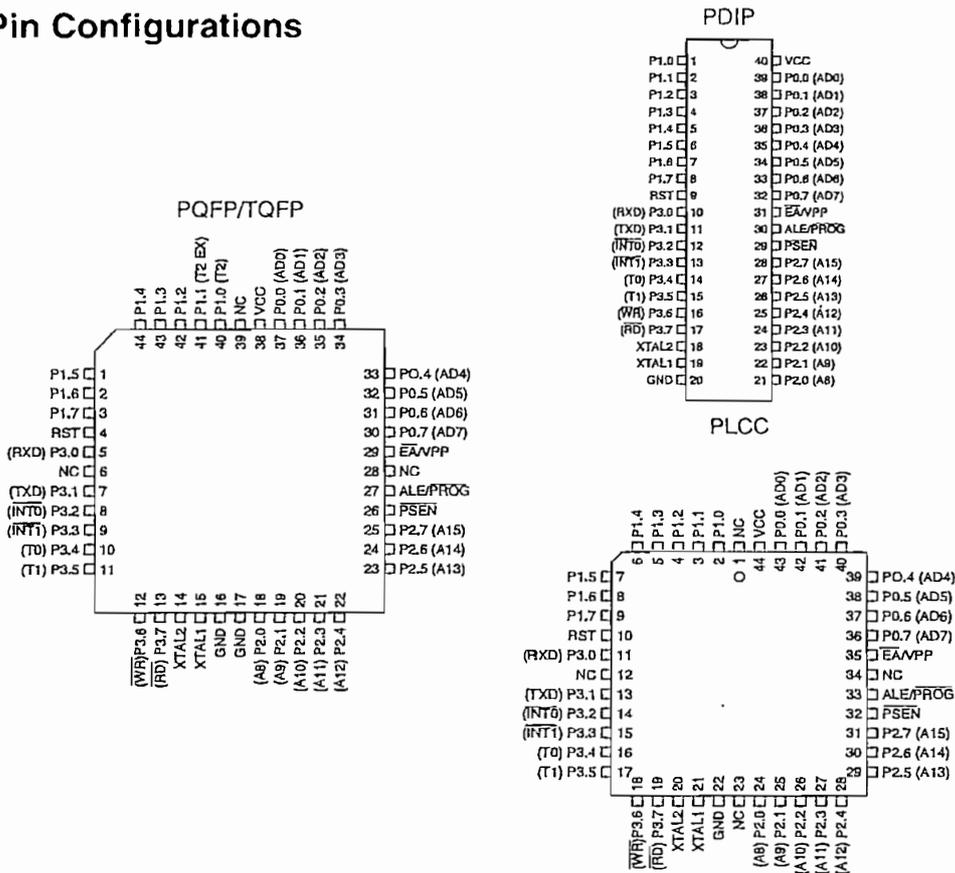
## Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

## Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

## Pin Configurations



## 8-bit Microcontroller with 4K Bytes Flash

## AT89C51



## Absolute Maximum Ratings\*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground .....	-1.0V to +7.0V
Maximum Operating Voltage .....	6.6V
DC Output Current.....	15.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 20\%$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units	
$V_{IL}$	Input Low-voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V	
$V_{IL1}$	Input Low-voltage ( $\overline{EA}$ )		-0.5	$0.2 V_{CC} - 0.3$	V	
$V_{IH}$	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V	
$V_{IH1}$	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V	
$V_{OL}$	Output Low-voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V	
$V_{OL1}$	Output Low-voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V	
$V_{OH}$	Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V	
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V	
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V	
$V_{OH1}$	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V	
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V	
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V	
$I_{IL}$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$	
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$	
$I_{LI}$	Input Leakage Current (Port 0, $\overline{EA}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$	
RRST	Reset Pull-down Resistor		50	300	$\text{K}\Omega$	
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF	
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		20	mA	
		Idle Mode, 12 MHz		5	mA	
	Power-down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$			100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$			40	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 10 mA

Maximum  $I_{OL}$  per 8-bit port: Port 0: 26 mA

Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power-down is 2V.

## Features

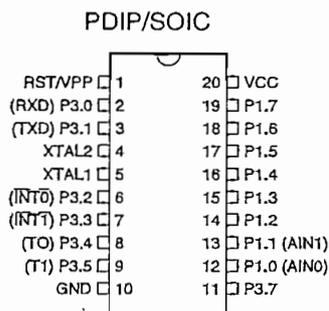
- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

## Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

## Pin Configuration



## 8-bit Microcontroller with 2K Bytes Flash

### AT89C2051



## Absolute Maximum Ratings\*

Operating Temperature .....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground .....	-1.0V to +7.0V
Maximum Operating Voltage .....	6.6V
DC Output Current.....	25.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 2.0\text{V}$  to  $6.0\text{V}$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low-voltage		-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IH}$	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low-voltage <sup>(1)</sup> (Ports 1, 3)	$I_{OL} = 20\text{ mA}$ , $V_{CC} = 5\text{V}$ $I_{OL} = 10\text{ mA}$ , $V_{CC} = 2.7\text{V}$		0.5	V
$V_{OH}$	Output High-voltage (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		V
$I_{IL}$	Logical 0 Input Current (Ports 1, 3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1, 3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-750	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
$V_{OS}$	Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$		20	mV
$V_{CM}$	Comparator Input Common Mode Voltage		0	$V_{CC}$	V
RRST	Reset Pull-down Resistor		50	300	K $\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		5/1	mA
	Power-down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		20	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 20 mA

Maximum total  $I_{OL}$  for all output pins: 80 mA

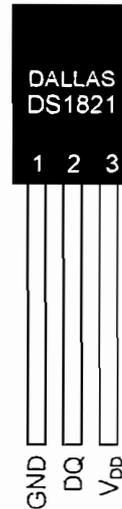
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power-down is 2V.

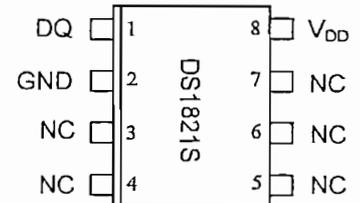
### FEATURES

- Requires no external components
- Unique 1-Wire<sup>®</sup> interface requires only one port pin for communication
- Operates over a -55°C to +125°C (67°F to +257°F) temperature range
- Functions as a standalone thermostat with user-definable trip-points
- Provides 8-bit (1°C resolution) centigrade temperature measurements
- Accuracy is ±1°C over 0°C to +85°C range
- Converts temperature to a digital word in 1 second (max)
- Available in 3-pin PR35 and 8-pin SOIC packages
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

### PIN ASSIGNMENT



(BOTTOM VIEW)  
 PR35  
 (DS1821)



8-pin 208-mil SOIC  
 (DS1821S)

### PIN DESCRIPTION

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

### DESCRIPTION

The DS1821 can function as a standalone thermostat with user-programmable trip-points or as 8-bit temperature sensor with a 1-wire digital interface. The thermostat trip-points are stored in nonvolatile memory, so DS1821 units can be programmed prior to system insertion for true standalone operation. The DS1821 has an operating temperature range of -55°C to +125°C and is accurate to ±1°C over a range of 0°C to +85°C. Communication with the DS1821 is accomplished through the open-drain DQ pin; this pin also serves as the thermostat output.

**DETAILED PIN DESCRIPTIONS Table 1**

PR35	8-PIN SOIC*	SYMBOL	DESCRIPTION
1	2	GND	Ground pin.
2	1	DQ	Open drain data input/output pin – 1-wire operation; Open drain thermostat output pin – thermostat operation.
3	8	V <sub>DD</sub>	Power supply pin.

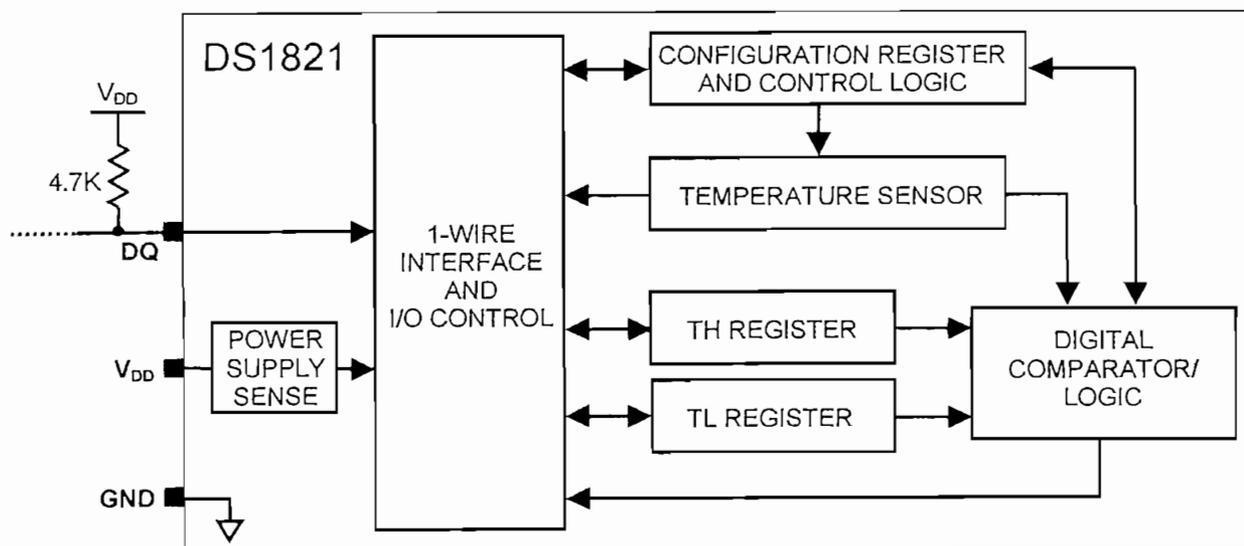
\*All pins not specified in this table are “No Connect” pins.

**OVERVIEW**

Figure 1 shows a block diagram of the DS1821 and pin descriptions are given in Table 1. The DS1821 can operate as a standalone thermostat with user-programmable trip-points or as 8-bit temperature sensor with a 1-wire digital interface. The open-drain DQ pin functions as the thermostat output for thermostat operation and as the data I/O pin for 1-wire communications. The 1-wire interface provides user access to the nonvolatile (EEPROM) thermostat trip-point registers (T<sub>H</sub> and T<sub>L</sub>), the status/configuration register, and the temperature register.

When configured as standalone thermostat, temperature conversions start immediately at power-up. In this mode, the DQ pin becomes active when the temperature of the DS1821 exceeds the limit programmed into the T<sub>H</sub> register, and remains active until the temperature drops below the limit programmed into the T<sub>L</sub> register.

The DS1821 uses Dallas’ exclusive 1-wire bus protocol that implements bus communication with one control signal. This system is explained in detail in the 1-WIRE BUS SYSTEM section of this datasheet.

**DS1821 BLOCK DIAGRAM Figure 1****TEMPERATURE SENSOR FUNCTIONALITY**

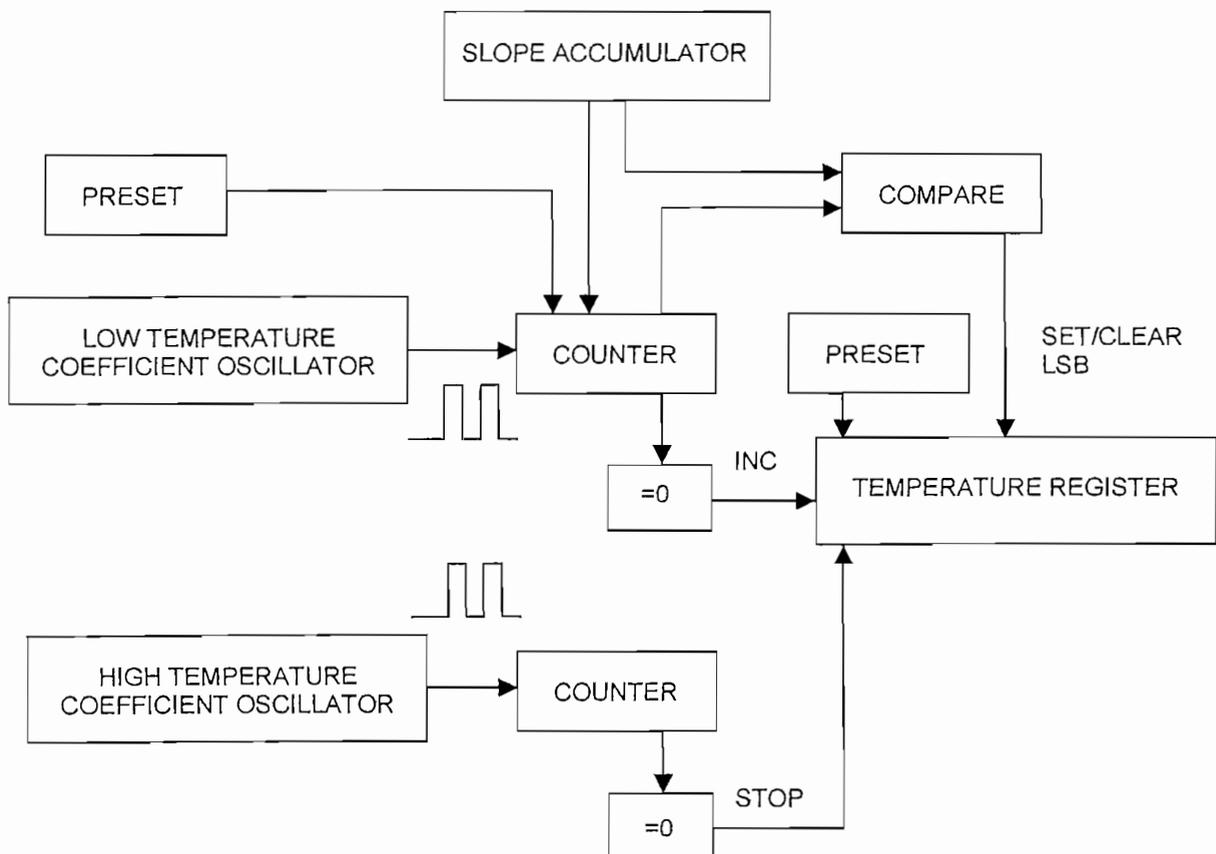
The core functionality of the DS1821 is its proprietary direct-to-digital temperature sensor, which provides 8-bit (1°C increment) centigrade temperature readings over the range of -55°C to +125°C.

A block diagram of the temperature measurement circuitry is shown in Figure 2. This circuit measures the temperature by counting the number of clock cycles generated by an oscillator with a low temperature coefficient (temp-co) during a gate period determined by a high temp-co oscillator. The low temp-co

counter is preset with a base count that corresponds to  $-55^{\circ}\text{C}$ . If the counter reaches 0 before the gate period is over, the temperature register, which is preset to  $-55^{\circ}\text{C}$ , is incremented by one degree, and the counter is again preset with a starting value determined by the slope accumulator circuitry. The preset counter value is unique for every temperature increment and compensates for the parabolic behavior of the oscillators over temperature.

At this time, the counter is clocked again until it reaches 0. If the gate period is not over when the counter reaches 0, the temperature register is incremented again. This process of presetting the counter, counting down to zero, and incrementing the temperature register is repeated until the counter takes less time to reach zero than the duration of the gate period of the high temp-co oscillator. When this iterative process is complete, the value in the temperature register will indicate the centigrade temperature of the device.

## TEMPERATURE MEASURING CIRCUITRY Figure 2



## OPERATING MODES

The DS1821 has two operating modes: 1-wire mode and thermostat mode. The power-up operating mode is determined by the user-programmable T/R bit in the status/configuration register: if T/R = 0 the device powers-up in 1-wire mode, and if T/R = 1 the device powers-up in thermostat mode. The T/R bit is stored in nonvolatile memory (EEPROM), so it will retain its value when the device is powered down.

### 1-WIRE MODE

The DS1821 arrives from the factory in 1-wire mode (T/R = 0). In this mode, the DQ pin of the DS1821 is configured as a 1-wire port for communication with a microprocessor using the protocols described in

the 1-WIRE BUS SYSTEM section of this datasheet. These communications can include reading and writing the high and low thermostat trip-point registers ( $T_H$  and  $T_L$ ) and the configuration register, and reading the temperature, counter, and slope accumulator registers. Also in this mode, the microprocessor can initiate and stop temperature measurements as described in the OPERATION – MEASURING TEMPERATURE section of this datasheet.

The  $T_H$  and  $T_L$  registers and certain bits (THF, TLF, T/R, POL and 1SHOT) in the status/configuration register are stored in nonvolatile EEPROM memory, so they will retain data when the device is powered down. This allows these registers to be pre-programmed when the DS1821 is to be used as a standalone thermostat. Writes to these nonvolatile registers can take up to 10ms. To avoid data corruption, no writes to nonvolatile memory should be initiated while a write to nonvolatile memory is in progress. Nonvolatile write status can be monitored by reading the NVB bit in the status/configuration register: NVB = 0 – a write to EEPROM memory is in progress, NVB = 1 – nonvolatile memory is idle.

## THERMOSTAT MODE

In thermostat mode ( $T/R = 1$ ), the DS1821 can operate as a standalone thermostat that triggers according to the  $T_H$  and  $T_L$  trip-points programmed while the device was in 1-wire mode. In thermostat mode the DS1821 powers-up performing continuous temperature conversions, and the DQ pin acts as the thermostat output. Detailed operation of the thermostat output is provided in the OPERATION – STANDALONE THERMOSTAT section of this datasheet.

Communications can be re-established with the DS1821 while it is in thermostat mode by pulling  $V_{DD}$  to 0V while the DQ line is held high, and then toggling the DQ line low 16 times as shown in Figure 12. This temporarily places the DS1821 in 1-wire mode, allowing microprocessor communication with the DS1821 via the DQ pin. At this time any I/O function can be performed, such as reading/writing the  $T_H$ ,  $T_L$  or configuration registers or reading the temperature register. To return to thermostat mode, the same procedure can be performed (pulling  $V_{DD}$  to 0V while the DQ line is held high, and then clocking the DQ line 16 times) or the power can be cycled. Note that temporarily putting the DS1821 into 1-wire mode does not change the power-up mode of the device; this can only be changed by rewriting the T/R bit in the status/configuration register. Also note that holding both  $V_{DD}$  and DQ low for more than approximately 10 seconds will cause the DS1821 to be powered down.

## OPERATION – MEASURING TEMPERATURE

DS1821 output temperature data is calibrated in degrees centigrade and is stored in two's complement format in the 1-byte (8-bit) temperature register (see Figure 3), which the user can access when the DS1821 is in 1-wire mode ( $T/R = 0$  in the status/configuration register). The sign bit (S) indicates 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. For Fahrenheit measurements, a lookup table or conversion routine must be used.

The DS1821 can be configured by the user to take continuous temperature measurements (continuous conversion mode) or single measurements (one-shot mode). The desired configuration can be achieved by setting the nonvolatile 1SHOT bit in the status/configuration register: 1SHOT = 0 – continuous conversion mode, 1SHOT = 1 – one-shot mode. Note that the 1SHOT setting only controls the operation of the device in 1-wire mode; in thermostat mode, continuous temperature conversions are started automatically at power-up.

In continuous conversion mode, the Start Convert T [EEh] command initiates continuous temperature conversions, which can be stopped using the Stop Convert T [22h] command. In one-shot mode the Start Convert T [EEh] command initiates a single temperature conversion after which the DS1821 returns to a low-power standby state. In this mode, the microprocessor can monitor the DONE bit in the

configuration register to determine when the conversion status: DONE = 0 — conversion in progress, DONE = 1 — conversion complete. The DONE bit does not provide conversion status in continuous conversion mode since measurements are constantly in progress (i.e., DONE will always be 0).

### TEMPERATURE, T<sub>H</sub> and T<sub>L</sub> REGISTER FORMAT Figure 3

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	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>

### TEMPERATURE/DATA RELATIONSHIP Table 2

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C*	0111 1101	7Dh
+85°C	0101 0101	55h
+25°C	0001 1001	19h
0°C	0000 0000	00h
-1°C	1111 1111	FFh
-25°C	1110 0111	E7h
-55°C	1100 1001	C9h

### HIGH-RESOLUTION TEMPERATURE READINGS

The user can calculate temperature values with higher than 8-bit resolution using the data remaining in the counter and slope accumulator when the temperature conversion is complete. To do this the user must first read the temperature from the 8-bit temperature register. This value is called TEMP\_READ in the high-resolution equation (see Eq. 1). The 9-bit counter value must then be obtained by issuing the Read Counter [A0h] command. This value is the count remaining in the counter at the end of the gate period and is called COUNT\_REMAIN in Eq. 1. Next the Load Counter [41h] command must be issued, which loads the 9-bit slope accumulator value into the counter register. The slope accumulator value (called COUNT\_PER\_C in Eq. 1) can then be read from the counter by again issuing the Read Counter [A0h] command. The slope accumulator value is called “COUNT\_PER\_C” because it represents the number of counts needed for an accurate measurement at a given temperature (i.e., the counts per degree C). The high-resolution temperature can then be calculated using Eq. 1:

$$Eq. 1) \quad TEMPERATURE = TEMP\_READ - 0.5 + \frac{(COUNT\_PER\_C - COUNT\_REMAIN)}{COUNT\_PER\_C}$$

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”.

## OPERATION – THERMOSTAT

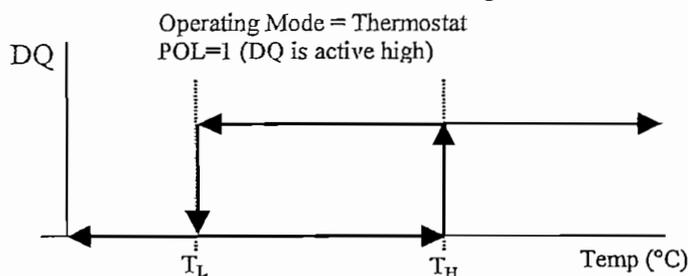
When the DS1821 is in thermostat mode ( $T/R = 1$  in the status/configuration register), temperature conversions are performed continuously beginning at power-up (regardless of the value of the 1SHOT bit), and the DQ pin serves as the thermostat output. The DQ output will become active when the temperature of the DS1821 exceeds the user-defined limit in the  $T_H$  register, and will remain active until the temperature drops below the user-defined limit in the  $T_L$  register as illustrated in Figure 4. Thus, the user can select  $T_H$  and  $T_L$  to provide the desired amount of thermostat output hysteresis.

The user-defined 8-bit centigrade trip-point values ( $T_H$  and  $T_L$ ) must be stored in two's complement format as shown in Figure 3. The sign bit (S) indicates if the temperature is positive or negative; for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . The non-volatile  $T_H$  and  $T_L$  registers must be programmed when the DS1821 is in 1-wire mode as explained in the OPERATING MODES section of this datasheet. The DS1821 can be temporarily switched from thermostat mode to 1-wire mode to change the  $T_H$  and  $T_L$  values as also explained in the OPERATING MODES section.

The polarity (i.e., the active state) of the DQ output is user-selectable with the nonvolatile POL bit in the status/configuration register. DQ is active-high when  $POL = 1$ , and DQ is active-low when  $POL = 0$ .

Two bits in the status/configuration register, THF and TLF, provide additional thermostatic information. The value of these bits is normally 0. The THF (temperature high flag) bit will be set to 1 if the measured temperature is ever greater than the value in the  $T_H$  register and will remain a 1 until the user rewrites the bit with a 0. The THL (temperature low flag) bit will be set to 1 if the temperature is ever lower than the value in the  $T_L$  register and will remain a 1 until the user rewrites the bit with a 0. These bits provide a record of the device temperature relative to the thermostat trip-points over a period of time. They are stored in nonvolatile memory, so the data stored in THF and TLF can be analyzed after any number of power cycles. The THF and THL bits function in both 1-wire and thermostat mode.

### DQ OPERATION IN THERMOSTATE MODE Figure 4



## STATUS/CONFIGURATION REGISTER

The status/configuration register provides information to the user about conversion status, EEPROM activity and thermostat activity. It also allows the user to program various DS1821 options such as power-up operating mode, thermostat output polarity and conversion mode. The status/configuration register is arranged as shown in Figure 5 and detailed descriptions of each bit are provided in Table 3. Note that the THF, THL, T/R, POL and 1SHOT bits are stored in nonvolatile memory (EEPROM).

### CONFIGURATION REGISTER Figure 5

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
DONE	1	NVB	THF*	THL*	T/R*	POL*	1SHOT*

\*Stored in EEPROM

**CONFIGURATION REGISTER BIT DESCRIPTIONS Table 3**

Bit Name (User Access)	Functional Description
DONE — Temperature Conversion Done (Read Only)	DONE = 0 — Temperature conversion is in progress. DONE = 1 — Temperature conversion is complete.
NVB — Non-volatile Memory Busy (Read Only)	NVB = 0 — Nonvolatile memory is not busy. NVB = 1 — A write to EEPROM memory is in progress
THF* — Temperature High Flag (Read/Write)	THF = 0 — The measured temperature has not exceeded the value stored in the $T_H$ register. THF = 1 — At some point in time the measured temperature has been higher than the value stored in the $T_H$ register. THF will remain a 1 until it is over-written with a 0 by the user.
TLF* — Temperature Low Flag (Read/Write)	TLF = 0 — The measured temperature has not been lower than the value stored in the $T_L$ register. TLF = 1 — At some point in time the measured temperature has been lower than the value stored in the $T_L$ register. TLF will remain a 1 until it is over-written with a 0 by the user.
T/R* — Power-up Operating Mode (Read/Write)	T/R = 0 — DS1821 powers up in 1-wire mode. T/R = 1 — DS1821 powers up in thermostat mode.
POL* — Thermostat Output (DQ) Polarity (Read/Write)	POL = 0 — Thermostat output (DQ) is active low. POL = 1 — Thermostat output (DQ) is active high.
1SHOT* — Conversion Mode (Read/Write)	1SHOT = 0 — Continuous conversion mode. The Start Convert T [EEh] command initiates continuous temperature conversions, which can be stopped with the Stop Convert T [22h] command. 1SHOT = 1 — One-shot mode. The Start Convert T [EEh] command initiates a single temperature conversion after which the DS1821 returns to a low-power standby state.

\*Stored in EEPROM

## 1-WIRE BUS SYSTEM

The 1-wire bus system uses a single bus master (i.e., a microprocessor) to control slave devices. The DS1821 functions as a slave device when it is used in 1-wire mode; however, since the DS1821 is not addressable or multi-droppable, a single 1-wire-mode DS1821 must be the only slave device on the bus. All data and commands are transmitted least significant bit first over the 1-wire bus.

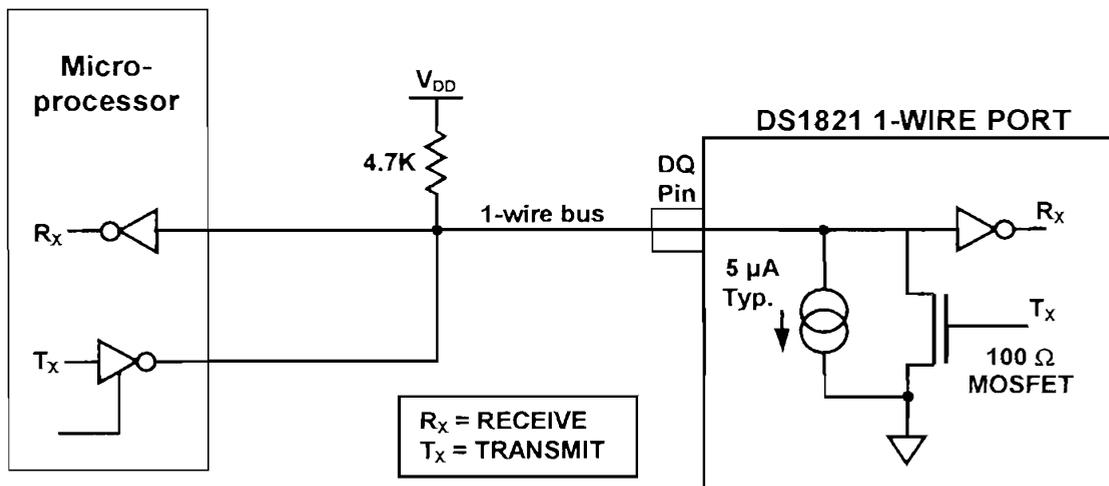
The following discussion of the 1-wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-wire signaling (signal types and timing).

## HARDWARE CONFIGURATION

The 1-wire bus has by definition only a single data line. Each device (in this case, the master and one DS1821) 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 that the bus is available for use by the other device. The 1-wire port of the DS1821 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 6.

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, the DS1821 will be reset.

## HARDWARE CONFIGURATION Figure 6



## TRANSACTION SEQUENCE

The transaction sequence for accessing the DS1821 via the 1-wire port is as follows:

- Initialization
- DS1821 Function Command
- Data Transmitted/Received

## 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 a presence pulse transmitted by the DS1821. The presence pulse lets the bus master know that the DS1821 is on the bus and ready to operate. Timing for the reset and presence pulses is detailed in the 1-WIRE SIGNALING section.

## DS1821 FUNCTION COMMANDS

The DS1821 function commands in this section allow the master to communicate with and configure the DS1821. The DS1821 function commands are summarized in Table 4.

### READ TEMPERATURE [AAh]

Provides read access to the 1-byte temperature register.

### START CONVERT T [EEh]

Initiates temperature conversions. If the part is in one-shot mode (1SHOT = 1), only one conversion will be performed. If it is in continuous mode (1SHOT = 0), continuous conversions will be performed until a Stop Convert T command is received.

### STOP CONVERT T [22h]

Stops temperature conversions when the device is in continuous conversion mode (1SHOT = 0). This opcode has no function if the device is in one-shot mode (1SHOT = 1).

### WRITE TH [01h]

### WRITE TL [02h]

Provides write access to the 8-bit  $T_H$  and  $T_L$  registers, respectively.

### READ TH [A1h]

### READ TL [A2h]

Provides read access to the 8-bit  $T_H$  and  $T_L$  registers, respectively.

### WRITE STATUS [0Ch]

Provides write access to the 8-bit status/configuration register.

### READ STATUS [ACh]

Provides read access to the 8-bit status/configuration register.

### READ COUNTER [A0h]

Provides read access to data in the 9-bit counter register for use in high-resolution temperature calculations. This is explained in detail in the HIGH-RESOLUTION TEMPERATURE READINGS section.

### LOAD COUNTER [41h]

Loads the 9-bit data from the slope accumulator register into the counter register so that it can be accessed using the Read Counter [A0h] command. Use of the Load Counter command is explained in detail in the HIGH-RESOLUTION TEMPERATURE READINGS section.

DS1821 FUNCTION COMMAND SET Table 4

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued
<b>TEMPERATURE CONVERSION COMMANDS</b>			
Read Temperature	Reads last converted temperature value from temperature register.	AAh	Master receives 8-bit temperature value from DS1821.
Start Convert T	Initiates temperature conversions.	EEh	None
Stop Convert T	Halts temperature conversions.	22h	None
<b>THERMOSTAT and STATUS/CONFIGURATION COMMANDS</b>			
Write TH	Writes data to the T <sub>H</sub> register.	01h	Master transmits 8-bit T <sub>H</sub> value to DS1821.
Write TL	Writes data to the T <sub>L</sub> register.	02h	Master transmits 8-bit T <sub>L</sub> value to DS1821.
Read TH	Reads data from the T <sub>H</sub> register.	A1h	Master receives 8-bit T <sub>H</sub> value from DS1821.
Read TL	Reads data from the T <sub>L</sub> register.	A2h	Master receives 8-bit T <sub>L</sub> value from DS1821.
Write Status	Writes data to the status/configuration register.	0Ch	Master transmits 8-bit status/configuration value to DS1821.
Read Status	Reads data from the status/configuration register.	ACh	Master receives 8-bit status/configuration value from DS1821.
<b>HIGH-RESOLUTION COMMANDS</b>			
Read Counter	Reads data from the counter register	A0h	Master receives 9-bit counter value from DS1821.
Load Counter	Loads slope accumulator data into the counter register	41h	None

## 1-WIRE SIGNALING

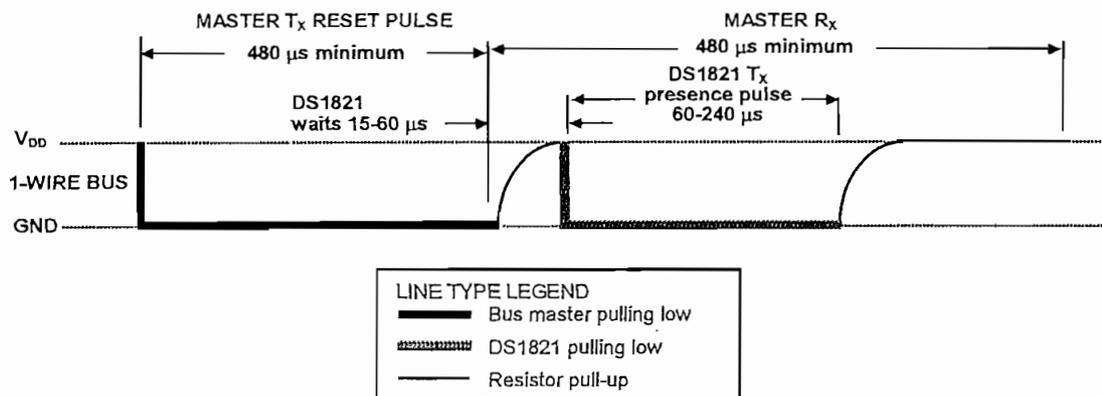
The DS1821 uses a strict 1-wire communication protocol to insure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

## INITIALIZATION PROCEDURE: RESET AND PRESENCE PULSES

All communication with the DS1821 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS1821. This is illustrated in Figure 7. When the DS1821 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 given an appropriate function command.

During the initialization sequence the bus master transmits (T<sub>X</sub>) the reset pulse by pulling the 1-wire bus low for a minimum of 480 μs. The bus master then releases the bus and goes into receive mode (R<sub>X</sub>). When the bus is released, the 5k pullup resistor pulls the 1-wire bus high. When the DS1821 detects this rising edge, it waits 15–60 μs and then transmits a presence pulse by pulling the 1-wire bus low for 60–240 μs.

## INITIALIZATION TIMING Figure 7



## READ/WRITE TIME SLOTS

The bus master writes data to the DS1821 during write time slots and reads data from the DS1821 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 DS1821 and a Write 0 time slot to write a logic 0 to the DS1821. All write time slots must be a minimum of 60 μs in duration with a minimum of a 1 μs 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 8).

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 μs. 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 μs).

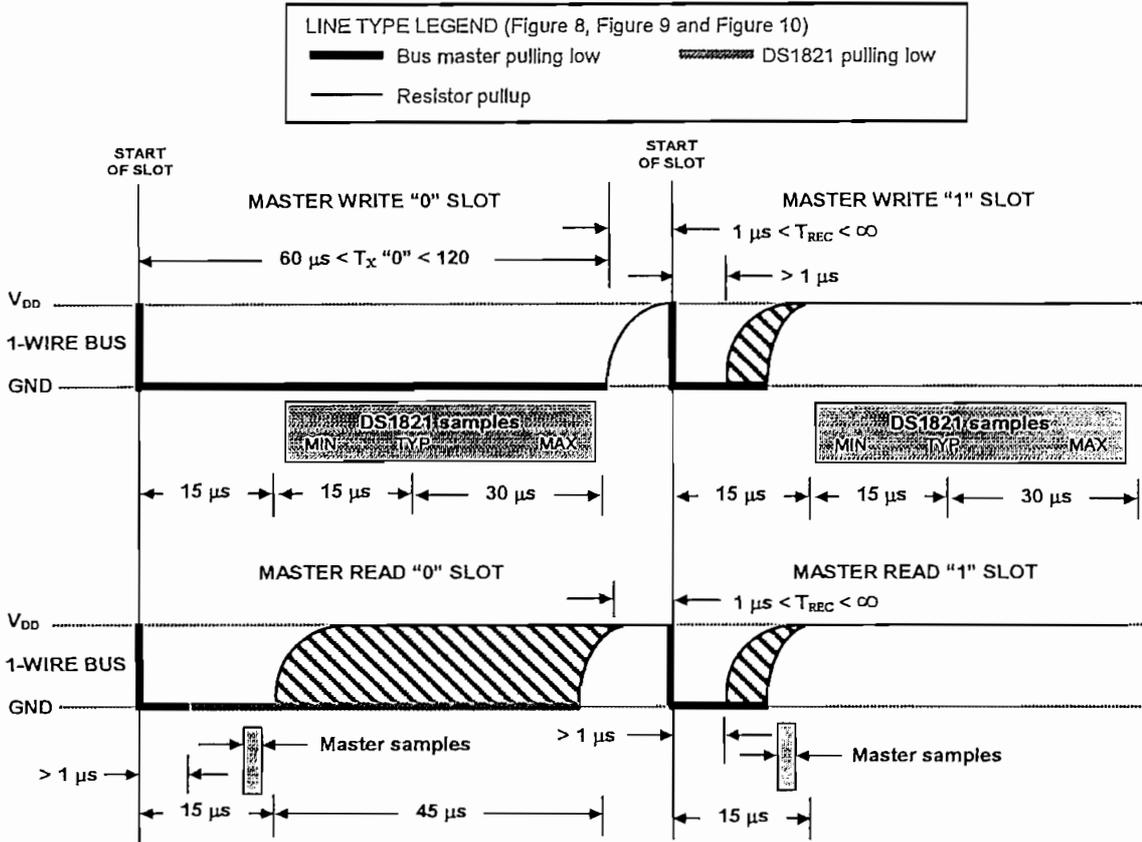
The DS1821 samples the 1-wire bus during a window that lasts from 15 μs to 60 μs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS1821. If the line is low, a 0 is written to the DS1821.

## READ TIME SLOTS

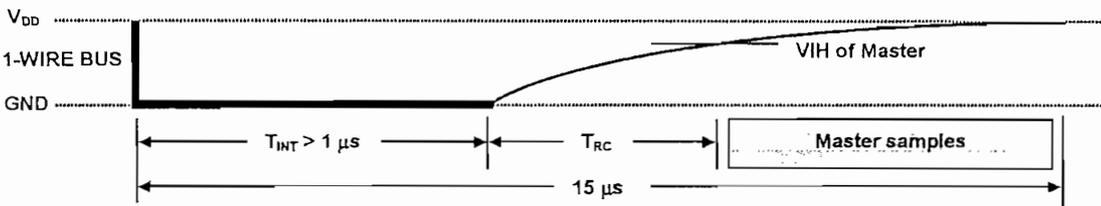
The DS1821 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a read command (e.g., Read Temperature [AAh]), so that the DS1821 can provide the requested data. All read time slots must be a minimum of 60 μs in duration with a minimum of a 1 μs 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 μs and then releasing the bus (see Figure 8). After the master initiates the read time slot, the DS1821 will begin transmitting a 1 or 0 on bus. The DS1821 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS1821 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 DS1821 is valid for 15 μs 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 μs from the start of the slot.

Figure 9 illustrates that the sum of  $T_{INT}$ ,  $T_{RC}$ , and  $T_{SAMPLE}$  must be less than  $15\ \mu\text{s}$  for a read time slot. Figure 10 shows that system timing margin is maximized by keeping  $T_{INT}$  and  $T_{RC}$  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.

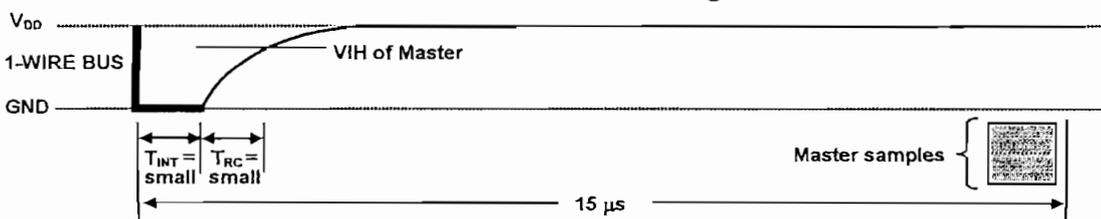
**READ/WRITE TIME SLOT TIMING DIAGRAM Figure 8**



**DETAILED MASTER READ 1 TIMING Figure 9**



**RECOMMENDED MASTER READ 1 TIMING Figure 10**



## RELATED APPLICATION NOTES

The following Application Notes pertain to the DS1821. 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 67: "Applying and Using the DS1620 in Temperature Control Applications"

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

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

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

## DS1821 OPERATION EXAMPLE

In this example, the master device programs the DS1821 with  $T_L = +10^\circ\text{C}$  and  $T_H = +40^\circ\text{C}$  and verifies that the data has been saved correctly. The master then programs the status/configuration register so that the device will power-up in thermostat mode ( $T/R = 1$ ) and the thermostat output will have active high polarity ( $POL = 1$ ).

MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Master issues reset pulse.
RX	Presence	DS1821 responds with presence pulse.
TX	01h	Master issues Write TH command.
TX	28h	Master sends data for $T_H = +40^\circ\text{C}$ .
TX	Reset	Master issues reset pulse.
RX	Presence	DS1821 responds with presence pulse.
TX	02h	Master issues Write TL command.
TX	0Ah	Master sends data for $T_L = +10^\circ\text{C}$ .
TX	Reset	Master issues reset pulse.
RX	Presence	DS1821 responds with presence pulse.
TX	A1h	Master issues Read TH command.
RX	28h	Master reads stored $T_H$ value to verify data.
TX	Reset	Master issues reset pulse.
RX	Presence	DS1821 responds with presence pulse.
TX	A2h	Master issues Read TL command.
RX	0Ah	Master reads stored $T_L$ value to verify data.
TX	Reset	Master issues reset pulse.
RX	Presence	DS1821 responds with presence pulse.
TX	0Ch	Master issues Write Status command.
TX	06h	Master sends status/configuration data to the DS1821 with $T/R = 1$ (thermostat mode at power-up) and $POL = 1$ (active high thermostat output).
—	—	Power is cycled; DS1821 powers-up in thermostat mode.

**ABSOLUTE MAXIMUM RATINGS\***

Voltage on any pin relative to ground	-0.5V to +7.0V
Operating temperature	-55°C to +125°C
Storage temperature	-55°C to +125°C
Soldering temperature	See-JTD-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;  $V_{DD}=2.7V$  to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	$V_{DD}$		+2.7		+5.5		1
Thermometer Error	$t_{ERR}$	0°C to +85°C $V_{DD}=3.6V$ to 5.5V			±1	°C	2,3,4
		-55°C to +125°C $V_{DD}=3.6V$ to 5.5V		See Typical Curve (Figure 11)			
DQ Logic Low	$V_{IL}$		-0.3		+0.8	V	1,5
DQ Logic High	$V_{IH}$		+2		The lower of +5.5 or $V_{DD} + 0.3$	V	1,6
Sink Current	$I_L$	$V_{DQ} = 0.4V$ $V_{DD} = 3.6V$ to 5.5V	4			mA	1
Standby Current	$I_Q$	-55°C to +85°C		1	3	μA	7
Active Current	$I_{DD}$	$V_{DD} = 5V$		500	1000	μA	8
DQ Input Current	$I_{DQ}$			5		μA	9

**NOTES:**

- All voltages are referenced to ground.
- Thermometer error reflects the sensor accuracy as tested during calibration.
- See typical performance curve in Figure 11 for specification limits outside the 0°C to +85°C range.
- For  $T < 0^\circ C$ , accuracy degrades by 0.5°C/V for  $V_{DD} < 4.3V$ .
- Logic low voltages are specified at a sink current of 4 mA.
- Logic high voltages are specified at a source current of 1 mA.
- Standby current is typically 5 μA at 125°C.
- Active current refers to supply current during active temperature conversions or EEPROM writes.
- DQ line is high ("hi-Z" state).

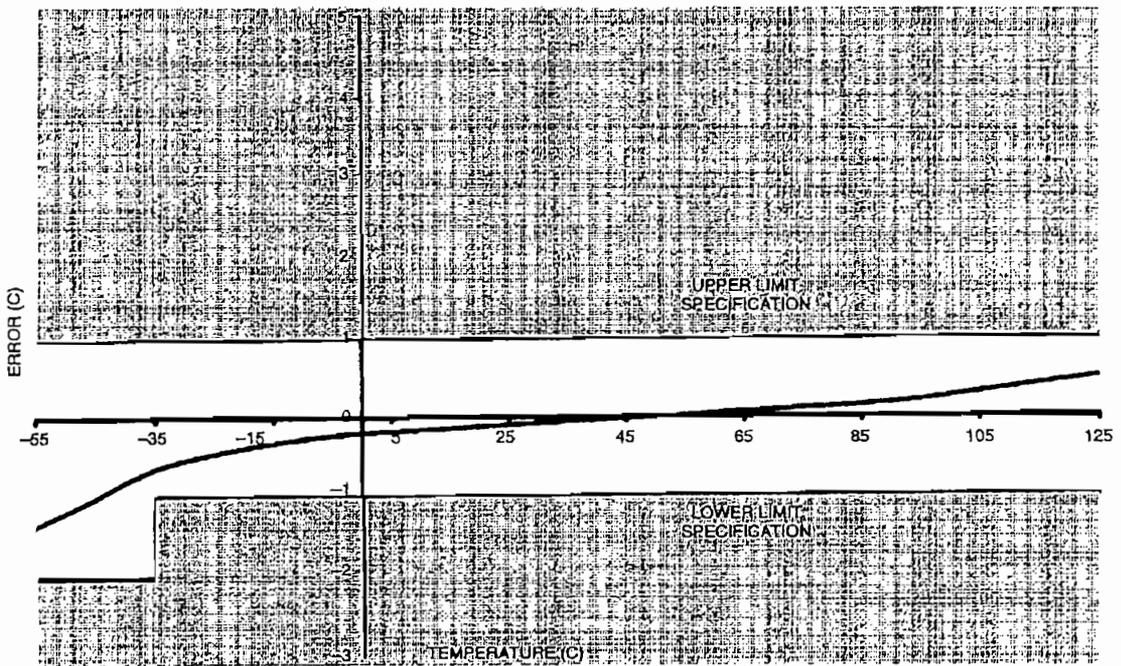
**AC ELECTRICAL CHARACTERISTICS:** (-55°C to +125°C;  $V_{DD}=3.6V$  to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	$t_{CONV}$		0.4	1.0	s	
EEPROM Write Time	$t_{WR}$		10	50	ms	
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
$V_{DD}$ Low to Mode Toggle Clock Low	$t_{PC}$	100			ns	1,3
Mode Toggle Clock 16 High to $V_{DD}$ High	$t_{CP}$	100			ns	1
Mode Toggle Clock Pulse Low Time	$t_{CL}$	0.1		10	$\mu s$	1
Mode Toggle Clock Pulse High Time	$t_{CH}$	0.1			$\mu s$	1
Mode Toggle Clock High-to-Low or Low-to-High Transition Time	$t_T$			100	ns	1
Capacitance	$C_{IN/OUT}$			25	pF	

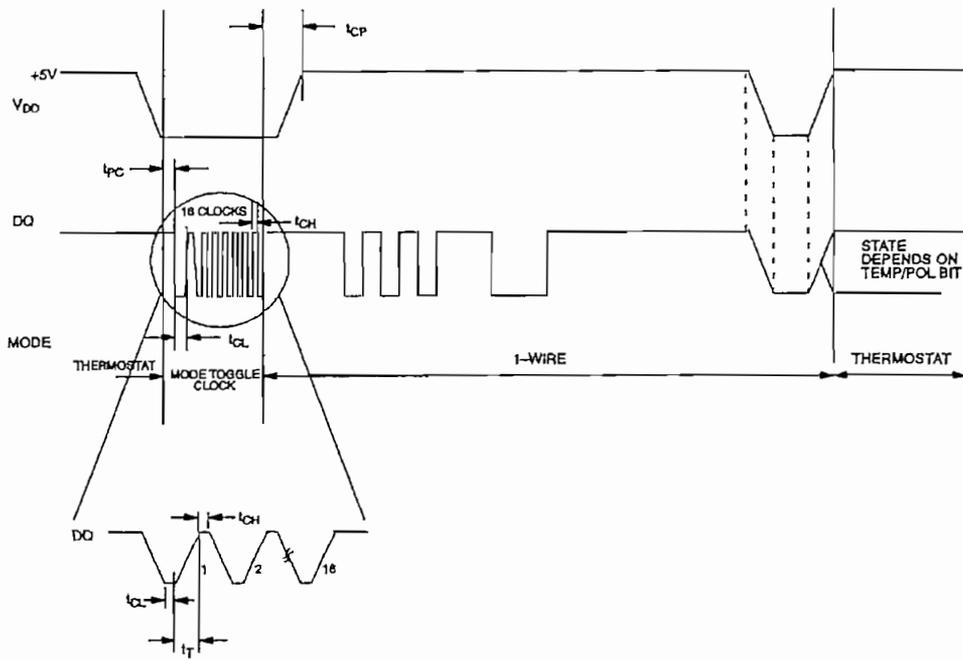
**NOTES:**

1. Refer to timing diagrams in Figure 13.
2. If  $t_{RSTL} > 960 \mu s$ , a power-on-reset may occur.
3. Time required for part to disable thermostat output.

TYPICAL PERFORMANCE CURVE Figure 11

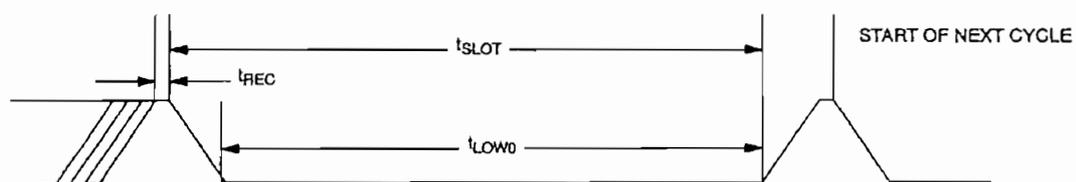


MODE TOGGLE TIMING WHEN T/R = 1 Figure 12

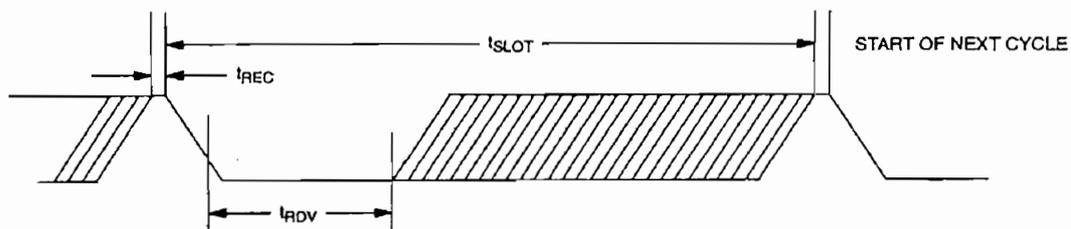


## TIMING DIAGRAMS Figure 13

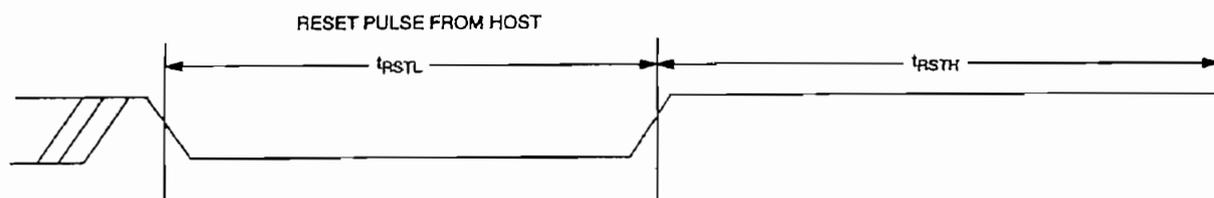
## 1-WIRE WRITE ZERO TIME SLOT



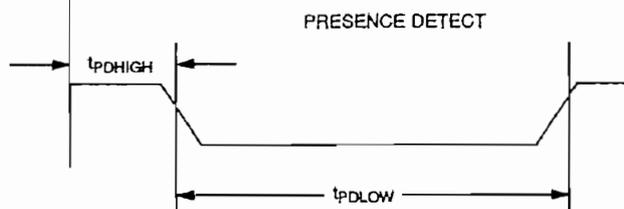
## 1-WIRE READ ZERO TIME SLOT



## 1-WIRE RESET PULSE



## 1-WIRE PRESENCE DETECT





## 6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds All JEDEC Registered Specifications
- *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.*

### Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

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

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

#### INPUT LED

Reverse Voltage	$V_R$	3	Volts
Forward Current — Continuous	$I_F$	60	mA
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Output Detector Derate above $25^\circ\text{C}$	$P_D$	120	mW
		1.41	mW/ $^\circ\text{C}$

#### OUTPUT TRANSISTOR

Collector-Emitter Voltage	$V_{CEO}$	30	Volts
Emitter-Collector Voltage	$V_{ECO}$	7	Volts
Collector-Base Voltage	$V_{CBO}$	70	Volts
Collector Current — Continuous	$I_C$	150	mA
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Input LED Derate above $25^\circ\text{C}$	$P_D$	150	mW
		1.76	mW/ $^\circ\text{C}$

#### TOTAL DEVICE

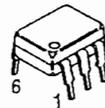
Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 sec Duration)	$V_{ISO}$	7500	Vac(pk)
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	250	mW
		2.94	mW/ $^\circ\text{C}$
Ambient Operating Temperature Range <sup>(2)</sup>	$T_A$	-55 to +100	$^\circ\text{C}$
Storage Temperature Range <sup>(2)</sup>	$T_{stg}$	-55 to +150	$^\circ\text{C}$
Soldering Temperature (10 sec, 1/16" from case)	$T_L$	260	$^\circ\text{C}$

1. Isolation surge voltage 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.  
Global Optoisolator is a trademark of Motorola, Inc.

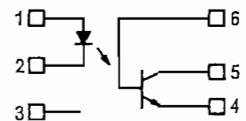
**4N25**  
**4N25A**  
**4N26**  
[CTR = 20% Min]  
**4N27**  
**4N28**  
[CTR = 10% Min]

### STYLE 1 PLASTIC



STANDARD THRU HOLE  
CASE 730A-04

### SCHEMATIC



- PIN 1. LED ANODE  
2. LED CATHODE  
3. N.C.  
4. EMITTER  
5. COLLECTOR  
6. BASE

# 4N25 4N25A 4N26 4N27 4N28

## ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ unless otherwise noted)<sup>(1)</sup>

Characteristic	Symbol	Min	Typ <sup>(1)</sup>	Max	Unit	
<b>INPUT LED</b>						
Forward Voltage ( $I_F = 10\text{ mA}$ )	$V_F$	$T_A = 25^\circ\text{C}$	—	1.15	1.5	Volts
		$T_A = -55^\circ\text{C}$	—	1.3	—	
		$T_A = 100^\circ\text{C}$	—	1.05	—	
Reverse Leakage Current ( $V_R = 3\text{ V}$ )	$I_R$	—	—	100	$\mu\text{A}$	
Capacitance ( $V = 0\text{ V}$ , $f = 1\text{ MHz}$ )	$C_J$	—	18	—	pF	

## OUTPUT TRANSISTOR

Collector–Emitter Dark Current ( $V_{CE} = 10\text{ V}$ , $T_A = 25^\circ\text{C}$ )	4N25,25A,26,27 4N28	$I_{CEO}$	—	1	50	nA
	( $V_{CE} = 10\text{ V}$ , $T_A = 100^\circ\text{C}$ )	All Devices	$I_{CEO}$	—	1	—
Collector–Base Dark Current ( $V_{CB} = 10\text{ V}$ )		$I_{CBO}$	—	0.2	—	nA
Collector–Emitter Breakdown Voltage ( $I_C = 1\text{ mA}$ )		$V_{(BR)CEO}$	30	45	—	Volts
Collector–Base Breakdown Voltage ( $I_C = 100\text{ }\mu\text{A}$ )		$V_{(BR)CBO}$	70	100	—	Volts
Emitter–Collector Breakdown Voltage ( $I_E = 100\text{ }\mu\text{A}$ )		$V_{(BR)ECO}$	7	7.8	—	Volts
DC Current Gain ( $I_C = 2\text{ mA}$ , $V_{CE} = 5\text{ V}$ )		$h_{FE}$	—	500	—	—
Collector–Emitter Capacitance ( $f = 1\text{ MHz}$ , $V_{CE} = 0$ )		$C_{CE}$	—	7	—	pF
Collector–Base Capacitance ( $f = 1\text{ MHz}$ , $V_{CB} = 0$ )		$C_{CB}$	—	19	—	pF
Emitter–Base Capacitance ( $f = 1\text{ MHz}$ , $V_{EB} = 0$ )		$C_{EB}$	—	9	—	pF

## COUPLED

Output Collector Current ( $I_F = 10\text{ mA}$ , $V_{CE} = 10\text{ V}$ )	4N25,25A,26 4N27,28	$I_C$ (CTR) <sup>(2)</sup>	2 (20)	7 (70)	—	mA (%)
			1 (10)	5 (50)	—	
Collector–Emitter Saturation Voltage ( $I_C = 2\text{ mA}$ , $I_F = 50\text{ mA}$ )		$V_{CE(sat)}$	—	0.15	0.5	Volts
Turn–On Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>		$t_{on}$	—	2.8	—	$\mu\text{s}$
Turn–Off Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>		$t_{off}$	—	4.5	—	$\mu\text{s}$
Rise Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>		$t_r$	—	1.2	—	$\mu\text{s}$
Fall Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>		$t_f$	—	1.3	—	$\mu\text{s}$
Isolation Voltage ( $f = 60\text{ Hz}$ , $t = 1\text{ sec}$ ) <sup>(4)</sup>		$V_{ISO}$	7500	—	—	Vac(pk)
Isolation Resistance ( $V = 500\text{ V}$ ) <sup>(4)</sup>		$R_{ISO}$	$10^{11}$	—	—	$\Omega$
Isolation Capacitance ( $V = 0\text{ V}$ , $f = 1\text{ MHz}$ ) <sup>(4)</sup>		$C_{ISO}$	—	0.2	—	pF

1. Always design to the specified minimum/maximum electrical limits (where applicable).

2. Current Transfer Ratio (CTR) =  $I_C/I_F \times 100\%$ .

3. For test circuit setup and waveforms, refer to Figure 11.

4. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.



## 6-Pin DIP Optoisolators Darlington Output

The 4N29/A, 4N30, 4N31, 4N32<sup>(1)</sup> and 4N33<sup>(1)</sup> devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon photodarlington detector.

This series is designed for use in applications requiring high collector output currents at lower input currents.

- Higher Sensitivity to Low Input Drive Current
- Meets or Exceeds All JEDEC Registered Specifications
- *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.*

### Applications

- Low Power Logic Circuits
- Interfacing and coupling systems of different potentials and impedances
- Telecommunications Equipment
- Portable Electronics
- Solid State Relays

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

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

#### INPUT LED

Reverse Voltage	$V_R$	3	Volts
Forward Current — Continuous	$I_F$	60	mA
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	120 1.41	mW mW/ $^\circ\text{C}$

#### OUTPUT DETECTOR

Collector-Emitter Voltage	$V_{CEO}$	30	Volts
Emitter-Collector Voltage	$V_{ECO}$	5	Volts
Collector-Base Voltage	$V_{CBO}$	30	Volts
Collector Current — Continuous	$I_C$	150	mA
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	150 1.76	mW mW/ $^\circ\text{C}$

#### TOTAL DEVICE

Isolation Surge Voltage <sup>(2)</sup> (Peak ac Voltage, 60 Hz, 1 sec Duration)	$V_{ISO}$	7500	Vac(pk)
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	250 2.94	mW mW/ $^\circ\text{C}$
Ambient Operating Temperature Range <sup>(3)</sup>	$T_A$	-55 to +100	$^\circ\text{C}$
Storage Temperature Range <sup>(3)</sup>	$T_{stg}$	-55 to +150	$^\circ\text{C}$
Soldering Temperature (10 sec, 1/16" from case)	$T_L$	260	$^\circ\text{C}$

1. Difference in 4N32 and 4N33 is JEDEC Registration for VISO only. All Motorola 6-Pin devices exceed JEDEC specification and are 7500 Vac(pk). The same applies for 4N29 and 4N30.
2. Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
3. 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 4

© Motorola, Inc. 1995

**4N29**  
**4N29A**

**4N30\***  
[CTR = 100% Min]

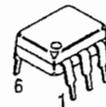
**4N31**  
[CTR = 50% Min]

**4N32\***

**4N33\***  
[CTR = 500% Min]

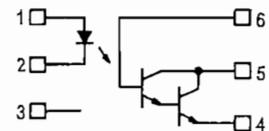
\*Motorola Preferred Devices

STYLE 1 PLASTIC



STANDARD THRU HOLE  
CASE 730A-04

#### SCHEMATIC



- PIN 1. LED ANODE  
2. LED CATHODE  
3. N.C.  
4. EMITTER  
5. COLLECTOR  
6. BASE



**MOTOROLA**

# 4N29 4N29A 4N30 4N31 4N32 4N33

## ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ unless otherwise noted)<sup>(1)</sup>

Characteristic	Symbol	Min	Typ <sup>(1)</sup>	Max	Unit
----------------	--------	-----	--------------------	-----	------

### INPUT LED

*Reverse Leakage Current ( $V_R = 3\text{ V}$ , $R_L = 1\text{ M ohms}$ )	$I_R$	—	0.05	100	$\mu\text{A}$
*Forward Voltage ( $I_F = 10\text{ mA}$ )	$V_F$	—	1.34	1.5	Volts
Capacitance ( $V_R = 0\text{ V}$ , $f = 1\text{ MHz}$ )	$C$	—	1.8	—	$\text{pF}$

### OUTPUT DETECTOR ( $T_A = 25^\circ\text{C}$ and $I_F = 0$ , unless otherwise noted)

*Collector–Emitter Dark Current ( $V_{CE} = 10\text{ V}$ , Base Open)	$I_{CEO}$	—	—	100	$\text{nA}$
*Collector–Base Breakdown Voltage ( $I_C = 100\ \mu\text{A}$ , $I_E = 0$ )	$V_{(BR)CBO}$	30	—	—	Volts
*Collector–Emitter Breakdown Voltage ( $I_C = 100\ \mu\text{A}$ , $I_B = 0$ )	$V_{(BR)CEO}$	30	—	—	Volts
*Emitter–Collector Breakdown Voltage ( $I_E = 100\ \mu\text{A}$ , $I_B = 0$ )	$V_{(BR)ECO}$	5	—	—	Volts
DC Current Gain ( $V_{CE} = 5\text{ V}$ , $I_C = 500\ \mu\text{A}$ )	$h_{FE}$	—	16K	—	—

### COUPLED ( $T_A = 25^\circ\text{C}$ unless otherwise noted)

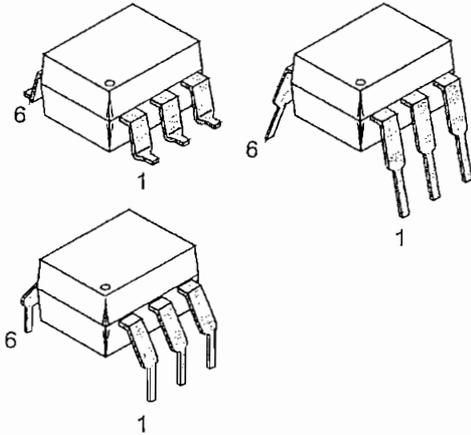
*Collector Output Current <sup>(3)</sup> ( $V_{CE} = 10\text{ V}$ , $I_F = 10\text{ mA}$ )	4N32, 4N33 4N29, 4N30 4N31	$I_C$ (CTR) <sup>(2)</sup>	50 (500) 10 (100) 5 (50)	— — —	— — —	$\text{mA} (\%)$
Isolation Surge Voltage <sup>(4,5)</sup> (60 Hz ac Peak, 1 Second)	4N29/A, 4N30, 31, 32, 33 *4N29, 4N32 *4N30, 4N31, 4N33	$V_{ISO}$	7500 2500 1500	— — —	— — —	$\text{Vac}(\text{pk})$
Isolation Resistance <sup>(4)</sup> ( $V = 500\text{ V}$ )		$R_{ISO}$	—	$10^{11}$	—	Ohms
*Collector–Emitter Saturation Voltage <sup>(3)</sup> ( $I_C = 2\text{ mA}$ , $I_F = 8\text{ mA}$ )	4N31 4N29, 4N30, 4N32, 4N33	$V_{CE(\text{sat})}$	— —	— —	1.2 1	Volts
Isolation Capacitance <sup>(4)</sup> ( $V = 0\text{ V}$ , $f = 1\text{ MHz}$ )		$C_{ISO}$	—	0.2	—	$\text{pF}$
Turn–On Time <sup>(6)</sup> ( $I_C = 50\text{ mA}$ , $I_F = 200\text{ mA}$ , $V_{CC} = 10\text{ V}$ )		$t_{on}$	—	0.6	5	$\mu\text{s}$
Turn–Off Time <sup>(6)</sup> ( $I_C = 50\text{ mA}$ , $I_F = 200\text{ mA}$ , $V_{CC} = 10\text{ V}$ )	4N29, 30, 31 4N32, 33	$t_{off}$	— —	17 45	40 100	$\mu\text{s}$

\* Indicates JEDEC Registered Data. All Motorola 6-pin devices have  $V_{ISO}$  rating of 7500  $\text{Vac}(\text{pk})$ .

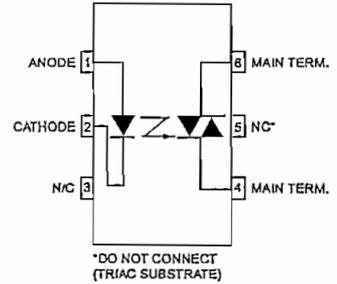
1. Always design to the specified minimum/maximum electrical limits (where applicable).
2. Current Transfer Ratio (CTR) =  $I_C/I_F \times 100\%$ .
3. Pulse Test: Pulse Width = 300  $\mu\text{s}$ , Duty Cycle  $\leq 2\%$ .
4. For this test, Pins 1 and 2 are common and Pins 4, 5 and 6 are common.
5. Isolation Surge Voltage,  $V_{ISO}$ , is an internal device dielectric breakdown rating.
6. For test circuit setup and waveforms, refer to Figure 11.

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M

**PACKAGE**



**SCHEMATIC**



**DESCRIPTION**

The MOC301XM and MOC302XM series are optically isolated triac driver devices. These devices contain a GaAs infrared emitting diode and a light activated silicon bilateral switch, which functions like a triac. They are designed for interfacing between electronic controls and power triacs to control resistive and inductive loads for 115 VAC operations.

**FEATURES**

- Excellent  $I_{FT}$  stability—IR emitting diode has low degradation
- High isolation voltage—minimum 5300 VAC RMS
- Underwriters Laboratory (UL) recognized—File #E90700
- Peak blocking voltage
  - 250V-MOC301XM
  - 400V-MOC302XM
- VDE recognized (File #94766)
  - Ordering option V (e.g. MOC3023VM)

**APPLICATIONS**

- Industrial controls
- Traffic lights
- Vending machines
- Solid state relay
- Lamp ballasts
- Solenoid/valve controls
- Static AC power switch
- Incandescent lamp dimmers
- Motor control

**MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M**

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

Parameters	Symbol	Device	Value	Units
<b>TOTAL DEVICE</b>				
Storage Temperature	$T_{STG}$	All	-40 to +150	$^\circ\text{C}$
Operating Temperature	$T_{OPR}$	All	-40 to +85	$^\circ\text{C}$
Lead Solder Temperature	$T_{SOL}$	All	260 for 10 sec	$^\circ\text{C}$
Junction Temperature Range	$T_J$	All	-40 to +100	$^\circ\text{C}$
Isolation Surge Voltage <sup>(1)</sup> (peak AC voltage, 60Hz, 1 sec duration)	$V_{ISO}$	All	7500	Vac(pk)
Total Device Power Dissipation @ 25°C Derate above 25°C	$P_D$	All	330	mW
			4.4	mW/ $^\circ\text{C}$
<b>EMITTER</b>				
Continuous Forward Current	$I_F$	All	60	mA
Reverse Voltage	$V_R$	All	3	V
Total Power Dissipation 25°C Ambient Derate above 25°C	$P_D$	All	100	mW
			1.33	mW/ $^\circ\text{C}$
<b>DETECTOR</b>				
Off-State Output Terminal Voltage	$V_{DRM}$	MOC3010M/1M/2M MOC3020M/1M/2M/3M	250 400	V
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	$I_{TSM}$	All	1	A
Total Power Dissipation @ 25°C Ambient Derate above 25°C	$P_D$	All	300	mW
			4	mW/ $^\circ\text{C}$

**Note**

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.

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M

**ELECTRICAL CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  Unless otherwise specified)

**INDIVIDUAL COMPONENT CHARACTERISTICS**

Parameters	Test Conditions	Symbol	Device	Min	Typ	Max	Units
<b>EMITTER</b>							
Input Forward Voltage	$I_F = 10\text{ mA}$	$V_F$	All		1.15	1.5	V
Reverse Leakage Current	$V_R = 3\text{ V}, T_A = 25^\circ\text{C}$	$I_R$	All		0.01	100	$\mu\text{A}$
<b>DETECTOR</b>							
Peak Blocking Current, Either Direction	Rated $V_{DRM}, I_F = 0$ (note 1)	$I_{DRM}$	All		10	100	nA
Peak On-State Voltage, Either Direction	$I_{TM} = 100\text{ mA peak}, I_F = 0$	$V_{TM}$	All		1.8	3	V

**TRANSFER CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  Unless otherwise specified.)

DC Characteristics	Test Conditions	Symbol	Device	Min	Typ	Max	Units
LED Trigger Current	Voltage = 3V (note 3)	$I_{FT}$	MOC3020M			30	mA
			MOC3010M			15	
			MOC3021M				
			MOC3011M			10	
			MOC3022M				
			MOC3012M			5	
MOC3023M							
Holding Current, Either Direction		$I_H$	All		100		$\mu\text{A}$

**Note**

1. Test voltage must be applied within  $dv/dt$  rating.
2. This is static  $dv/dt$ . See Figure 5 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  $I_{FT}$  (30 mA for MOC3020M, 15 mA for MOC3010M and MOC3021M, 10 mA for MOC3011M and MOC3022M, 5 mA for MOC3012M and MOC3023M) and absolute max  $I_F$  (60 mA).

DISPLAY UNIT USER'S MANUAL

---

---

# Dot-Matrix LCD Units

(with built-in controllers)

---

---

**SHARP**

## FEATURES

- Interface with either 4-bit or 8-bit microprocessor.
- Display data RAM
- 80 × 8 bits (80 characters).
- Character generator ROM
- 160 different 5 × 7 dot-matrix character patterns.
- Character generator RAM
- 8 different user programmed 5 × 7 dot-matrix patterns.
- Display data RAM and character generator RAM may be accessed by the microprocessor.
- Numerous instructions
- Clear Display, Cursor Home, Display ON/OFF, Cursor ON/OFF, Blink Character, Cursor Shift, Display Shift.
- Built-in reset circuit is triggered at power ON.
- Built-in oscillator.

## OVERVIEW

The LCD unit receives character codes (8 bits per character) from a microprocessor or microcomputer, latches the codes to its display data RAM (80-byte DD RAM for storing 80 characters), transforms each character code into a 5 × 7 dot-matrix character pattern, and displays the characters on its LCD screen.

The LCD unit incorporates a character generator ROM which produces 160 different 5 × 7 dot-matrix character patterns. The unit also provides a character generator RAM (64 bytes) through which the user may define up to eight additional 5 × 7 dot-matrix character patterns, as required by the application.

To display a character, positional data is sent via the data bus from the microprocessor to the LCD unit, where it is written into the instruction register. A character code is then sent and written into the data register. The LCD unit displays the corresponding character pattern in the specified position. The LCD unit can either increment or decrement the display position automatically after each character entry, so that only successive characters codes need to be entered to display a continuous character string. The display/cursor shift instruction allows the entry of characters in either the left-to-right or right-to-left direction. Since the display data RAM (DD RAM) and the character generator RAM (CG RAM) may be accessed by the microprocessor, unused portions of each RAM may be used as general purpose data areas. The LCD unit may be operated with either dual 4-bit or single 8-bit data transers, to accommodate interfaces with both 4-bit and 8-bit microprocessors. The low power feature of the LCD unit will be further appreciated when combined with a CMOS microprocessor.

## HARDWARE

### Interface Signals

Table 2. Interface Signals

SIGNAL NAME	INPUT/OUTPUT	EXTERNAL CONNECTION	FUNCTION
RS	Input	MPU	Register select signal "0": Instruction register (when writing) Busy flag and address counter (when reading) "1": Data register (when writing and reading)
RW	Input	MPU	Read/write select signal: "0": Writing; "1": Reading
E	Input	MPU	Operation (data read/write) enable signal
DB <sub>4</sub> - DB <sub>7</sub>	Input/Output	MPU	High-order lines of data bus with three-state, bidirectional function for use in data transactions with the MPU. DB <sub>7</sub> may also be used to check the busy flag.
DB <sub>0</sub> - DB <sub>3</sub>	Input/Output	MPU	Low-order lines of data bus with three-state, bidirectional function for use in data transactions with the MPU. These lines are not used when interfacing with a 4-bit microprocessor.
V <sub>DD</sub> , V <sub>SS</sub>		Power Supply	V <sub>DD</sub> : +5 V, V <sub>SS</sub> : GND
V <sub>0</sub>		Power Supply	Contrast adjustment voltage

### Functional Blocks

#### Registers

The LCD unit has two 8-bit registers - an instruction register (IR) and a data register (DR). The instruction register stores instruction codes such as "clear display" or "shift cursor", and also stores address information for the display data RAM and character generator RAM. The IR can be accessed by the microprocessor only for writing.

The data register is used for temporarily storing data during data transactions with the microprocessor. When writing data to the LCD unit, the data is initially stored in the data register, and is then automatically written into either the display data RAM or character generator RAM, as determined by the

current operation. The data register is also used as a temporary storage area when reading data from the display data RAM or character generator RAM. When address information is written into the instruction register, the corresponding data from the display data RAM or character generator RAM is moved to the data register. Data transfer is completed when the microprocessor reads the contents of the data register by the next instruction. After the transfer is completed, data from the next address position of the appropriate RAM is moved to the data register, in preparation for subsequent reading operations by the microprocessor. One of the two registers is selected by the register select (RS) signal.

Table 3. Register Selection

RS	RW	OPERATION
0	0	Write to instruction register, and execute internal operation (clear display, etc.)
0	1	Read busy flag (DB <sub>7</sub> ) and address counter (DB <sub>0</sub> - DB <sub>6</sub> )
1	0	Write to data register, and execute internal operation (DR → DD RAM or DR → CG RAM)
1	1	Read data register and execute internal operation (DD RAM → DR or CG RAM → DR)

**Busy Flag (BF)**

When the busy flag is set at a logical "1", the LCD unit is executing an internal operation, and no instruction will be accepted. The state of the busy flag is output on data line DB7 in response to the register selection signals RS = 0, R/W = 1 as shown in Table 3. The next instruction may be entered after the busy flag is reset to logical "0".

**Address Counter (AC)**

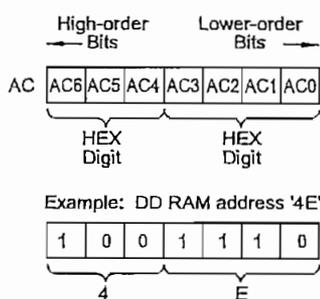
The address counter generates the address for the display data RAM and character generator RAM. When the address set instruction is written into the instruction register, the address information is sent to the address counter. The same instruction also determines which of the two RAM's is to be selected.

After data has been written to or read from the display data RAM or character generator RAM, the address counter is automatically incremented or decremented by one. The contents of the address counter are output on data lines DB0 - DB6 in response to the register selection signals RS = 0, R/W = 1 as shown in Table 3.

**Display Data RAM (DD RAM)**

This 80 x 8 bit RAM stores up to 80 8-bit character codes as display data. The unused area of the RAM may be used by the microprocessor as a general purpose RAM area.

The display data RAM address, set in the address counter, is expressed in hexadecimal (HEX) numbers as follows:



The address of the display data RAM corresponds to the display position on the LCD panel as follows:

a. Address type a . . . .For dual-line display

Digit	Display Position															
	1	2	3	4	5	6	7	8	9	. . .				39	40	
Line 1	00 <sub>H</sub>	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	08 <sub>H</sub>	. . .				26 <sub>H</sub>	27 <sub>H</sub>	
Line 2	40 <sub>H</sub>	41 <sub>H</sub>	42 <sub>H</sub>	43 <sub>H</sub>	44 <sub>H</sub>	45 <sub>H</sub>	46 <sub>H</sub>	47 <sub>H</sub>	48 <sub>H</sub>	. . .				66 <sub>H</sub>	67 <sub>H</sub>	

DD RAM Address (HEX)

When a display shift takes place, the addresses shift as follows:

Left Shift	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	08 <sub>H</sub>	09 <sub>H</sub>	. . .				27 <sub>H</sub>	00 <sub>H</sub>
	41 <sub>H</sub>	42 <sub>H</sub>	43 <sub>H</sub>	44 <sub>H</sub>	45 <sub>H</sub>	46 <sub>H</sub>	47 <sub>H</sub>	48 <sub>H</sub>	49 <sub>H</sub>	. . .				67 <sub>H</sub>	40 <sub>H</sub>

Right Shift	27 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	. . .				25 <sub>H</sub>	26 <sub>H</sub>
	67 <sub>H</sub>	40 <sub>H</sub>	41 <sub>H</sub>	42 <sub>H</sub>	43 <sub>H</sub>	44 <sub>H</sub>	45 <sub>H</sub>	46 <sub>H</sub>	47 <sub>H</sub>	. . .				65 <sub>H</sub>	66 <sub>H</sub>

The addresses for the second line are not continuous to the addresses for the first line. A 40-character RAM area is assigned to each of the two line as follows:

line 1: 00<sub>H</sub> - 27<sub>H</sub>

line 2: 40<sub>H</sub> - 67<sub>H</sub>

For an LCD unit with a display capacity of less than 40 characters per line, characters equal in number to the display capacity, as counted from display position 1, are displayed.

b. Address type b . . . .For single-line display with logically dual-line addressing

Digit	Display Position															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Line 1	00 <sub>H</sub>	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	08 <sub>H</sub>	09 <sub>H</sub>	0A <sub>H</sub>	0B <sub>H</sub>	0C <sub>H</sub>	0D <sub>H</sub>	0E <sub>H</sub>	0F <sub>H</sub>

DD RAM Address (HEX)

When a display shift takes place, the addresses shift as follows:

Left Shift	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	08 <sub>H</sub>	09 <sub>H</sub>	0A <sub>H</sub>	0B <sub>H</sub>	0C <sub>H</sub>	0D <sub>H</sub>	0E <sub>H</sub>	0F <sub>H</sub>	10 <sub>H</sub>
	41 <sub>H</sub>	42 <sub>H</sub>	43 <sub>H</sub>	44 <sub>H</sub>	45 <sub>H</sub>	46 <sub>H</sub>	47 <sub>H</sub>	48 <sub>H</sub>	49 <sub>H</sub>	4A <sub>H</sub>	4B <sub>H</sub>	4C <sub>H</sub>	4D <sub>H</sub>	4E <sub>H</sub>	4F <sub>H</sub>	50 <sub>H</sub>

Right Shift	27 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	02 <sub>H</sub>	03 <sub>H</sub>	04 <sub>H</sub>	05 <sub>H</sub>	06 <sub>H</sub>	07 <sub>H</sub>	08 <sub>H</sub>	09 <sub>H</sub>	0A <sub>H</sub>	0B <sub>H</sub>	0C <sub>H</sub>	0D <sub>H</sub>	0E <sub>H</sub>
	67 <sub>H</sub>	40 <sub>H</sub>	41 <sub>H</sub>	42 <sub>H</sub>	43 <sub>H</sub>	44 <sub>H</sub>	45 <sub>H</sub>	46 <sub>H</sub>	47 <sub>H</sub>	48 <sub>H</sub>	49 <sub>H</sub>	4A <sub>H</sub>	4B <sub>H</sub>	4C <sub>H</sub>	4D <sub>H</sub>	4E <sub>H</sub>

The right-hand eight characters, for the purposes of addressing and shifting, may be considered to constitute a second display line. For the address type of each model, see Table 12.

### Character Generator ROM (CG ROM)

This ROM generates a 5 x 7 dot-matrix character pattern for each of 160 different 8-bit character codes. The correspondence between character codes and character patterns is shown in Tables 4 and 5. Inquiries are invited for units with custom character patterns.

### Character Generator RAM (CG RAM)

This RAM stores eight arbitrary 5 x 7 dot-matrix character patterns, as programmed by the user. For displaying a character pattern stored in the CG RAM, a character code corresponding to the left-most column in Tables 4 and 5 is written into the display data RAM.

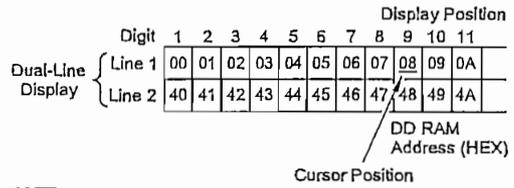
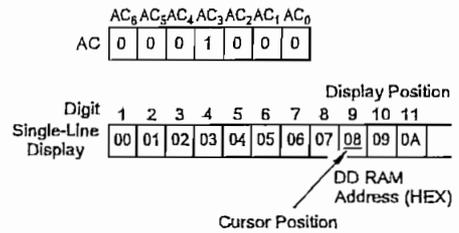
For the relationship among the CG RAM address, the display data, and the displayed pattern, see Table 6. As shown in Table 6., the unused portion of the CG RAM may be used as a general purpose RAM area.

### Timing Generator

The timing generator produces timing signals used for the internal operation of the display data RAM, character generator ROM, and character generator RAM. Timing is controlled so that read-out of the RAM for display and access to the RAM by the external microprocessor do not interfere. Display flicker when data is written to the display data RAM is eliminated.

### Cursor/Blink Controller

This circuit can be used to generate a cursor or blink a character in the display position indicated by the DD RAM address, which is set in the address counter (AC). The following example shows the cursor position when the address counter contains "08" (HEX).



**NOTE:**  
The address counter has the dual function of containing either a DD RAM address or a CG RAM address. The cursor/blink controller does not distinguish between these two functions, and thus, when activated, it always considers the address counter to contain a DD RAM address. To avoid spurious cursor/blink effects, the cursor/blink function should be turned off while the microprocessor writes to or reads from the CG RAM.

### Parallel-to-Serial Converter

This circuit converts parallel data read from the CG ROM or CG RAM to serial data for use by the display driver.

### Bias Voltage Generator

This circuit provides the bias voltage level required for driving the liquid crystal display. Some models incorporate a temperature compensation circuit which generates a temperature dependent bias voltage in order to provide constant display contrast at all ambient temperature levels.

### LCD Driver

This circuit receives display data, timing signals, and bias voltage, and produces the common and segment display signals.

### LCD Panel

This is a dot-matrix liquid crystal display panel arranged in either 1 row of 16 characters, 2 rows of 16 characters, 2 rows of 20 characters, or 2 rows of 40 characters.

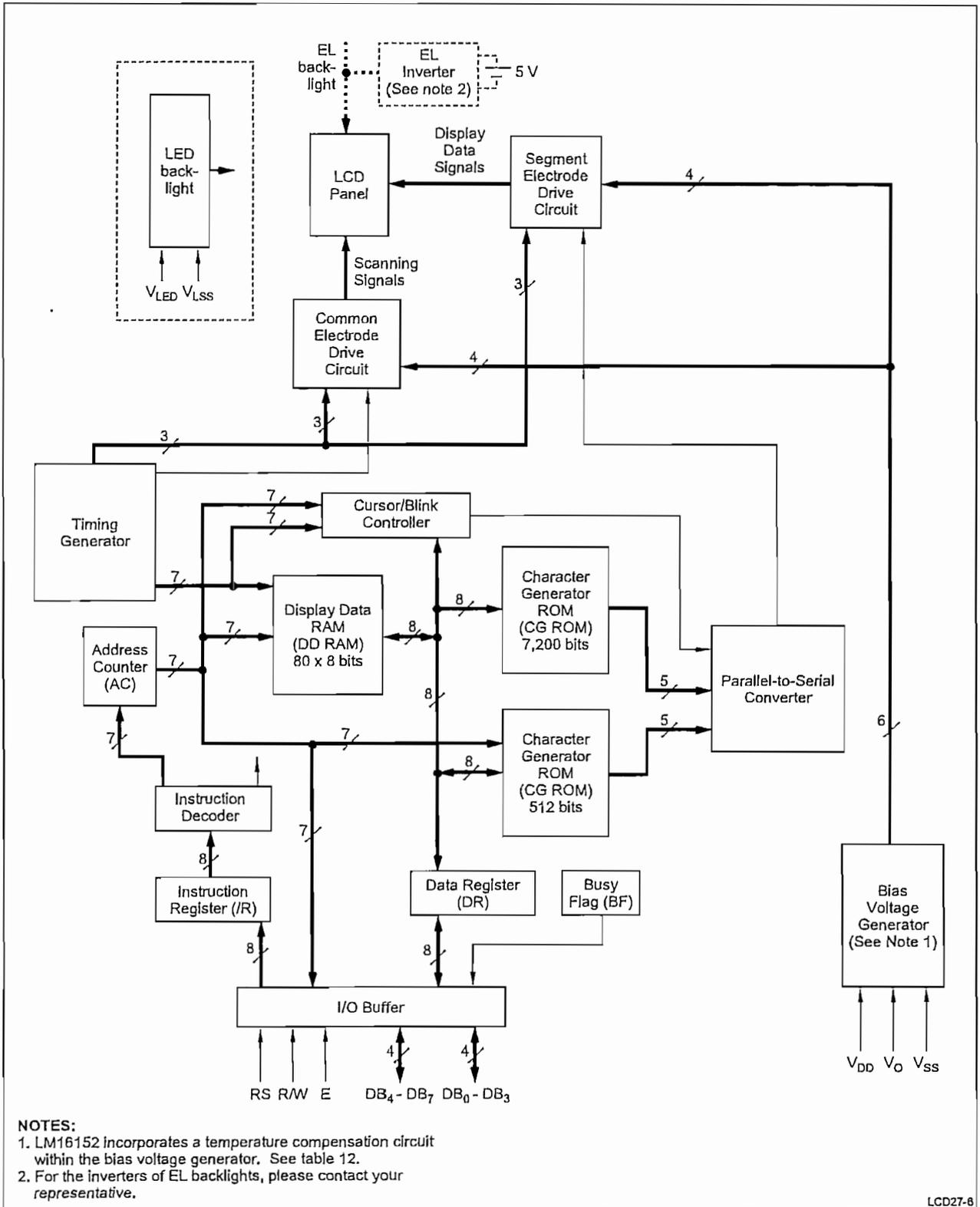


Figure 1. Functional Block Diagram

Table 4. Character Codes

HIGH-ORDER 4 BIT LOW-ORDER 4 BIT		0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	1	2	3	4	5	6	7	8	9	A	B
	(2)	!	1	A	Q	a	4	u	7	7	6	6	6	6
xxxx0010	(3)	"	2	B	R	b	r	r	4	u	x	x	P	Q
	(4)	#	3	C	S	c	s	s	u	u	7	7	e	e
xxxx0100	(5)	\$	4	D	T	d	t	t	u	u	7	7	u	u
	(6)	%	5	E	e	e	e	e	u	u	u	u	u	u
xxx0101	(7)	&	6	F	F	f	f	f	u	u	u	u	u	u
	(8)	'	7	G	W	g	w	w	7	7	7	7	g	g
xxxx1000	(1)	(	8	H	X	h	x	x	u	u	u	u	u	u
	(2)	)	9	I	Y	i	y	y	u	u	u	u	u	u
xxxx1010	(3)	*	#	J	Z	j	z	z	u	u	u	u	u	u
	(4)	+	#	K	C	k	c	c	u	u	u	u	u	u
xxxx1100	(5)	,	<	L	*	l	l	l	u	u	u	u	u	u
	(6)	---	=	M	I	m	i	i	u	u	u	u	u	u
xxxx1110	(7)	.	>	N	^	n	+	+	u	u	u	u	u	u
	(8)	/	?	O	_	o	+	+	u	u	u	u	u	u

NOTES:

1. The CG RAM generates character patterns in accordance with the user's program.
2. Shaded areas indicate 5 x 10 dot character patterns.

Table 5. Character Codes

High-Order Low-Order 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	@	P	\	p		一	夕	ミ	α	ρ
xxxx0001	(2)	!	l	A	Q	a	q	。	ア	チ	△	ã	q
xxxx0010	(3)	"	2	B	R	b	r	Γ	イ	ツ	ノ	β	θ
xxxx0011	(4)	#	3	C	S	c	s	J	ウ	テ	モ	ε	∞
xxxx0100	(5)	S	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ū
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	、	7	G	W	g	w	フ	キ	ヌ	ラ	π	
xxxx1000	(1)	(	8	H	X	h	x	イ	ク	ネ	リ	√	$\bar{x}$
xxxx1001	(2)	)	9	I	Y	i	y	。	ケ	ノ	ル	-1	y
xxxx1010	(3)	*	:	J	Z	j	z	×	コ	ハ	レ	j	
xxxx1011	(4)	+	;	K	[	k	{	*	サ	ヒ	ロ	x	
xxxx1100	(5)	,	<	L	¥	l		〒	シ	フ	ワ	φ	
xxxx1101	(6)	-	=	M	]	m	}	≠	ス	へ	ン	£	+
xxxx1110	(7)	.	>	N	^	n	→	■	セ	ホ	↑	$\bar{n}$	
xxxx1111	(8)	/	?	O	_	o	←	フ	ソ	マ	。	ö	■

**Table 6. Relationship Among Character Code (DD RAM), CG RAM Address, and Character Pattern (CG RAM)**

Character Code (DD RAM Data)								CG RAM Address								Character Pattern (CG RAM Data)														
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0									
High-order bit				Low-order bit				High-order bit				Low-order bit				High-order bit				Low-order bit										
0 0 0 0 * 0 0 0								0 0 0				0	0	0		* * *	1	1	1	1	0	<p>Sample Character Pattern (1)</p>								
												0	0	1		1	0	0	0	1										
												0	1	0		1	0	0	0	1										
												0	0	0	0	1	1	1	1	1	1									0
												1	0	0		1	0	1	0	0										
												1	0	1		1	0	0	1	0										
												1	1	0		1	0	0	0	1										
												1	1	1		* * *	0	0	0	0	0									
0 0 0 0 * 0 0 1								0 0 1				0	0	0		* * *	1	0	0	0	1	<p>Sample Character Pattern (2)</p>								
												0	0	1		0	1	0	1	0										
												0	1	0		1	1	1	1	1										
												0	0	0	0	1	1	0	0	1	0									0
												1	0	0		1	1	1	1	1										
												1	0	1		0	0	1	0	0										
												1	1	0		0	0	1	0	0										
												1	1	1		* * *	0	0	0	0	0									
0 0 0 0 * 1 1 1								1 1 1				0	0	0		* * *														
												0	0	1		1	0	0												
												0	1	0		1	0	1												
												0	0	0	0	1	1	1	1	0										
1	0	0		1	1	1		* * *																						

**NOTES:**

- Character code bits 0 - 2 correspond to CG RAM address bits 3 - 5. Each of the 8 unique bit strings designates one of the 8 character patterns.
- CG RAM address bits 0 - 2 designate the row position of each character pattern. The 8th row is the cursor position. CG RAM data in the 8th row is OR'ed with the display cursor. Any '1' bits in the 8th row will result in a displayed dot regardless of the cursor status (ON/OFF). Accordingly, if the cursor is to be used, CG RAM data for the 8th row should be set to '0'.
- CG RAM data bits 0 - 4 correspond to the column position of each character pattern bit 4 corresponding to the left most column of the character pattern. CG RAM data bits 5 - 7 are not used for displaying character patterns, but may be used as a general purpose RAM area.
- As shown in tables 4 and 5, character patterns in the CG RAM are accessed by character codes with bits 4 - 7 equal to '0'. For example, the character pattern 'R', shown in the first sample character pattern of the table, is selected by the character code '00' (HEX) or '08' (HEX), since bit 3 of the character code is a don't care" bit (i.e., can take either value, '00' or '1').
- CG RAM data '1' produces a dark dot, and data '0' produces a light dot in the corresponding position on the display panel.
- \* = Signifies a "don't care" bit

LCD27-8