

UNIVERSIDAD NACIONAL DE INGENIERIA  
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



**DISEÑO Y CONSTRUCCIÓN DE UN MODULO BASADO EN UN  
MICROCONTROLADOR 8051-INTEL PARA APLICACIONES  
EN SISTEMAS DE CONTROL DIGITAL**

**INFORME DE INGENIERIA**

PARA OPTAR EL TITULO PROFESIONAL DE:

**INGENIERO ELECTRONICO**

PRESENTADO POR:

**JACOB ASTOCONDOR VILLAR**

PROMOCION 1988-I

LIMA-PERU  
2000

**A mis padres Fortunata y Liberato**

**A mi esposa Edelmira**

**A mis hijos Lisbeth y Jacob**

**DISEÑO Y CONSTRUCCION DE UN MODULO  
BASADO EN UN MICROCONTROLADOR 8051-  
INTEL PARA APLICACIONES EN SISTEMAS DE  
CONTROL DIGITAL**

## SUMARIO

En la actualidad es muy frecuente disponer de datos afectados por errores de medición tanto instrumental como los generados por el usuario. Es importante visualizar los errores y tratar de corregirlos ya que dicha información nos indica el valor de las variables de proceso. En la realización de este trabajo se tiene como objetivo minimizar lo anterior y para esto, se diseña un sistema para prácticas de control digital con el microcontrolador 8051 (8031, 8751), el cual se aplica a un proceso de tratamiento térmico. Dicho módulo posee las características siguientes

- control total del microcontrolador por el usuario,
- puertos de entrada/salida accesados por memoria
- Interfase para comunicación serial con la computadora.

Este controlador de E/S configurable por el usuario proporciona un potente interfase analógico y digital con aquellos ordenadores que dispongan de un interfase del tipo RS232, el controlador propuesto será de indudable valor en aquellas aplicaciones donde un simple ordenador lleva a cabo ya sea pequeñas o grandes tareas de control automático, bien de tipo industrial o doméstico.

El sistema se valida realizando un análisis de la influencia de las variables de procesamiento y el control de las mismas con el módulo, en control de tiempo y temperatura en el horno de tratamiento térmico.

## INDICE

<b>PROLOGO</b>	1
<b>CAPITULO I</b>	
<b>EL MÓDULO MICROCONTROLADOR PARA PRACTICAS DE SISTEMAS DE CONTROL DIGITAL</b>	3
1.1 Consideraciones generales	3
1.2 Característica funcional del módulo	4
<b>CAPITULO II</b>	
<b>CARACTERISTICAS DEL SISTEMA CON EL MICROCONTROLADOR 8031 ( 8051 ,8751)</b>	7
2.1. El módulo como elemento de control	7
2.2. Bloques esenciales del módulo	8
<b>CAPITULO III</b>	
<b>DISEÑO DEL MÓDULO CONTROLADOR</b>	11
3.1 Diseño de la unidad del módulo básico de procesamiento central ( CPU)	11
3.1.1 Organización de la memoria	15
3.1.2 El generador reloj	16
3.1.3 Configuración como microprocesador	18
3.1.4 Memoria de datos	19

3.1.5	Memoria de programa	19
3.1.6	Reset	21
3.2	Diseño del módulo de visualización	21
3.3	Diseño de los puertos accesados por memoria	24
3.3.1	Diseño de puerto de entrada analógica	25
3.4	Diseño de la interfase serial	26
3.5	Diseño de la fuente de alimentación	30
<b>CAPITULO IV</b>		
<b>DISEÑO DEL SOFTWARE DEL SISTEMA</b>		<b>33</b>
4.1	Aspectos generales	33
4.2	Programa monitor	33
<b>CAPITULO V</b>		
<b>DESARROLLO Y PRUEBAS DEL MÓDULO</b>		<b>37</b>
5.1	El Hardware	37
5.2	La tarjeta CPU	38
5.2.1	Características de la etapa del CPU	38
5.2.2	Funcionamiento de la CPU	38
<b>CAPITULO VI</b>		
<b>APLICACIÓN DEL SISTEMA PARA EL CONTROL DE UN PROCESO DE TRATAMIENTO TERMICO</b>		<b>42</b>
6.1	Consideraciones generales	42
6.2	Descripción del proceso	43
6.3	Sistema de control de temperatura	45
6.3.1	Medición de temperatura	49

6.3.2	Regulación de temperatura	52
6.3.3	Regulación de potencia	53
6.4	Modelamiento matemático del horno a controlar	54
6.4.1	Descripción	54
6.4.2	Conceptos térmicos : Transferencia de calor	55
6.4.3	Característica regulación del horno	57
6.5	Determinación de parámetros del horno	63
6.5.1	Método teórico	63
6.5.2	Método experimental	64
6.6	Característica dinámica del sensor	67
6.6.1	Función de transferencia del sensor	67
6.6.2	Sensor de temperatura contenido en una termofunda	67
6.7	Diseño del algoritmo del controlador	71
6.7.1	Control programado	71
6.7.2	Selección del tipo de control	73
6.8	Instrumentación del proceso	79
6.8.1	Descripción del hardware	79
6.8.2	Selección de elemento sensor	81
6.8.3	Tratamiento de señal analógica	83
6.8.4	Convertor analógico - digital (A - D)	89
6.8.5	Etapas del controlador o regulador	90
6.8.6	Control de potencia	90
6.8.7	Circuito de cruce por cero	92

6.9	Software del sistema de control automático para hornos de tratamiento térmico	93
6.9.1	Estructura del software de control de proceso térmico	93
6.9.2	Estructura del software para su supervisión del sistema	93
	<b>CONCLUSIONES</b>	97
	<b>ANEXO A</b>	
	<b>PROGRAMA MONITOR DEL MODULO DEL CONTROL DIGITAL</b>	98
	<b>ANEXO B</b>	
	<b>PROGRAMA DE CONTROL DE TEMPERATURA DE UN HORNO PARA TRATAMIENTO TERMICO DE CONTROL PID</b>	103
	<b>ANEXO C</b>	
	<b>PROGRAMACION DE PARAMETROS INTERFACE CON PC</b>	113
	<b>ANEXO D</b>	
	<b>CONJUNTO DE INSTRUCCIONES DEL MICROCONTROLADOR 8051</b>	125
	<b>ANEXO E</b>	
	<b>PLANO DEL MODULO</b>	166
	<b>BIBLIOGRAFIA</b>	167

## PROLOGO

El desarrollo constante de la electrónica digital ha dado lugar a microcircuitos cada vez más densos en su nivel de integración; entre ellos, el microcontrolador. Este es un dispositivo que contiene los subsistemas básicos de un sistema con microprocesador como son:

- la unidad central de procesamiento
- la memoria para almacenamiento de programas
- la memoria para almacenamiento temporal de datos
- los puertos de entrada/salida para el manejo de periféricos, contadores, temporizadores y sistemas de interrupciones [6].

Los microcontroladores se utilizan para resolver problemas en múltiples campos tales como

- electrodomésticos
- industria automotriz
- imagen y sonido
- medicina
- aplicaciones militares
- industria manufacturera
- equipos de regulación
- automatismo y robótica

- **Informática.**

En el estudio de los microcontroladores es necesario el uso de herramientas de apoyo, como son los simuladores y los entrenadores.

El propósito principal de éste trabajo es el desarrollo de un módulo para prácticas de control digital basado en el microcontrolador 8031( 8051 , 8751 -INTEL) y su aplicación en control de hornos de tratamiento térmico, este dispositivo presenta características muy útiles para aplicaciones de control, es fácil de usar, económico y disponible en el mercado. La aplicación se realizó en un proceso de tratamiento térmico en hornos de resistencias, se apreció la influencia del control de variables en las propiedades finales del artículo tratado , así como el control de funcionamiento del equipo digital.

# **CAPITULO I**

## **EL MODULO MICROCONTROLADOR PARA PRACTICAS DE SISTEMAS CONTROL DIGITAL**

### **1.1 Consideraciones generales**

Un sistema de automatización y control bien diseñado funcionará de acuerdo a las capacidades de la tecnología escogida para su implementación. Pero un sistema de control mal diseñado no funcionará bien ni con la mejor tecnología disponible.

Por lo tanto, antes de seleccionar la tecnología de implementación del sistema, es indispensable diseñar correctamente. Esto puede sonar obvio, pero aquí y en todas partes del mundo muchas veces se presta más atención a la tecnología que al diseño inicial

El diseño de un sistema de control apropiado debe conjugar armoniosamente conocimientos detallados sobre el proceso de producción y sobre técnicas de control, de manera que las segundas se apliquen sobre el primero. Un vez que se tenga un diseño adecuado (no antes), se deberá decidir la mejor manera de implementarlo.

Es generalmente aceptado en la industria el concepto de **niveles de control**, que identifica los distintos tipos de funciones que deben realizar por ejemplo:

- Nivel 0 : incluye los sensores básicos - temperatura , presión ,etc. y actuadores básicos , que interactúan directamente con el proceso .

- **Nivel 1** : Consiste en los controles regulatorios y secuenciales que implementan la estrategia de control de proceso

Existen varias tecnologías para realizar el nivel 1. En general lo podemos agrupar en los siguientes grandes grupos:

- Control directo basado en PC
- Control lógico programable ( PLC)
- Microcontroladores uC
- Sistemas de control distribuidos ( DCS)

El tema de qué tecnología utilizar para que aplicaciones es materia de intenso debate hoy en día, este trabajo tiene como finalidad presentar un sistema basado en el microcontrolador 8051 - INTEL

El microcontrolador es percibido como la solución más efectiva en costo para la implementación de uno o unos pocos lazos de control

## **1.2 Características funcionales del módulo**

Un sistema para practicas de control digital basado en un microcontrolador se utiliza para experimentar y analizar, el comportamiento de un programa, con el fin de depurarlo y modificarlo hasta obtener el funcionamiento óptimo del programa.

La ventaja que ofrece el modulo cuando se realizan los programas son:

- Se ejecutan desde una RAM
- se evita la necesidad de grabar los programas en memoria ROM
- Se ahorra la inherente pérdida de tiempo y dinero.

Para operar el sistema entrenador éste consta de periféricos de entrada y salida como son:

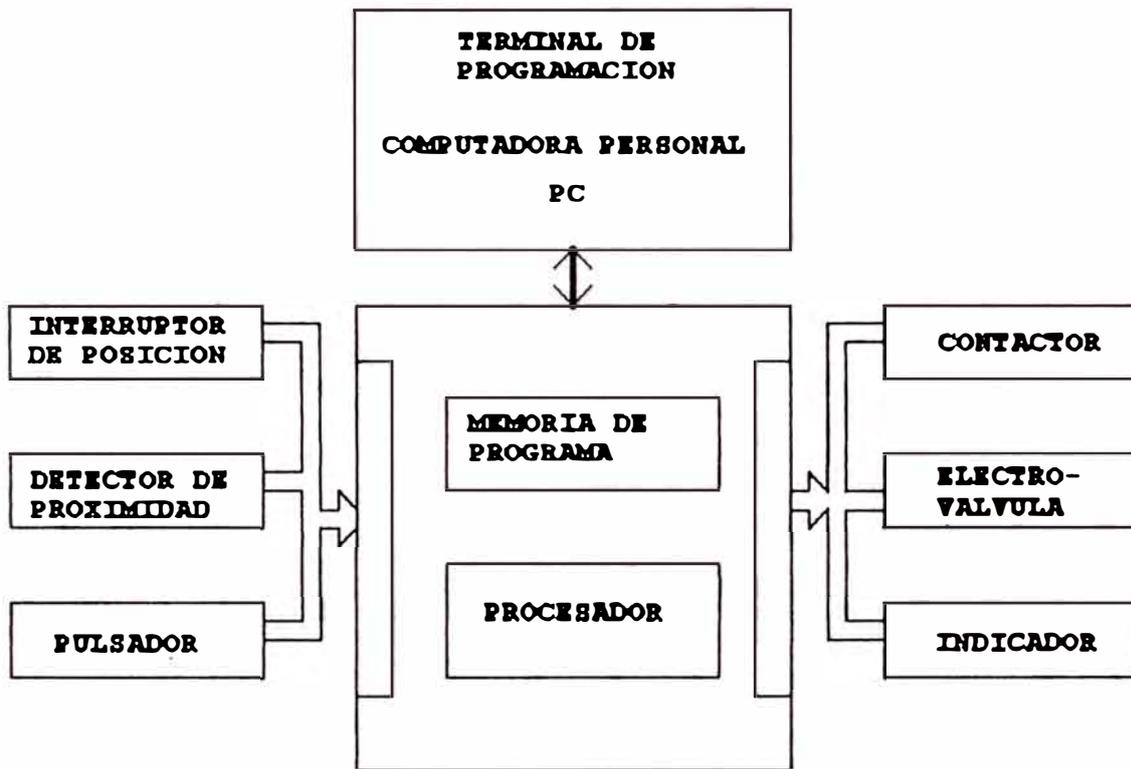
- la pantalla
- los puertos de entrada/salida
- los puertos para control de interrupciones
- una interfase de comunicación serial.

Para introducir los programas se utiliza la interfase de comunicación serial, en este caso si se utiliza la interfase de comunicación serial, se tiene la capacidad de recibir los programas desde la PC y de realizar la transmisión de programas relativamente grandes en unos cuantos segundos.

Los puertos de entrada y salida se utilizan con el fin de establecer contacto a través de otros sistemas con el mundo real

El puerto de control de interrupciones esta disponible para que los periféricos que se conectan al entrenador establezcan comunicación directa con el 8031.

El funcionamiento del entrenador se basa en un programa monitor cuya función es proporcionar al usuario un conjunto de herramientas para ayudarlo a desarrollar, depurar y ejecutar programas. El programa monitor y gestiona los recursos del sistema, elimina la implicación con manipulaciones repetitivas de segmentos de programa ver fig.1.1



**fig1.1 El modulo controlador**

## **CAPITULO II**

### **CARACTERISTICAS DEL SISTEMA CON EL MICROCONTROLADOR 8031 (8051 , 8751)**

#### **2.1 El módulo como elemento de control**

El módulo programable se puede considerar como un equipo electrónico de control, que se adapta al proceso a controlar mediante un programa específico (software) que contiene la secuencia de operaciones a realizar. Esta secuencia de operaciones se define sobre señales de entrada y salida al proceso.

Las señales de entrada pueden proceder de elementos digitales, como finales de carrera y detectores de proximidad, o analógicos, como sensores de temperatura y dispositivos de salida en tensión y corriente continua.

Las señales de salida son órdenes digitales todo o nada o señales analógicas en tensión y corriente , que se envía a los elementos indicadores y actuadores del proceso, como lámparas, contactores , válvulas ,etc.

El módulo gobierna las señales de salida según el programa de control previamente almacenado en una memoria en este caso la RAM a partir del estado de las señales de entrada.

Este programa se introduce en el módulo a través de la unidad de programación, que permite además funciones adicionales como depuración de programas, simulación, monitorización, control del módulo.

## **2.2 Bloques esenciales del módulo**

Desde el inicio de esta investigación se tuvo especial interés, en que las características del entrenador con el microcontrolador 8031 quedarán al servicio del usuario por lo que se tienen características tales como puertos de entrada/salida accesados por memoria.

El entrenador realizado consiste de:

- El microcontrolador 8051
- memoria del sistema (EEPROM)
- memoria para almacenamiento de datos (RAM)
- 4 display de 7 segmentos
- 2 puertos de entrada de datos accesados por memoria;
- 1 puerto para entrada de datos analógicos
- 1 puerto para el control del microcontrolador e interfase para comunicación serial con la computadora.
- 1 puerto para control de interrupciones
- Fuente de alimentación

El microcontrolador tiene un microprocesador interno que consulta el estado de las entradas y recoge de la memoria de programa la secuencia de instrucciones a ejecutar, elaborando a partir de ella las señales de salida u ordenes que se enviarán al proceso. Durante la ejecución del programa, las instrucciones son procesadas en serie, una tras otra.

Es responsable de actualizar continuamente los temporizadores y contadores internos que hayan sido programado.

La memoria del modulo contiene todos los datos e instrucciones que se necesita para ejecutar la tarea de control.

La memoria interna del microcontrolador (RAM interna) es la encargada de almacenar datos intermedios de cálculo que no aparece y variables internas que no parece directamente sobre las salidas.

La memoria de programa contiene la secuencia de operaciones que deben realizar sobre las señales de entrada para obtener las señales de salida, así como los parámetros de configuración del módulo.

Las interfases de entrada y salida establece la comunicación del modulo con la planta o proceso a controlar.

La etapa de visualización nos permite observar la evolución de las variables de la planta o proceso.

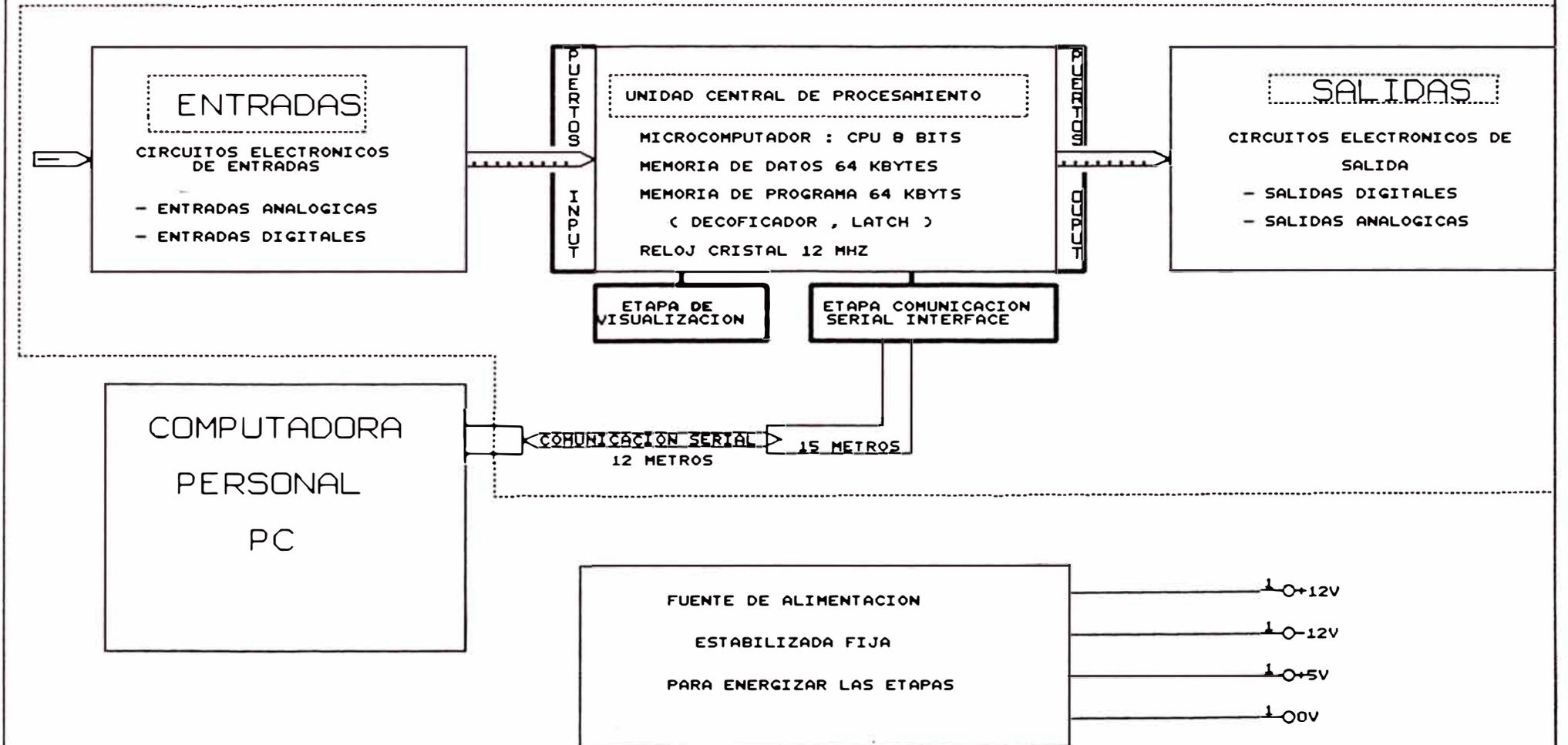
La fuente de alimentación proporciona, a partir de una exterior, las tensiones necesarias para el buen funcionamiento de los distintos circuitos electrónicos del sistema

El diseño del Software consiste en:

- almacenamiento de programas y/o datos en memoria RAM;
- verificación y modificación de localidades
- ejecución de programas desde la RAM
- programación de la RAM desde la PC
- transmisión de datos del entrenador hacia la PC

Ver figura 2.1

# PARTES COMPONENTES DEL MODULO



UNIVERSIDAD NACIONAL DE INGENIERIA DEPARTAMENTO DE ELECTRONICA		
Title		
PARTES DEL MODULO		
Size	Document Number	REV
A	PROYECTO MODULO DIDACTICO .FIG.2.1	
Date:	December 28, 1999	Sheet of

## **CAPITULO III DISEÑO DEL MÓDULO CONTROLADOR**

El diseño del prototipo se realizó por partes considerando para esto los siguientes módulos:

Diseño del módulo básico de procesamiento central ( CPU)

Diseño del módulo de visualización

Diseño de puertos accesados por memoria

Diseño de interfase serial

### **3.1 Diseño de la unidad del módulo básico de procesamiento central (CPU)**

La CPU construido alrededor al rededor de un microcontrolador 8051 (8031, 8751) que es el encargado de ejecutar el programa de usuario y de ordenar las transferencias de información en el sistema de entradas y salidas.

Adicionalmente, puede también establecer comunicación con periféricos externos, como son la unidad de programación (PC), Display, otros microcontroladores o computadores, etc.

El módulo puede funcionar adecuadamente con los microcontroladores 8031, 8051, 8751 Intel mencionamos las características de los siguientes microcontroladores:

a) Características del microcontrolador 8031 que son:

- microcomputador de 8 BITS

- memoria RAM de 128 bytes útiles para el usuario
- Puede direccionar hasta 64 Kbytes de memoria para programa externo
- Espacio de memoria de 64 Kbytes para datos externos
- Tiene dos contadores / temporizadores programables de 16 bits
- Salida serie I/O totalmente programable
- Cuatro puertos paralelos I/O direccionable de 8 BITS cada uno; los puertos Port P0 y Port P1 sólo pueden utilizados para el DATO y la DIRECCION de las memorias externas; el DATO está multiplexado con las DIRECCIONES más bajas. Algunos BITS del puerto Port P3 se necesitan para el control de memorias o periféricos, así como de entrada de interrupciones y para la salida y entrada serie, claro está que si no se usan las interrupciones o las salidas/entradas serie, estos pines pueden ser utilizados como BITS de entrada o salida.

En definitiva, sólo disponemos del puerto Port P1 para los datos de entradas/salidas.

Esto parece una limitación, pero no es así, sólo implica un poco más de hardware.

- Temporizadores e interrupciones I/O con dos niveles de prioridad .
- Funciones booleanas con 128 indicadores de software, muchos indicadores de hardware y 12 instrucciones de operandos de BITS.
- La UNIDAD LOGICA ARITMETICA incluye como funciones aritméticas suma, resta, multiplicación y división. Como funciones lógicas, AND, OR, OR-exclusiva y complemento.

- Un total de 111 instrucciones con 51 funciones básicas (Incluyendo movimientos de memoria).
- Velocidad 12 MHz.

#### b) El Microcontrolador 8051

Este "micro" además de todo lo anterior, contiene integrada una ROM de 4Kx8 programable de fábrica; es decir, el programa con el que ha funcionar el sistema está ya grabado en el momento de la fabricación del circuito integrado.

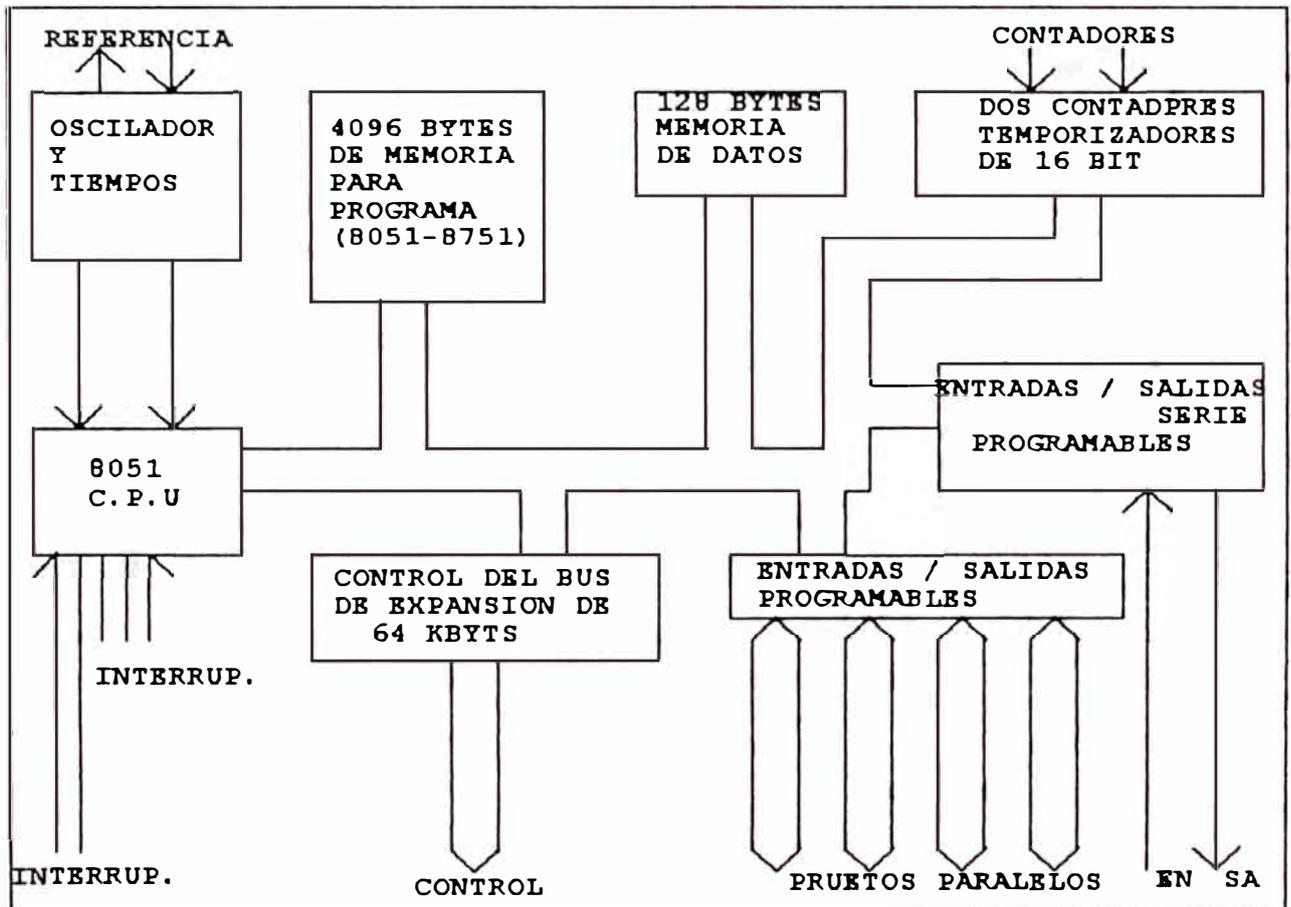
#### c) El Microcontrolador 8751

Este es el tercero de la familia, es exactamente igual que el 8051, sólo que la memoria ROM es sustituida por una memoria EPROM de la misma capacidad; es decir de 4Kx8. Esta memoria es programable por el usuario y borrable mediante luz ultravioleta. En todo caso lo demás no se diferencia en nada de los anteriores.

### **El Interior de la familia 8051**

La estructura por bloques es la de la figura 3.1 donde podemos apreciar:

- La unidad central de procesos o microprocesador (CPU)
- La memoria de programa (según versiones)
- La RAM interna con los registros de funciones especiales ( SFRs)
- Los puertos programables de entrada y salida
- Bloque de control para acceder a memoria externa
- Los temporizadores/contadores
- El sistema de interrupción
- El puerto serie



**Figura .3.1: Estructura Interna del microcontrolador**

## Configuración de los pines

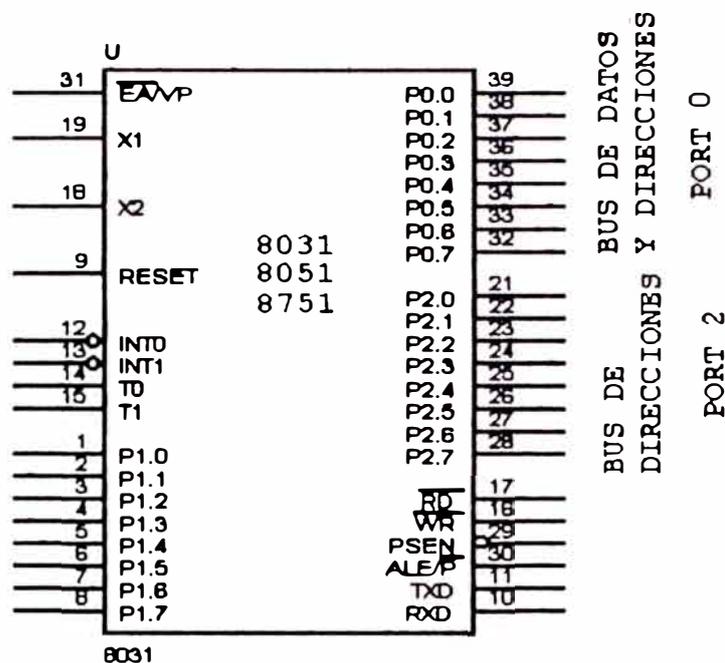


Figura 3.2: Configuración de pines del microcontrolador

### 3.1.1 Organización de la memoria

Los microcontroladores 8051, 8751 tiene separados los espacios de memoria de direcciones para memoria de programas y memoria de datos ver figura 3.3.

- **RAM Interna de accesos tanto directo como indirecto (00-7FH);** se accede a esta área con instrucciones como MOV ó MOV @Ri. Los primeros 32 bits corresponden a los 4 bancos de registros, de 8 registros (R0 a R7) cada uno. Los 16 bits siguientes corresponden a la zona de bits, estos 128 bits pueden ser manipulados directamente, dando una enorme versatilidad a la programación.

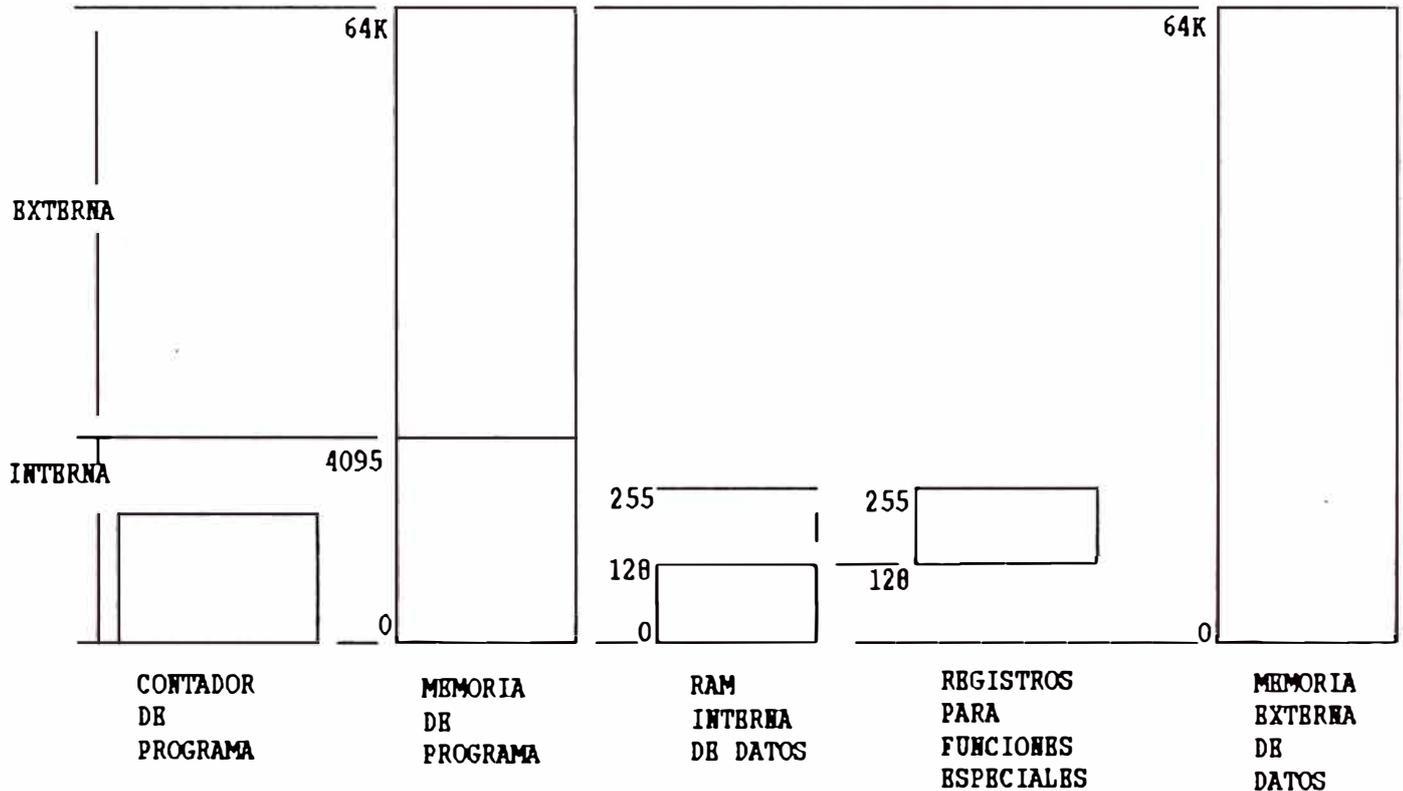
Los bits restantes son para propósito general.

**- RAM Interna de acceso solamente Indirecto (80 - FFH)**

Solo mediante instrucciones MOV @RI, PUSP, o POP, se puede acceder a esta área de memoria, cuyo uso es de propósito general, o para contener la pila.

**- Registros especiales de función , solo de acceso directo ( 80 - FFH)**

Debido a la cantidad de circuitería que incorpora el microcontrolador, necesita un conjunto grande de registros , cada uno de los cuales con una función determinada.



**Figura 3.3: Estructura de memorias**

### 3.1.2 El generador de reloj

Proporciona las señales necesarias para la sincronización de todas las operaciones internas.

La frecuencia del cristal suele ser de 11,0592 MHz, aunque existe versiones que pueden operar a frecuencias más elevadas. Dicha frecuencia

### 3.1.3 Configuración como microprocesador

La memoria de programas sólo puede ser leída y tiene como máximo 64K.

En el 8051 la memoria de programas interna es de 4 K.

La señal de autorización de lectura en la memoria de programas externas es PSEN .

Entonces la memoria de programas puede ser externa (en otro chip) o puede ser interna (en el propio microcontrolador).

La selección se realiza por hardware con la señal EA (External Access) poniendo este pin a VCC o a Vss; se dispone el terminal EA a tierra para que el acceso a memoria sea externo

Cuando utiliza memoria externa de programas el pin EA = VSS tal como se muestra la figura.3.6

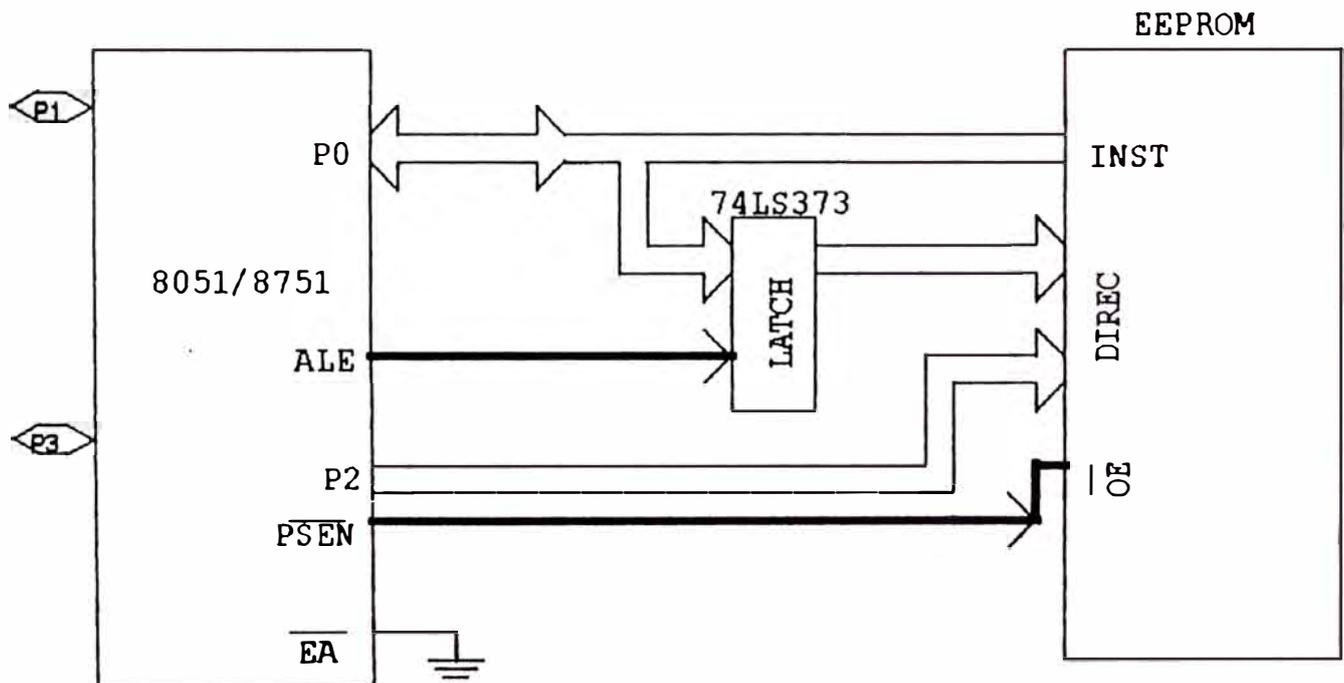


figura.3.6 Conexión de memoria externa de programas

Por medio de los puertos de direcciones (P0 y P2), se multiplexea en el tiempo al puerto P0, bus de dirección y como bus de datos, para ello almacena la dirección de P0 por medio de un Flip - Flop de 8 bits.

Para la compatibilidad de tiempos con las señales de control del microcontrolador se hace necesario interconectar la entrada de reloj FF-D con la señal ALE.

Al mismo tiempo que emite el microcontrolador el PCL por P0, la parte alta del contador de programas (PCH) se emite por P2 .Entonces PSEN autoriza la lectura al microcontrolador del código de instrucción a través del puerto P0.

#### **3.1.4 Memoria de datos**

El 8031 / 80518751 puede direccionar hasta 64 Kbytes de memoria de datos externa, la figura. muestra la configuración para acceder a 2 Kbytes de memoria de datos externa. La memoria de datos coexiste en el mismo espacio de direcciones, en este caso se dice que la memoria es combinada, se obtiene aplicando las señales RD y PSEN a una puerta lógica AND y utilizando la salida de la puerta como **strobe del chip** de la memoria de datos.

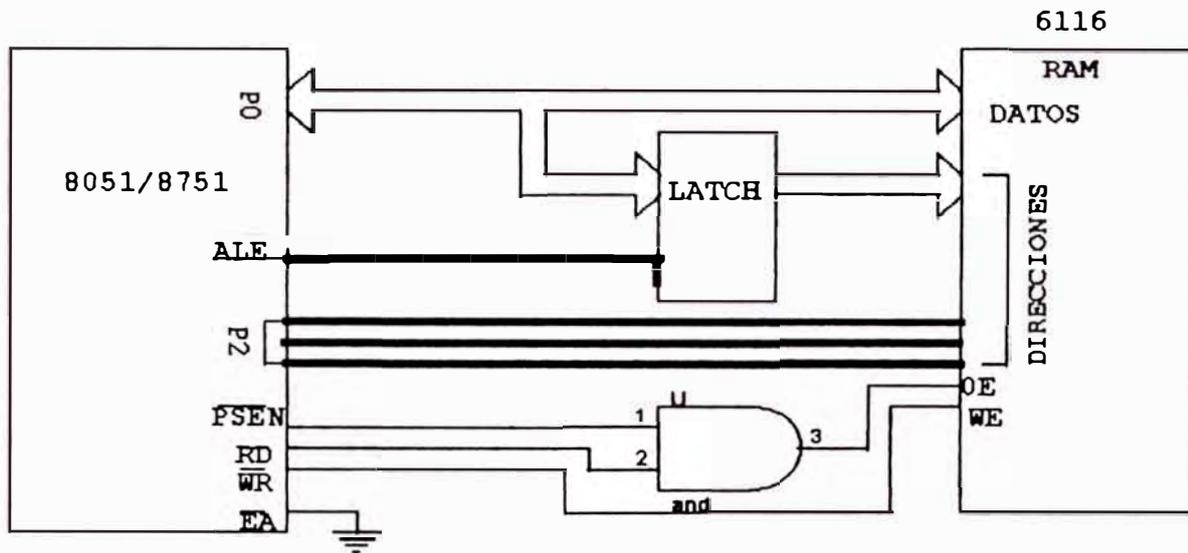
Ver Figura.3.7

#### **3.1.5 Memoria de programa**

Se dispone de una EEPROM (2816) que contiene el programa monitor y una RAM ( 6116) separadas con la técnica "mapa de memoria".

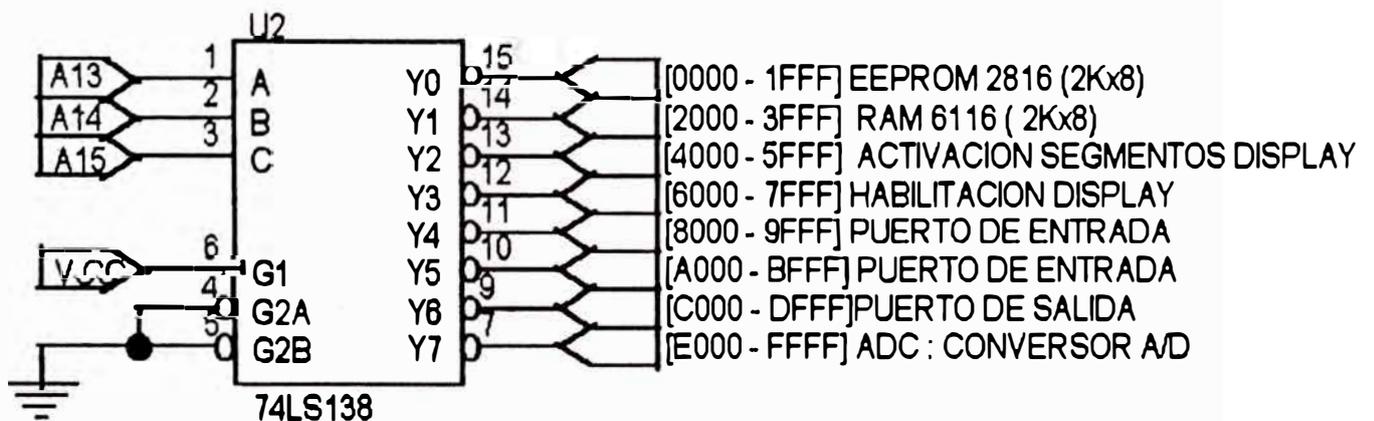
El mapa de memoria es implementado mediante el decodificador de 3 a 8 con el 74LS138 ; dicho mapa de memoria posee 65536 direcciones ( 0000H

a la FFFFH) es decir 64Kbytes. Los 64Kbytes esta dividido en zonas de 8Kytes cada una



**Figura.3.7 Conexión de memoria de datos**

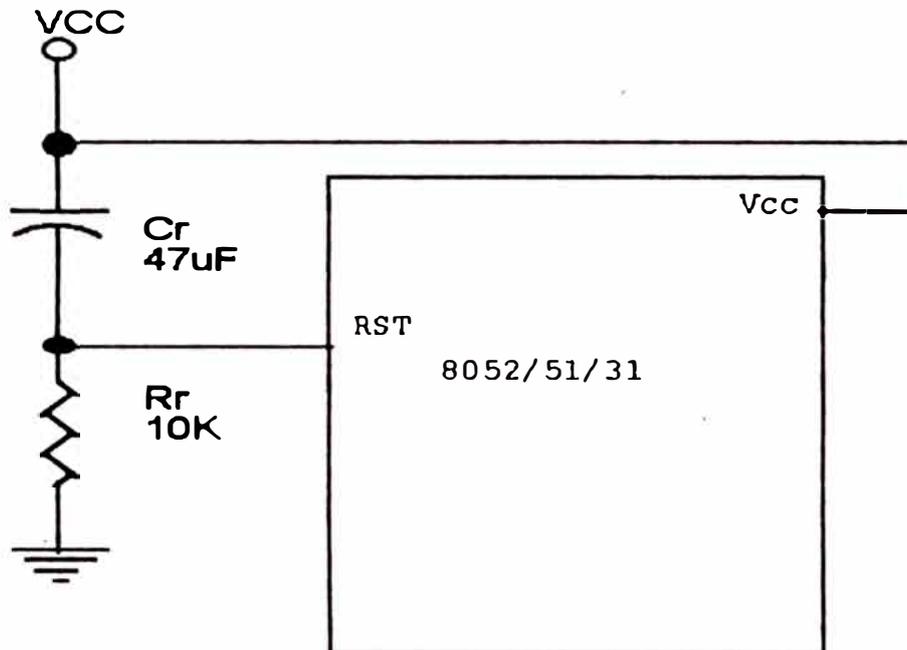
Para ello se conecta las tres líneas de mayor peso del bus de direcciones en este caso: A15,A14,A13 a las entradas del decodificador, en cada una de las salidas del decodificador tenemos los bloques de la Figura 3.6



**Figura.3.8 Mapa de memoria**

### 3.1.6 Reset

Se conecta un módulo de RESET, se recurre a un circuito que permite su activación y desactivación [4].



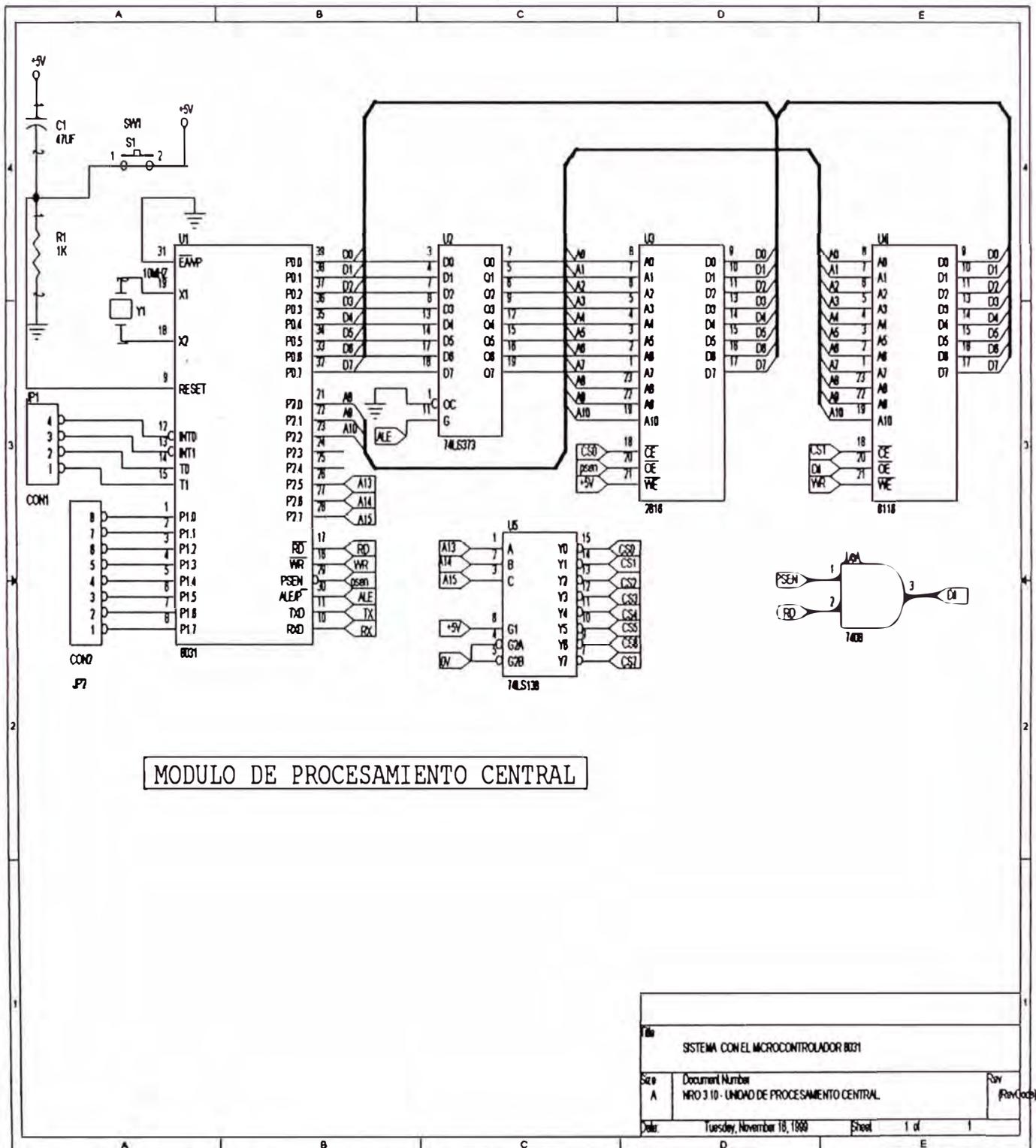
**Figura.3.9: Circuito Reset**

El diagrama de conexiones del CPU básico se muestra en la **figura 3.10**

### 3.2 Diseño del módulo de visualización

El módulo de visualización lo usa el usuario para observar la información desplegada, esta subrutina del ensamblador le permite al microcontrolador tomar el menor tiempo posible de su atención para la transferencia de información.

Este módulo contiene dos 74LS373 de salida que sirve para aislar el bus de datos de la memoria, este módulo está dentro de la zona "mapa de memoria", para separar estas señales se usan las terminales RD y WR, teniéndose además una línea codificadora de dirección la cual proviene del CPU; para realizar la activación de los display se requiere cuatro transistores



**MODULO DE PROCESAMIENTO CENTRAL**

Título		
SISTEMA CON EL MICROCONTROLADOR 8031		
Serie	Document Number	Rev
A	NRO 3 10 - UNIDAD DE PROCESAMIENTO CENTRAL	Rev. Cady
Fecha	Tuesday, November 18, 1990	Sheet 1 of 1

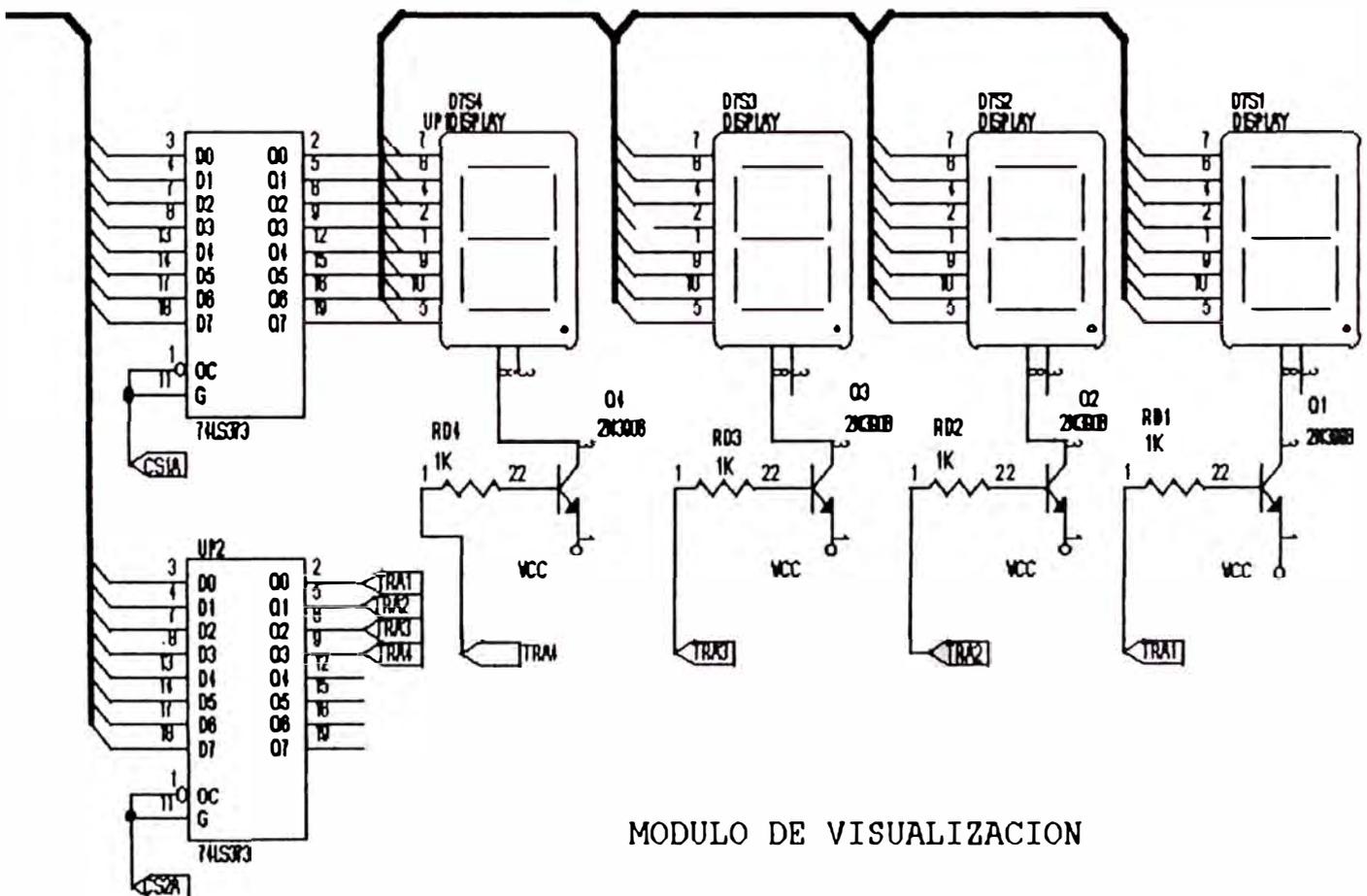
PNP, a través de uno de los laths ( 74LS373 ) , escribiendo los códigos correspondiente se activan los transistores.

El módulo se muestra en la figura 3.11

Para apagar los display se realizará las siguientes instrucciones :

```

MOV A ,#0FFH      ; El acumulador se carga con el código 0FFh
MOV DPTR, #6000H  ; Se apunta la ubicación de los transistores
                  ; de activación de los display
MOVX @DPTR, A     ; se envía el código a la posición externa de
                  ; activación de los display
    
```



MODULO DE VISUALIZACION

Figura 3.11: Módulo de visualización

### 3.3 Diseño de puertos accesados por memoria

Para incrementar la potencialidad del sistema se añadieron puertos de ocho bits de entrada y salida accesados a memoria que consta de:

- dos de entrada y uno de salida
- Un puerto de señal analógica de 0 a 5 Voltios ( procesamiento de señales) utilizándose para cada puerto un 74LS373. Se uso el bus de datos, las señales RD, WR y dos líneas codificadoras de dirección para establecer la interfaz de estos puertos con el 8031. Los puertos se encuentran decodificados en pares, una de entrada y uno de salida, con la misma línea decodificadora de dirección. Los puertos de entrada contienen resistencias " pull-up" en sus entradas.

El programa monitor cuenta con subrutinas para acesar a estos puertos en forma independiente; las rutinas se encargan de direccionar el puerto y realizar la acción indicada en forma transparente al usuario. En la figura 3.12 se muestra el circuito utilizado en estos puertos.

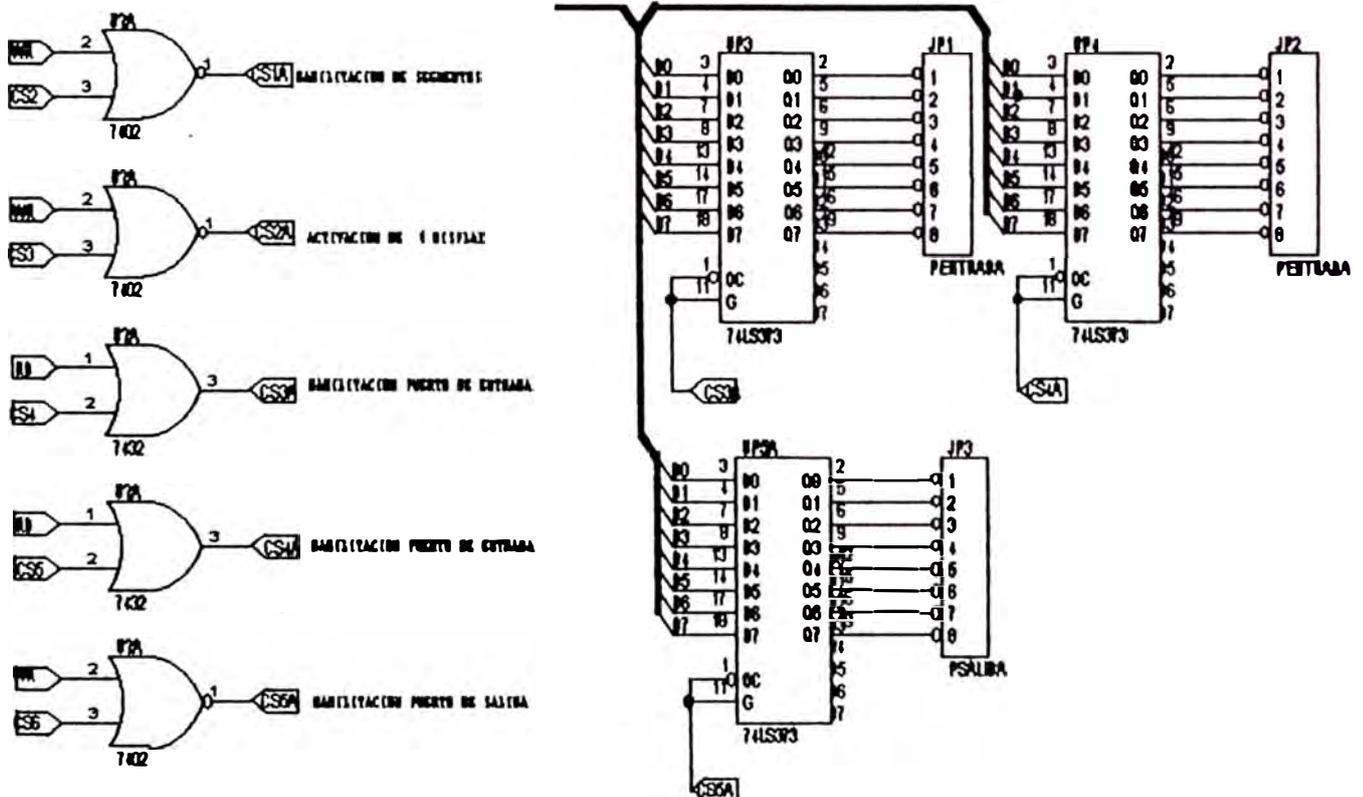


Figura 3.12 Puertos entrada /salida

Algunas instrucciones para realizar lectura-escritura de los puertos podemos indicar

Ubicación del puerto de entrada : 8000h ( lectura )

MOV DPTR , #8000H ; Se apunta la ubicación del puerto

MOVX A , @DPTR ; Se realiza la lectura a través del Acc

Ubicación del puerto de salida : 0A000h ( Lectura )

MOV DPTR, #0A000H ; Se apunta la ubicación del puerto

MOVX @DPTR, A ; Se escribe el dato en el puerto

### 3.3.1 Diseño de puerto de entrada analógica

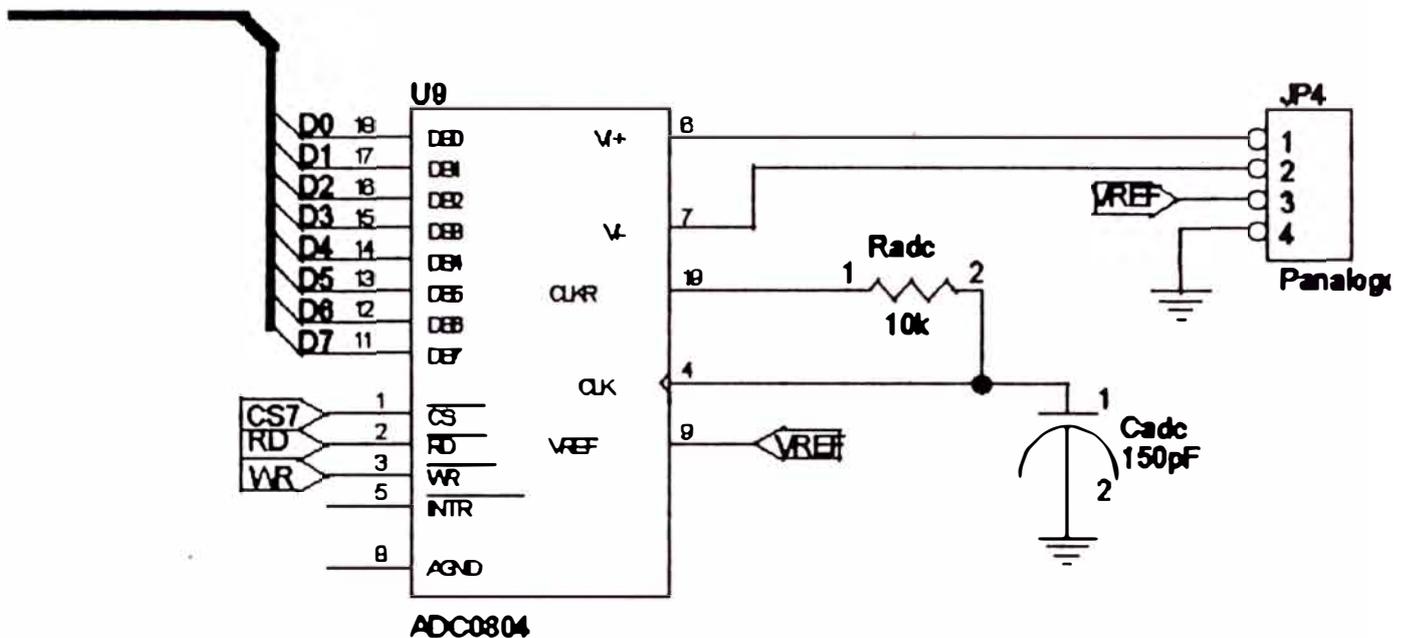
Para procesamiento de señales analógicas, el módulo dispone de una entrada para señales analógicas de 0 a 5Voltios, elemento procesador es el ADC0804/Intel, es un conversor análogo digital compatible con los procesadores de la familia Intel, es de 8 bits, a este puerto también es accesado por memoria, la lectura escritura es controlado por el microcontrolador previamente apuntado a la posición de memoria, para esto dispone de dos pines WR y RD . tal como se muestra en la figura.3.13

Esta constituido por :

- 01 conversor análogo - digital : ADC 0804 de 8 bits
- Red RC para el reloj

Los controles de la etapa esta constituido por

- Control de activación del ADC-0804 : CS7 a través del decoder 74LS138
- Control de escritura WR y lectura RD directamente activado por el Microcontrolador.



ETAPA DE PROCESAMIENTO ( adquisición de

**Figura.3.13 Etapa de procesamiento de señales analógicas**

El trabajo de esta etapa es realizado mediante Software

MOV DPTR, #0F000H ; Ubicación del ADC en el mapa de memoria

MOV A, #0FFH ; Código para orden de conversión

MOVX @DPTR, A ; Orden para el ADC de realizar conversión

CALL DELAY ; Retardo de conversión

MOVX A, @DPTR ; Lectura del dato convertido a digital se carga al

; al acumulador del microcontrolador

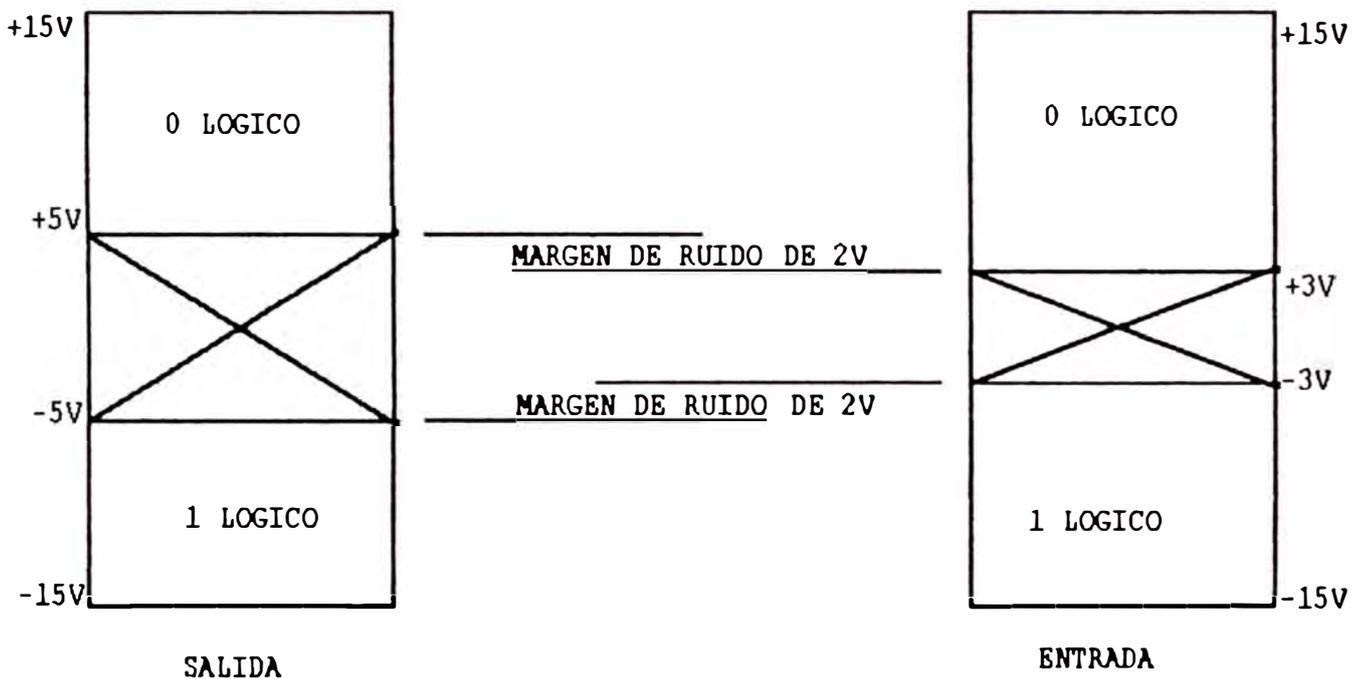
### 3.4 Diseño de la Interfase serial

Debido a que el puerto serial de la computadora utiliza la norma RS-232, define los niveles de voltaje para el 0 y el 1 lógicos, a si como muchas otras cosas necesarias para hacer que el equipo se comuniquen con otro.

El estándar RS - 232 es sistema de lógica negativa o sea los voltajes de salida fluctúan desde 3.0 Volts hasta 15.0 Volts para un "0" lógico y desde -

3.0 Volts hasta -15 Volts para "1" lógico y el margen de ruido es de 2 voltios y el intervalo de variación de voltajes para un 1 o 0 válidos es de 10 Volts.

En la figura 3.14 se muestra los niveles de voltajes empleados para definir el 1 y 0 lógicos.

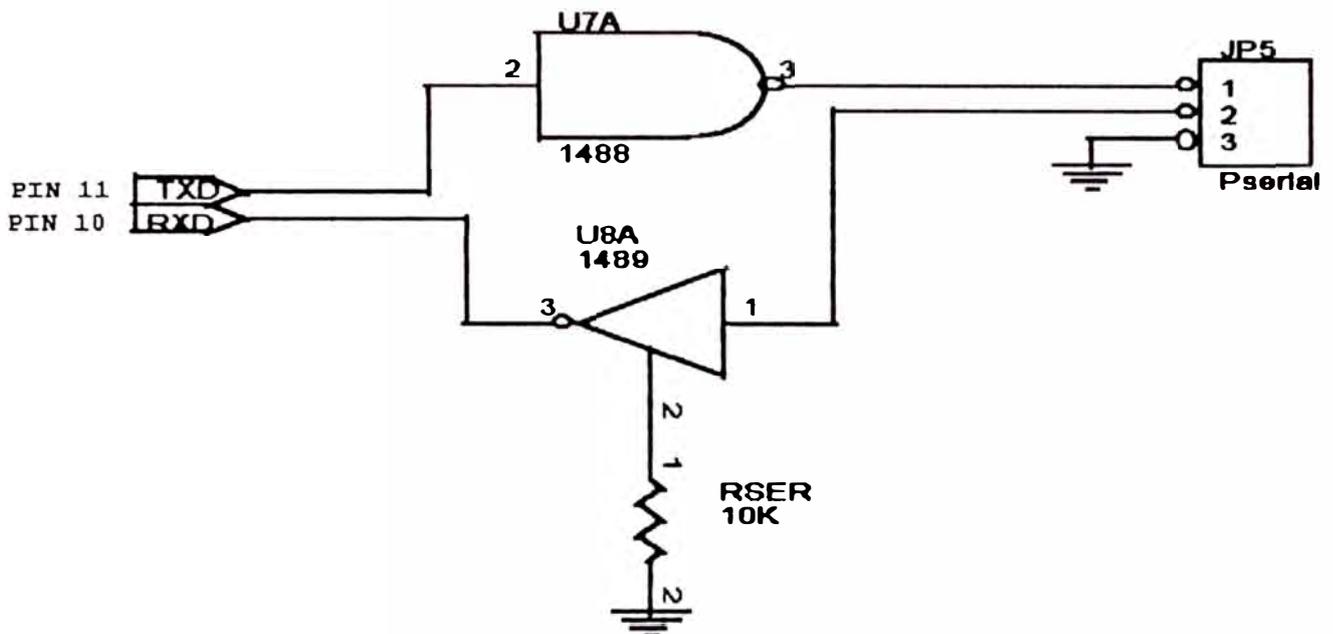


**Figura 3.14 Niveles de tensión de norma RS232**

Conversión de este sistema de lógica negativa a un sistema de lógico TTL se tiene la necesidad de convertir estos voltajes a niveles TTL para que el 8031 los pueda detectar. Para esto se utilizan los microcircuitos MC1488 y MC1489.

El MC1488 convierte un nivel de voltaje TTL a un nivel RS-232 y el MC1489 convierte un nivel RS-232 a un nivel TTL. La configuración utilizada se muestra en la figura 3.15 se conecta la entrada y la salida estos circuitos a las

terminales de recepción y transmisión respectivamente. Además se utiliza un cable telefónico para conectar los dos puertos seriales, se usa un canal para recepción y otro para transmisión.



## ETAPA DE COMUNICACION SERIAL

**Figura.3.15 Circuito comunicación serial**

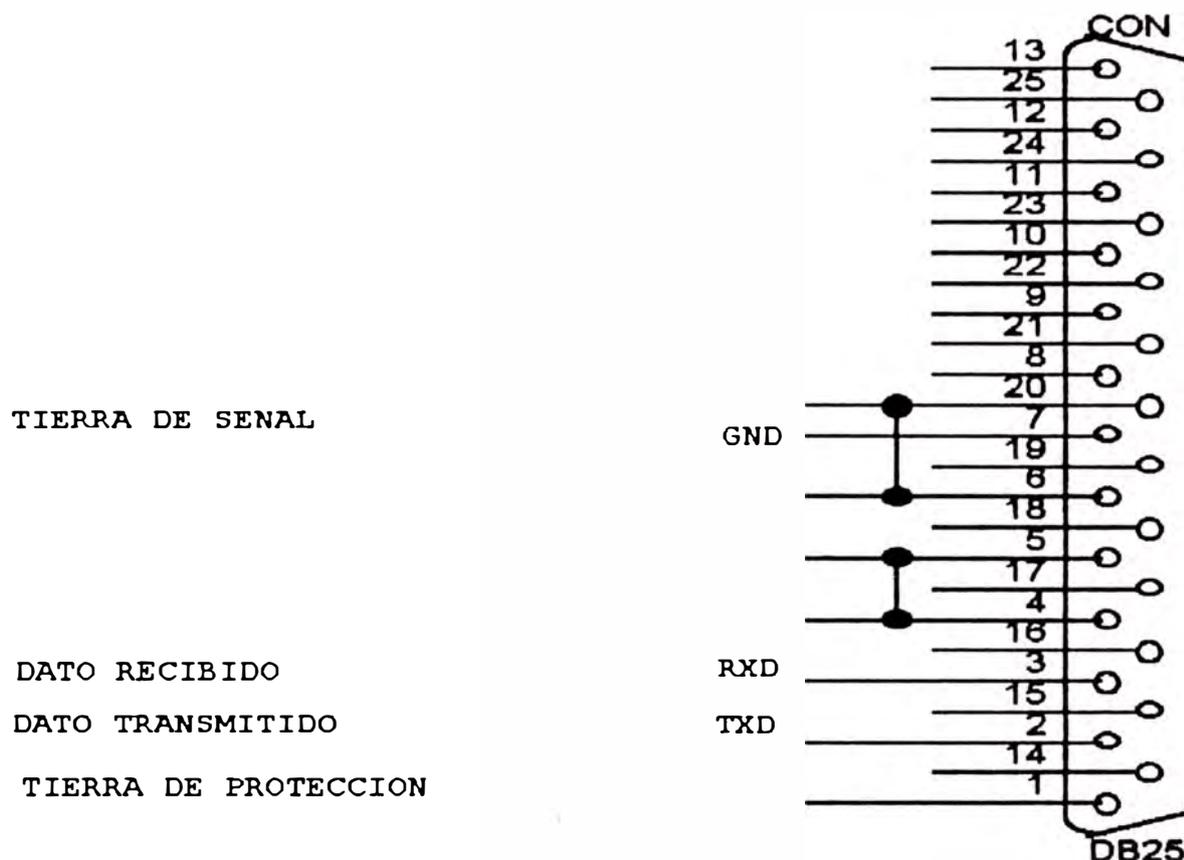
En la PC se tiene un conector DB\_25 macho tiene 25 terminales , pero la mayor parte del tiempo sólo se emplean ocho de ellas y lo más común es que se utilicen únicamente tres

- PIN 2 TXD
- PIN 3 RXD
- PIN 7 GND

Los terminales TXD, RXD, GND son los que se necesitan para establecer la comunicación serial, considerando el protocolo de software para controlar el flujo de datos en serie.

Los pines 4 ,5 y6 y 20 se emplean para controlar por hardware el flujo de datos en serie y deben estar conectados tal como se muestra la figura

definiciones:



**figura.3.16 configuración de pines Puerto serial PC**

#### **PIN 4 SOLICITUD DE ENMIO**

Este pin se emplea para solicitar la transmisión del dispositivo DTE al dispositivo DCE.

#### **PIN 5 LIBRE PARA ENMIO**

Este pin se utiliza para indicar al dispositivo DTE que el dispositivo DCE está listo para recibir datos en serie

#### **PIN 6 UNIDAD DE DATOS LISTA**

Esta pin se usa para indicar al dispositivo DTE que el dispositivo DCE esta encendido y conectado al canal de comunicación

#### **PIN 20 TERMINAL DE DATOS LISTA**

Utilizado para indicar al dispositivo DCE que el dispositivo DTE está listo para trabajar.

### **3.5 Fuente de alimentación**

La fuente de alimentación general para todo este montaje se representa en la figura 3.16 A.

Es una fuente perfectamente estabilizada y sobradamente capaz para alimentar todo el proyecto.

Sus características son:

Tensión de alimentación	Alterna 220 consumo máximo de 60 voltios
Tensión de salida	Continua estabilizada 12 V, - 12V, 5 V
Intensidad de corriente	Máxima 1.5 A.

Protegida frente a cortocircuito en el circuito de salida  
totalmente aislada de la red eléctrica

Las características de los materiales de esta fuente son:

Transformador	primario 220 V. secundario 15-0-15 V, 1.5-A 9 V 1.0 A
Puente rectificador	50 V 10 A
Estabilizador de voltaje	C.I. 7805, CI 7812
Condensador	Electrolítico 2, 200 $\mu$ f 50 V
Condensador	Electrolítico 1 $\mu$ f 50 V

Transistor                      BD 136, BD 135

Diodo Zener                    12V

### **Circuito total del proyecto**

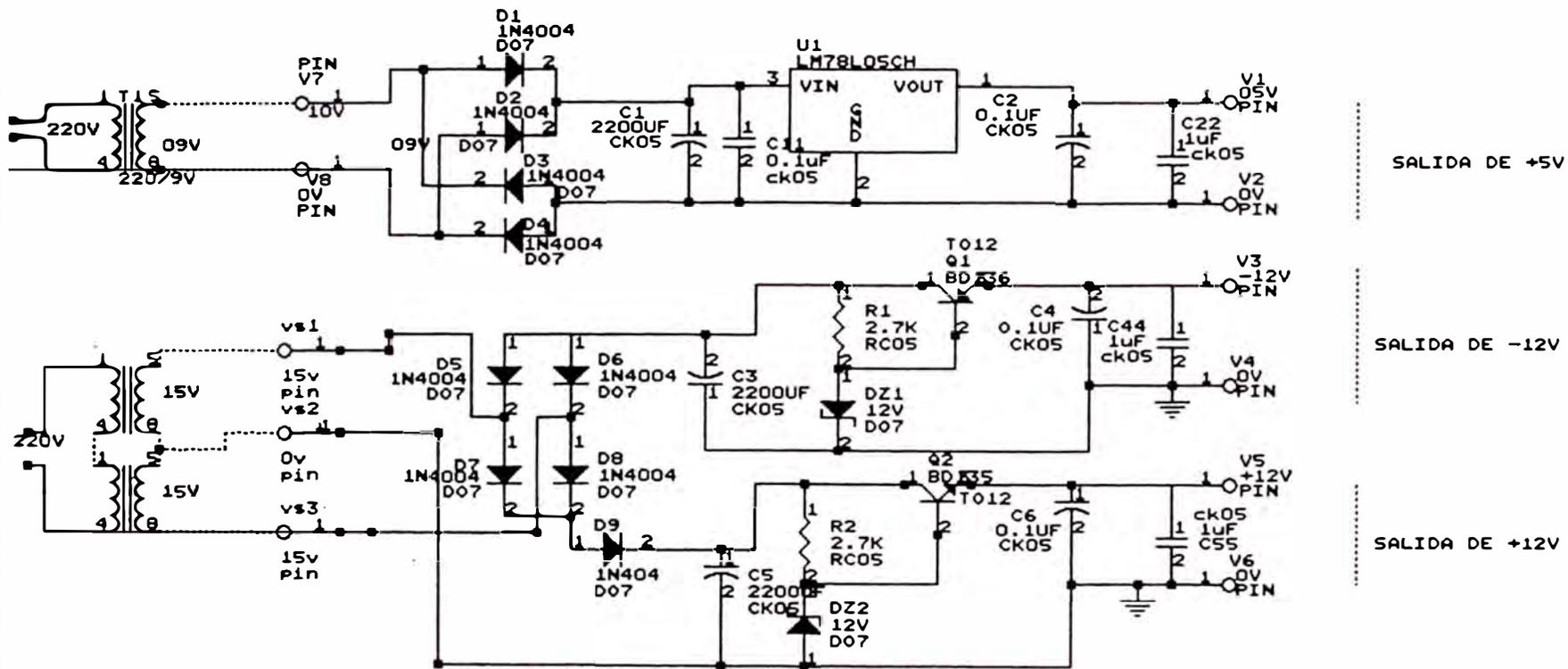
Con el conjunto de circuitos ( etapas) hemos realizado ni mas ni menos que un buen y potente **AUTOMATA PROGRAMABLE** es un autómata programable capaz para más de 16 entradas y 16 salidas

En cuanto a la capacidad de memoria de este diseño ,puedo asegurar que con 2 Kbytes de memoria de programa supone una equivalencia de más de 10 Kbytes de memoria de programa de un autómata comercial.Esto es debido a que el automata comercial está basado en funciones preprogramadas y el programa usuario hace llamadas a estas funciones ;en cambio en nuestro diseño el programa está escrito directamente en ensamblador ; es decir en lenguajes máquina

Cada una de estas funciones de un autómata comercial es un programa pequeño o grande ; de este programa que forma la función , se utiliza todo o una parte , pero el programa función queda escrito completo en la memoria ocupando un sitio normalmente mayor que el necesario para lo que se ha programado en ese determinado trabajo , en cambio en nuestro caso sólo se programa estrictamente lo necesario , ni una instrucción más que las apropiadas para el trabajo determinado . Y siendo además posible diseñar una tarjeta con toda la posibilidad de direccionamiento , que en memoria no tiene limite

Finalmente adjunto el circuito de implementación del modulo ver anexo

## FUENTE DE ALIMENTACION



## FUENTE DE ALIMENTACION

FUENTE DE ALIMENTACION		
Size	Document Number	REV
B	TARJETA DE PROYECTO FIG 3.16A	
Date:	December 28, 1999	Sheet of

## **CAPITULO IV DISEÑO DEL SOFTWARE DEL SISTEMA**

### **4.1 Aspectos generales**

El diseño del software consiste de un programa monitor. La función del programa monitor, es proporcionar al usuario un conjunto de herramientas para ayudarlo a desarrollar, depurar, y ejecutar un programa [2]. Gestiona los recursos del sistema al eliminar manipulaciones repetitivas de código máquina. En forma particular el programa monitor incluye las funciones siguientes:

Almacenamiento de datos en RAM, verificación y modificación de localidades de memoria, ejecución de programas almacenados en RAM, programación desde la PC, y transferencia de datos del sistema hacia la PC.

El desarrollo del programa monitor, de algoritmos, y de diagramas de flujo se dividió en módulos, los cuales realizan funciones específicas, estos módulos fueron: módulo de inicialización, almacenamiento de datos, verificación ó modificación, ejecución de programas y módulo de habilitación ó deshabilitación de la comunicación serial.

### **4.2 Programa monitor**

El programa monitor incluye las siguientes funciones:

- Almacenamiento de datos en memoria RAM
- Verificación y modificación de localidades de memoria
- Ejecución de programas almacenados en RAM

- Programación desde la PC
- Transferencia de datos del sistema hacia la PC

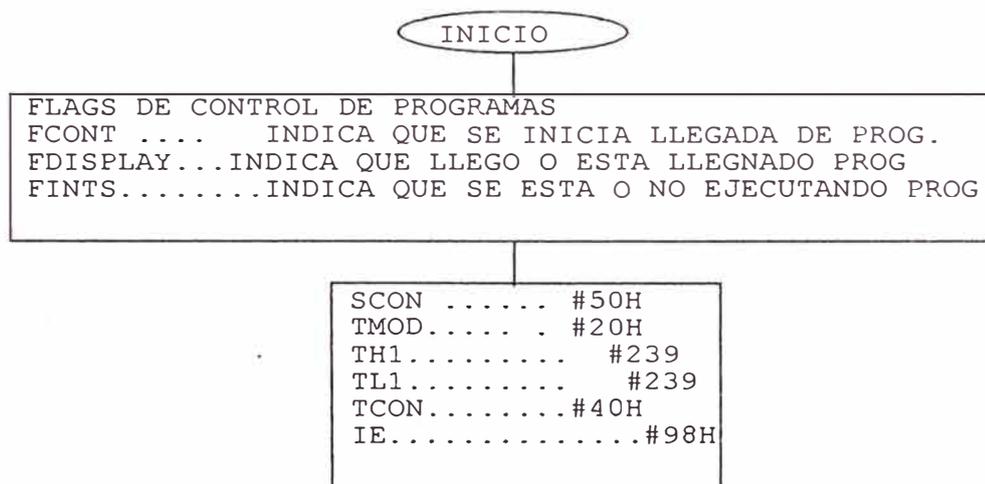
Partes del programa monitor

El programa desarrollado se dividió en :

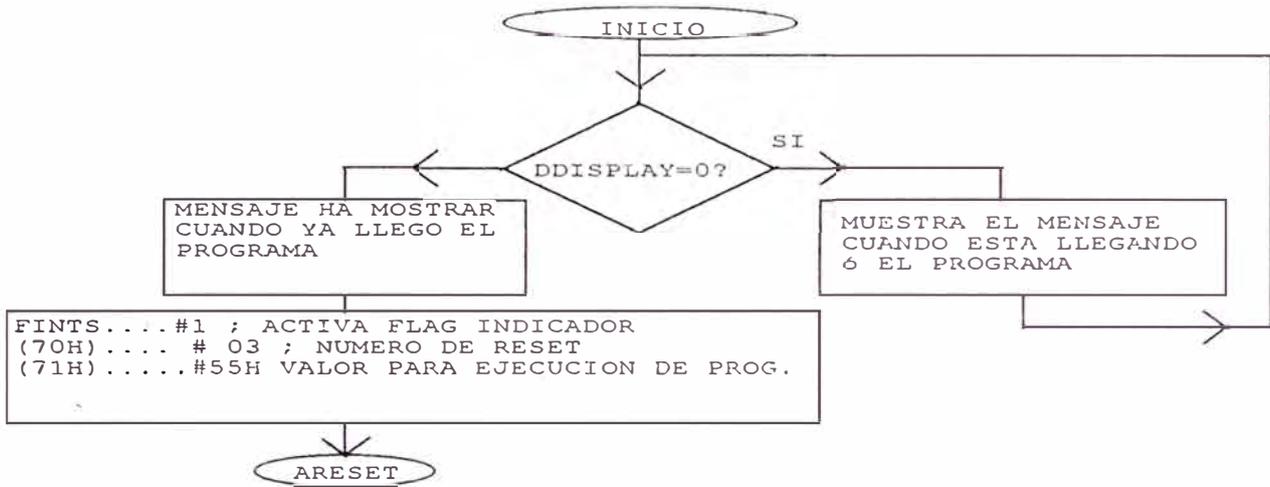
- Módulo de inicialización
- Módulo de visualización de grabación del programa
- Módulo de almacenamiento de datos, verificación
- Ejecución de programas
- Habilitación de comunicación serial.

### Diagrama de flujo:

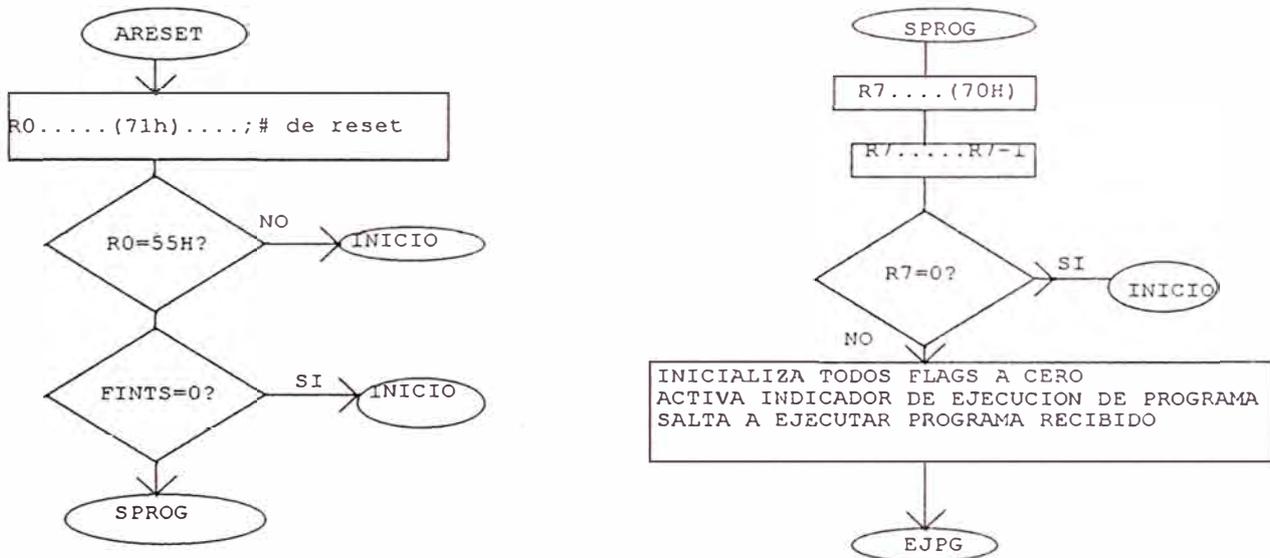
Se dispone de una **subrutina de inicialización** para programar y configurar el timer 1 , registro de comunicación serial, registro de cuenta de los timers y los flags de control del programa.



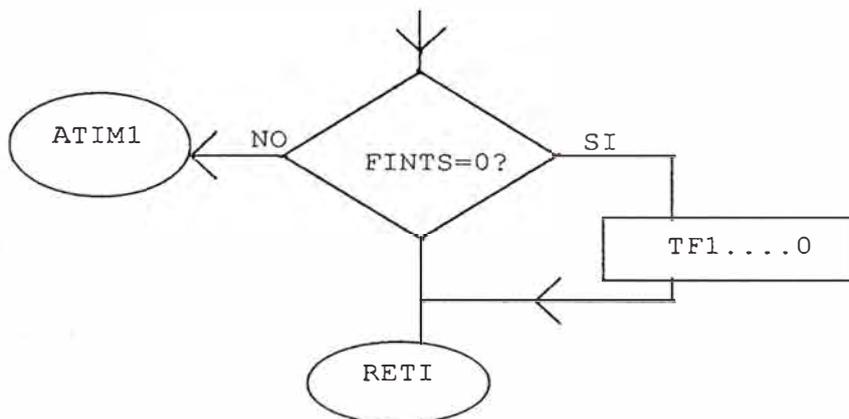
**Subrutina de visualización :** Nos permite visualizar el mensaje mientras llega ó no el programa y cuando llega.



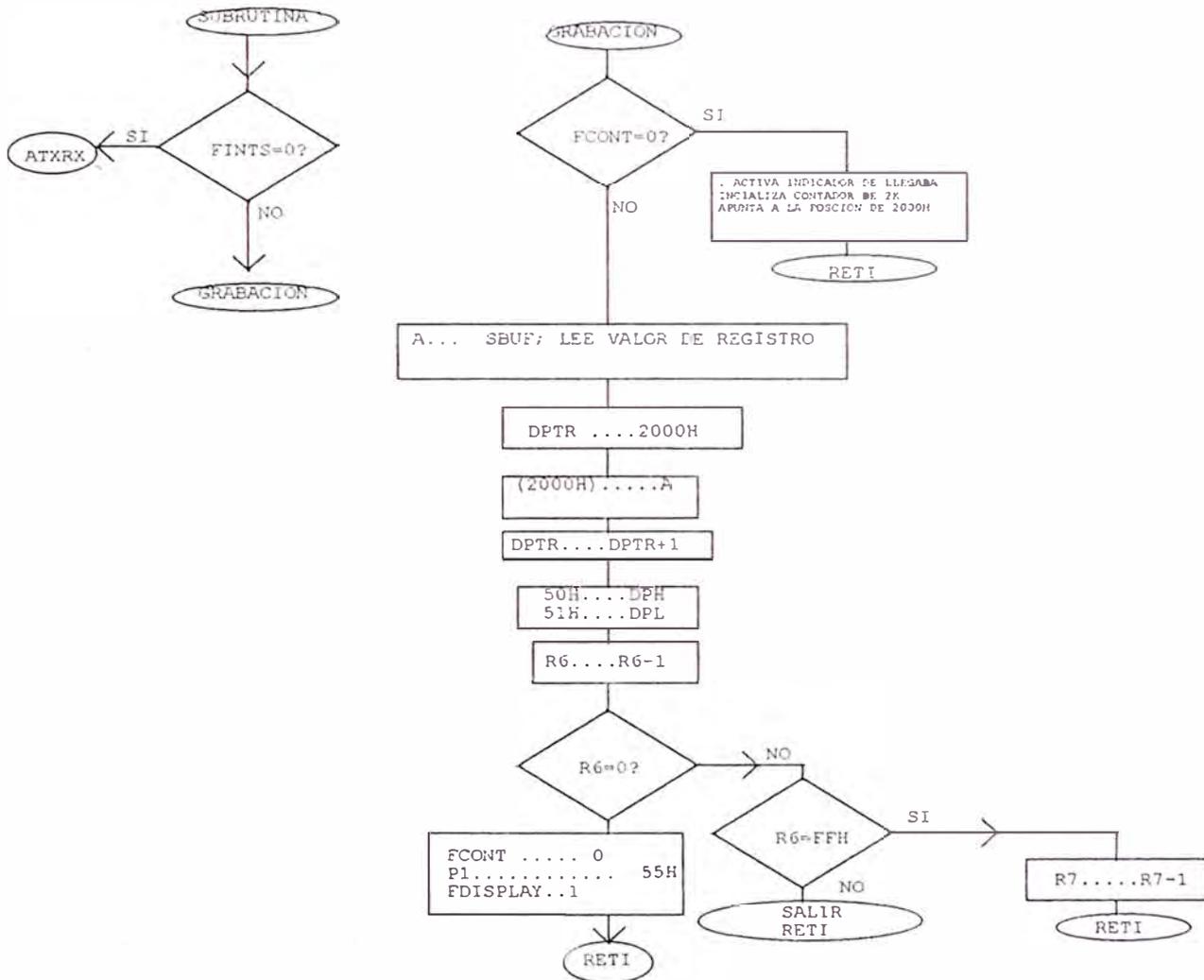
Subrutina de Reset nos permite realizar el proceso de empezar el programa bios o ejecutar el programa grabado en RAM



Subrutina de interrupción interna Timer 1 para realizar la comunicación serial ( nos permite configurar la velocidad



Subrutina de comunicación serial realiza la función de grabar el programa en la memoria RAM , y también nos permite atender la comunicación del programa grabado.



```

ETIQUETAS Y DIRECCIONES DEL PROGRAMA
RECIBIDO:
ORG .....2000H
EJPG
ORG .....2003H
AINT0
ORG .....200BH
ATIM0
ORG .....2013H
AINT1
ORG .....201BH
ATIM1
ORG 2023H
ATXRX
    
```

FIN

## **CAPITULO V DESARROLLO Y PRUEBAS DEL MODULO**

### **5.1 El Hardware**

Como es natural y comprensible, todo lo que hemos diseñado no serviría de nada si no tenemos un medio para ponerlo en práctica y hacerlo útil; es decir obtener un trabajo útil con el microcontrolador.

Cualquier microcontrolador necesita para su trabajo apropiado con elementos periféricos que cumplen misiones específicas necesarias según que casos , además lógicamente , una alimentación adecuada.

Primero, se ha desarrollado el esquema electrónico correspondiente utilizando el Programa Capture for Windows; posteriormente se ha diseñado el PCB mediante el programa Layout for Windows; ambos programas están fuertemente integrado bajo el nombre de ORCADWIN for Windows.

El primero sin duda, tiene que ser precisamente el circuito donde se encuentra el micro en cuestión . a este circuito, que montado consta de una tarjeta principal de circuito impreso de doble cara. Consta de las partes siguientes:

- La Unldad central de proceso
- La etapa de visualización
- La etapa de Interfase de puertos
- La etapa de comunicación serial

En la etapa de CPU puede emplearse el 8031 ó 8051 ó 8751 - Intel

## 5.2 La tarjeta CPU

Los materiales para la CPU consta de:

### Circuitos integrados

- 1 microcontrolador que puede ser 8031,8051, ó 8751 Intel
- 1 memoria EEPROM del tipo 2816 - 2 Kx8 ( 2048 Bytes)
- 1 memoria RAM del Tipo 6116 - 2Kx8 ( 2048 Bytes)
- 1 Integrado Octal Latch 74LS373
- 1 Integrado decodificador de 3 a 8 líneas 74LS138
- 1 Integrado puerta AND 74LS08
- 1 cristal de cuarzo para 12 MHZ
- Material Varios

### 5.2.1 Características de la etapa de CPU

Las características de esta etapa son las siguientes:

- 2 Kbytes en memoria EEPROM para códigos de programa
- 2 Kbytes I/O en memoria RAM para datos, direccionable a partir de la dirección 2000h.
- 2 entradas de interrupciones
- 2 entradas de temporizadores / contadores

### 5.2.2 Funcionamiento de la CPU

En el momento de conectar la tarjeta a su alimentación se produce el **reset**.

Transcurrida la operación de **reset** , el micro procede a buscar la primera instrucción. Esto es automático; es decir, en los microprocesadores, su

primera intención es después del **reset** debe ser, necesariamente, buscar la primera instrucción. Esta primera instrucción tiene que estar situada obligatoriamente en la dirección 0000H y además tiene que ser una sentencia; es decir, tiene que ser una instrucción que implique un trabajo de inmediato, generalmente es una instrucción de salto incondicional, bien a una subrutina de inicialización o al principio del programa, que ha de estar necesariamente, después de los vectores de interrupción

Por Hardware; es decir, por construcción, el micro tiene todos sus buses con el valor 0. En el momento de conectar se presenta en el bus de dirección la dirección 0000H. Luego las señales de control realizan el siguiente proceso:

- Lectura del código / instrucción de la memoria de programa

Para leer un código / instrucción, las señales que se utilizan son exclusivamente las señales ALE y PSEN mientras que RD y WR permanecen en alto. Naturalmente, esta función la hace automáticamente el micro después de ejecutar cada instrucción.

- Lectura escritura de datos en memoria RAM

Para leer o escribir un dato, las señales que se utilizan son la señal ALE, que es imprescindible, y la señal RD o WR respectivamente, mientras la señal PSEN permanece en alto 1. Para que se realice esta función, es condición indispensable que nosotros lo hemos programado; es decir que sea una instrucción de transferencia de un dato, como es:

La función de la patilla ALE es la misma que para la lectura del código /instrucción; es decir para fijar en el integrado 74LS373 el valor del byte bajo de la instrucción.

La patilla PSEN adquiere en este modo el valor 1, por lo que la memoria EEPROM queda inhabilitada en todo el proceso.

Durante el proceso de fijación de la dirección en el integrado 71LS373, la patilla RD tiene el estado 1. Este pin está conectado a la puerta AND, y que en la otra entrada de la puerta AND está conectada la patilla PSEN. El resultado de esto es que cuando  $RD = 1$  y  $PSEN = 1$  la salida será en este caso  $D/I = 1$ , luego observando la salida de D/I se conecta a la patilla OE de la memoria RAM estando en el estado alto, luego la salida del dato de dichas memorias no están autorizadas

Instantes después, cuando han transcurrido los impulsos de reloj prefijados en el hardware del micro en los que la dirección ha quedado definida, la patilla RD pasa al estado bajo, con esta condición la salida D/I pasa a estado bajo, aplicándose a la patilla OE de la memoria RAM un nivel bajo, de forma que ya puede salir el dato de la RAM.

La tarjeta que acabamos de describir es , como ya he dicho , la que denominamos CPU por tener en su constitución física el microprocesador en cuestión.

### **Puertos de entradas / salidas**

La tarjeta CPU dispone de puertos de entradas salidas de acuerdo a la descripción realizado que son lo siguiente

- Puerto de salida ubicado en la posición de memoria 0A000H
- Puerto de entradas ubicado en la posición de memoria : 8000H , 0C000h
- Puerto de entrada analógica : ubicado en 0F000h
- Puerto de comunicación serial

- Puerto bidireccional del usuario P1

Para las entradas y salidas he utilizado integrados del tipo 74LS373 en tecnología TTL , este elemento se encarga de transferir el dato hacia la CPU o desde la CPU

La forma de realizar la salida o ingreso de datos es por software Por ejemplo para salidas

- Ubicación del puerto : 8000h puerto de entrada

```
MOV DPTR, # 8000H
```

```
MOVX A,@DPTR
```

- Ubicación del puerto de salida: 0A000h

```
MOV DPTR,#0A000H
```

```
MOVX @DPTR,A
```

Ubicación de puerto de entrada analógico : 0F000H

■ MOV DPTR,#0F000H ; ubicación del puerto o ADC

■ MOVX @DPTR,A ; Orden de realizar conversión

■ CALL DELAY; ; Retardo de conversión

■ MOV DPTR,#0F000H ; Apunta nuevamente a la posición del ADC

■ MOVX A,@DPTR ; Lee el valor convertido

Se recomienda que estos puertos de salidas deben estar aislados con optoacopladores.

### **La etapa de visualización**

Esta etapa esta formado por cuatro display ánodo común con sus respectivas transistores de activación , se utiliza dos latches 74LS373. Para sus direccionamiento respectivo.

## **CAPITULO VI APLICACIÓN DEL SISTEMA PARA EL CONTROL DE UN PROCESO DE TRATAMIENTO TERMICO**

### **6.1 Consideraciones generales**

En esta parte del presente trabajo tiene por finalidad diseñar y realizar la construcción de un SISTEMA DE CONTROL PROGRAMABLE PARA HORNOS DE TRATAMIENTO TERMICO basado en el sistema microcontrolador 8031 -Intel para la industria y laboratorios de tratamiento térmico.

Este trabajo muestra el diseño de un sistema de control digital que efectúa una regulación precisa de la temperatura en el que adicionalmente se puede programar las evoluciones del punto de consigna ( referencia ), mediante la definición Previa de una secuencia de tramos variables en su composición y en su longitud, los cuales pueden ser de tres tipos:

- . Subidas de temperaturas de consigna hacia una cota cualquiera, en un tiempo programable
- . Estabilizaciones de la temperatura de consigna durante un tiempo prefijado
- . Bajadas de temperatura de consigna hacia una cota cualquiera .

En el presente trabajo se muestra un caso particular de remplazo de los controladores analógicos existentes en un sistema de control por controlador digital, por razones de

- a) Confiabilidad

- b) Eficacia (mantener la calidad del producto)
- c) Costo
- d) Precisión
- e) Flexibilidad
- f) Evitar lesiones al personal que opera el horno o daño al equipo. (seguridad)

## **6.2 Descripción del proceso**

Los hornos de tratamiento térmico tiene por objeto, mejorar las propiedades y características de los aceros, y consiste en calentar las piezas y herramientas a temperaturas adecuadas, durante un cierto tiempo y luego enfriarlas en condiciones convenientes, de esta forma se modifica la estructura microscópica de los aceros, se verifican transformaciones físicas y a veces hay también cambios en la composición del metal.

El tiempo y la temperatura son los factores principales y hay que fijarlos siempre de antemano , de acuerdo con la composición del acero , la forma y el tamaño de las piezas , y las características que se quieren obtener.

Luego podemos indicar los hornos de tratamiento térmico deben operar a una temperatura concreta o seguir un ciclo determinado proceso como:

- . Calentamiento
- . Mantenimiento
- . Enfriamiento

Es realmente importante conseguir dicha temperatura o ciclo de tratamiento en la carga , pero es también importante que el horno no sobrepase una temperatura prefijada para proteger debidamente los elementos metálicos y cerámicos de su interior

El tema de regulación de temperatura en los hornos y, sobre todo, de la precisión es extraordinariamente importante y fuente, en la práctica de muchos errores por los siguiente puntos:

a) Una cosa es el termopar de detección de la temperatura en un punto del interior del horno junto con el equipo de regulación, y otra muy distinta la carga, en la que se pretende conseguir una cierta uniformidad de temperatura, no solo en la superficie, sino en toda la masa.

El equipo de regulación señalará y controlará con precisión, rapidez la temperatura correspondiente a la fuerza electromotriz que reciba del termopar y actuará en consecuencia sobre los elementos actuantes como los contactores, triacs, etc. que ajustarán la potencia al valor adecuado, pero deberá tenerse en cuenta que tiene una funda protectora que retrasará la recepción de la FEM correspondiente a la temperatura interior del horno, y además el termopar tendrá un error propio, es decir La FEM generada y enviada no corresponderá exactamente a la temperatura del interior del horno.

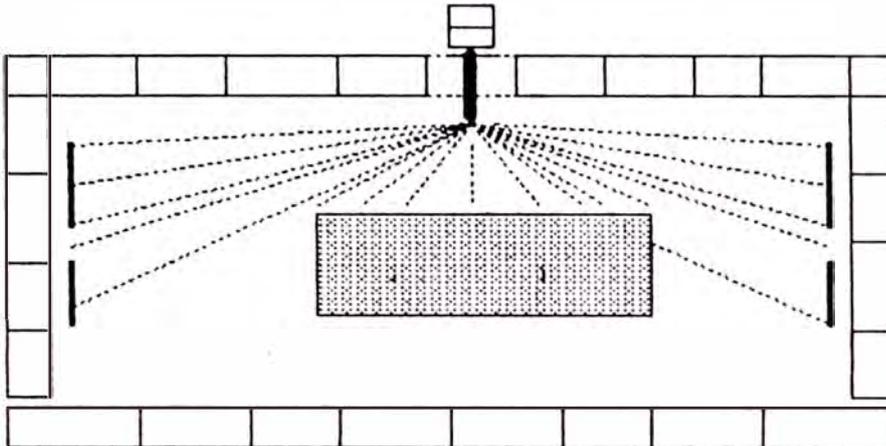
b) El termopar está situado en un punto concreto del interior del horno recibiendo, por ejemplo, radiación, no solamente de la carga, sino también de las resistencias y de las paredes refractarias, especialmente durante el calentamiento la diferencia de temperatura entre las resistencias, el refractario y la carga es muy elevada, señalando el termopar una temperatura intermedia entre ellas.

Luego por todo lo anterior expuesto es conveniente para la precisión considerar otros factores como:

. Disposición de las resistencias

. Efecto enfriador en los extremos del horno

Observar figura.6.1



**Figura 6.1 Termopar en un horno**

c) Finalmente el equipo de regulación no tiene, evidentemente, ninguna influencia en la conductividad térmica de la carga por lo que se puede tener una aceptable uniformidad de temperatura a lo largo del horno, es decir, en la superficie de la carga, pero una fuerte diferencia de temperatura entre la periférica y el centro .intervienen en este caso, mucho más que el equipo de regulación la disposición de la carga en capas por ejemplo, para facilitar la radiación al interior y para facilitar la transmisión por convección a la carga.

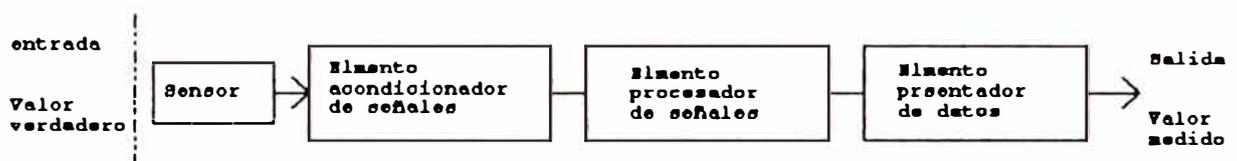
### **6.3 Sistema de control de temperatura**

En un equipo de regulación de temperatura se distinguen los siguientes elementos fundamentales; en este caso del horno a automatizar:

- Elemento sensor: que se le conoce como elemento primario, está en contacto con el proceso y genera una salida, la cual depende de la variable a medir (temperatura), luego el elemento primario es el termopar, donde la F.E.M en millivoltios depende de la temperatura.

- **Elemento acondicionador de señales:** Toma la salida del elemento sensor y lo convierte en una forma más adecuada para un procesamiento adicional, en este caso en un voltaje de C.D
- **Elemento procesador de señales:** Toma de la salida del elemento acondicionador y lo convierte en una forma mas adecuada para la presentación, en nuestro caso se tiene el convertidor análogo - digital.
- **Elemento presentador de datos:** Presenta el valor medido en una forma que el observador puede reconocer fácilmente

Hasta aquí se puede considerar como el sistema de medición de la variable a controlar ver figura 6.2.



**Figura 6.2 Estructura general de medición**

- **Controlador (regulador).**- Es el cerebro del sistema de control, recibe como señal el punto de consigna y la variable medida.
- **Elemento final de control:** Es el elemento que amplifica la potencia de la señal del controlador y suministra al proceso, actúa en consecuencia sobre la entrada de corriente en los hornos calentados eléctricamente.

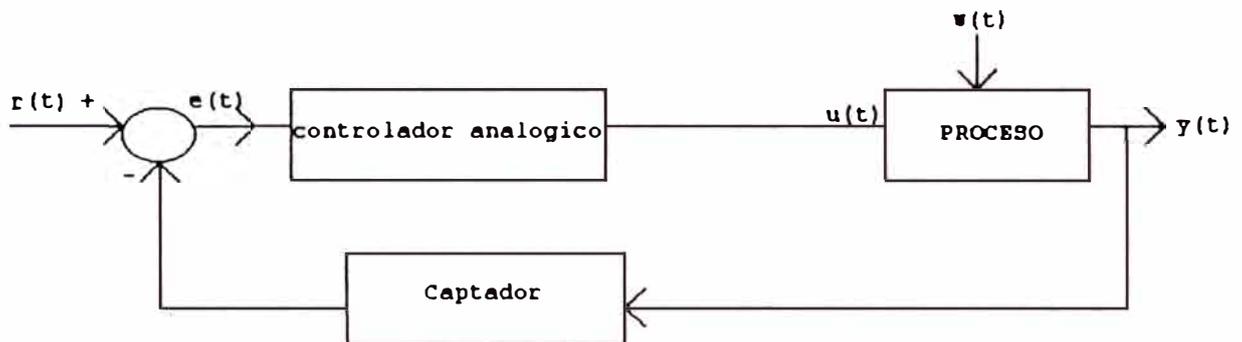
La importancia de estos elementos estriba en que realizan tres operaciones básicas que deben estar presentes en el sistema de control.

1.- **Medición ( M ).**- La medición de la variable que se controla, se hace generalmente mediante la combinación del sensor y transmisor.

2.- **Decisión (D).**- Con base en la medición, el controlador decide qué hacer para mantener la variable en el valor que se desea

3.- Acción (A).- como resultado de la decisión tomada del controlador se debe efectuar una acción en el sistema , generalmente ésta es realizada por el elemento final de control

En la figura 6.3 se muestra el sistema de control y sus componentes básicos.

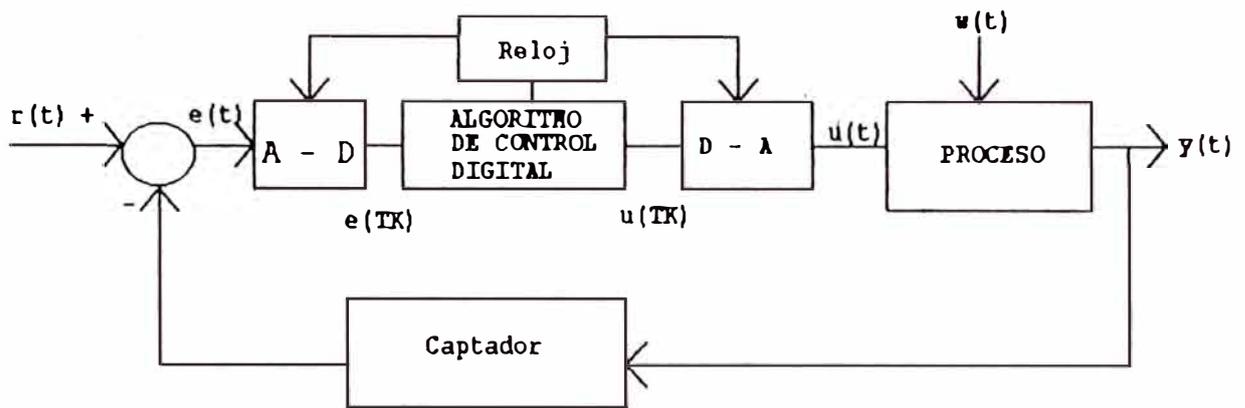


**Figura.6.3 Bucle de control analógico**

Tradicionalmente la regulación y seguimiento de puntos de consigna requiere de las tres operaciones indicadas obligatoriamente para todo el sistema .

El controlador se diseña para que la salida  $y(t)$  siga a la referencia  $r(t)$ , aunque se presentan perturbaciones externas  $w(t)$  sobre el proceso y errores representados por la acción de  $v(t)$  en las medidas de los sensores.

El presente trabajo está relacionado con las aplicaciones de los microcontroladores en regulación y seguimiento automático de puntos de consignas, en este caso el microcontrolador realiza las funciones denominadas de << control digital directo >> cerrando el bucle de regulación en sistemas de control, mediante algoritmos apropiados. Ver figura 6.4



**Figura 6.4 Bucle de control digital**

Las variables asociadas al proceso evolucionan de forma continua en el tiempo. Por consiguiente es necesario utilizar el empleo de convertidores analógico - digital para suministrarles la señal o señales al controlador. Así mismo si los actuadores son también analógico será necesario los convertidores digital - analógico.

Por otra parte, en nuestro caso no existe señal de referencia analógica, sino que estas se introducen como consignas en el propio controlador digital.

Por último el computador (controlador) debe estar sincronizada con el proceso, por ello es necesario un reloj en tiempo real.

Típicamente el CAD suministra un número al computador cada vez que llega un pulso de reloj. Los números se suministran a un período constante  $T$  (período de muestreo) .

La señal de entrada al computador (controlador) puede representarse mediante una variable  $e(kT)$  que toma valores en intervalos discretos de tiempo  $K= 0,1,\dots$ . La ejecución del algoritmo de control da como resultado una secuencia de números representada mediante la variable  $u(kT)$  que se aplica al proceso, a través de un CDA con el mismo período  $T$ , La señal analógica

suministrada al actuador se representa mediante la variable  $u(t)$  que mantiene un valor constante  $u(t) = u(kT)$  para  $kT < t < (k+1)T$

### 6.3.1 Medición de temperatura

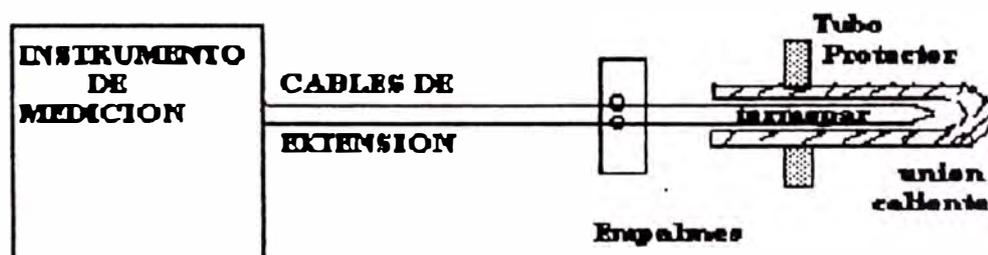
En la industria metalurgia como sensores se utilizan:

- Termopares o pirómetros termoeléctricos hasta 1400 °C
- Lunetas de radiación a partir de 900°C.

Los más utilizados en trabajos de metalurgia para medir y controlar temperaturas son los pirómetros

El pirómetro sencillo, mostrado en la figura 6.5 consta de las siguientes unidades .

1. El termopar, compuesto de dos metales o aleaciones diferentes
2. El bloque de empalmes, situado cerca y fuera del horno
3. Los cables de extensión
4. El instrumento indicador o de registro.



**Figura 6.5: Pirómetro termoeléctrico sencillo**

El funcionamiento de este pirómetro se basa en dos principios:

**Efecto Peltier** : Si dos alambres metálicos diferentes se ponen en contacto eléctrico, habrá una FEM. A través de la punta de contacto. La magnitud de

la FEM desarrollada se determina por la composición química de los alambres y la temperatura de la unión.

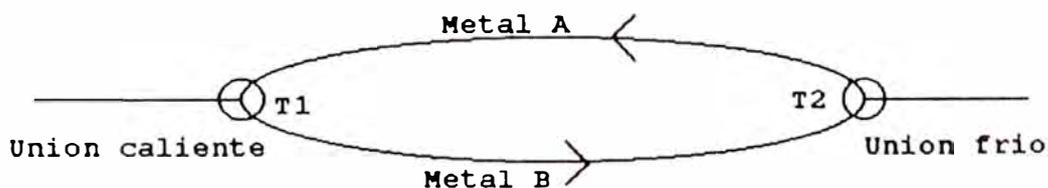
**Efecto Thomson:** Si hay una diferencia de temperatura entre los extremos de un alambre homogéneo único, existirá una FEM. Entre los extremos de dicho alambre, la magnitud de la FEM. Desarrollada quedará determinada por:

- composición química de los alambres
- La uniformidad química de los alambres
- Diferencia de temperatura.

Por lo tanto, la FEM total en un pirómetro termoelectrónico, algunas veces llamado **EFFECTO SEEBECK** es la suma algebraica de cuatro FEM:

- Dos FEM. De Peltier en las uniones caliente y fría
- Dos FEM. De Thompson a lo largo de cada alambre

Un termopar es un circuito cerrado que consta de dos empalmes (figura 6.6)



**Figura 6.5: Circuito de un termopar**

La FEM. Depende de las temperaturas  $T_1$ ,  $T_2$ , de ambos empalmes. La  $T_1$ , es la temperatura por medir, o sea la temperatura del empalme de medición, la  $T_2$ , es la temperatura del empalme de referencia.

A fin de deducir con precisión el valor de  $T_1$  a partir de la FEM medida, debe conocerse la temperatura  $T_2$ , del empalme de referencia. Si la

temperatura T2 o unión fría o referencia se mantiene a temperatura constante, entonces la FEM. Será función definida de la temperatura de la unión caliente.

Mediante una compensación adecuada y calibración es posible determinar una relación exacta entre FEM. Desarrollada y la temperatura de la unión.

### **Cables de conexión.**

El propósito del cable de extensión es mover la unión de referencia a un punto donde la temperatura no variará.

Los alambres del termopar no suelen ser suficiente largos ni se hallan tan aislados como para conectarse en forma directa al instrumento, los cables de extensión por lo general están hechos del mismo material que los alambres del termopar y se colocan en un cable doble, con la cubierta individual según el código de color para su identificación.

### **Tipos de termopares**

En teoría, dos alambres metálicos distintos desarrollan un FEM. Cuando hay una diferencia de temperatura entre sus puntos de unión; sin embargo desde el punto de vista de la industria, solo unas cuantas combinaciones se utilizan realmente para termopares, estos se escogen por:

- Potencial termoeléctrico
- costo razonable
- Estabilidad del tamaño del grano
- Linealidad de la curva Temperatura - FEM
- Puntos de fusión mayores que la temperatura que se va medir

El primer material en la combinación siempre se conecta al positivo.

Los termopares basados en metal ( base metal ) son:

#### **Cromel - Alumel : Tipo K**

El cromel ( 09% de níquel , 10% de cromo )

El alumel ( 94% de níquel , 3% de manganeso , 2% aluminio y 1% silicio )

Es una de las utilizadas industrialmente Por:

- Curva de calibración bastante lineal
- Buena resistencia a la oxidación
- Es útil en el intervalo de 1200 a 2000 °F

**Hierro - Constantan : Tipo J**

El constantan es una aleación que contiene 54% de cobre y 46 % de níquel es importante por.

- Bajo costo
- Alta energía termoeléctrica
- Adaptabilidad a atmósfera diferentes
- Rango de uso en temperaturas de 300 a 1400 °F

**Cobre - Constantan : Tipo T**

La aleación de constantan empleada con cobre difiere ligeramente de la utilizada con hierro y podría pequeñas cantidades de magnesio y hierro . esta combinación es mas adecuada para mediciones a bajas temperaturas : - 420 °F hasta 600 °F.

**Platino , Platino - Rodio al 10% :Tipo R**

Este es un termopar de " metal noble". Se utiliza para medir temperaturas demasiado altas para los termopares de base metal y en casos en que la radiación o los pirómetros ópticos no son satisfactorios

Rango de temperatura a utilizar es 32 a 3000 °F .

Los dos alambres están aislados entre si por pequeños cilindros de porcelana o tubos de cerámica

Los termopares se encuentra en tubos de porcelana o metálicos , el tubo protege al termopar contra daños mecánicos y previene la contaminación de los materiales del termopar por la atmósfera del horno.

**6.3.2 Regulación de temperatura**

El método de regulación mas adecuado en un horno industrial depende de:

- La importancia del proceso
- La inercia térmica de la carga y el horno
- La frecuencia de carga y descarga y variaciones posible en su peso
- Las diferencias admisibles entre las temperaturas de la carga , del interior del horno y del índice de regulación
- La gama de temperatura de funcionamiento del horno
- El método de calentamiento y variaciones en la potencia nominal

En la mayoría de los casos utiliza la regulación todo o nada (control on - off )

### **6.3.3 Regulación de potencia**

Se realiza mediante dos formas.

- a) Forma intermitente : Mediante contactores
- b) Forma continua : Mediante tiristores

#### **Regulación por contactores**

Durante muchos años la regulación de los hornos calentados por resistencias se realizó casi exclusivamente por contactores, componentes eléctricos simples y de fácil mantenimiento, pero que presentan algunos inconvenientes como:

- Riesgo de pegados de contactos
- Variación en la temperatura de las resistencias del horno conectado a desconectado

Durante La regulación el contactor esta abierto o cerrados , en el tiempo de mantenimiento de la temperatura.

La potencia demanda por el horno durante el tiempo que el contactor está abierto corresponde a la potencia de pérdidas .

Mientras que la potencia conectada durante los cortos tiempos en el que el contactor está cerrado , es la nominal del horno, por lo que las oscilaciones de temperatura son máximas cuando se requiere la máxima precisión de temperatura.

#### **Regulación por tiristores**

Un tiristor se comporta como un interruptor o contactor en el que el número de conexiones o el momento de la conexión puede elegirse libremente o ser tan elevado como se quiera. Además con una pequeña corriente de mando se puede modular una fuerte potencia y la potencia conectada, cuando el tiristor no conduce, es muy reducida (inferior al 1 por 100 de la potencia nominal).

Un tiristor opera por impulsos, determinando su duración los dos modos de funcionamiento posible.

- Impulsos relativamente largos , operando el tiristor por trenes de ondas (una onda en 50 Hz. Tiene una duración de 20 mseg.) es el más adecuado para calentamiento por resistencias metálica, por no producir perturbaciones de armónicos, la duración de un tren de ondas varía entre 0.5 y 10 -20 seg. , lo que en hornos industriales es equivalente a una regulación continua Impulsos relativamente cortos operando el tiristor por ángulo de fase ( duración inferior a 0.01 seg.).

Da lugar a la inyección de armónicos de la red , sin embargo , en el calentamiento por resistencias no metálicas de MOSi2 (fuerte coeficiente de variación de la resistencia con la temperatura) es interesante el modo de ángulo de fase ,durante el final del calentamiento y el mantenimiento.

#### **6.4 Modelamiento matemático del horno a controlar**

##### **6.4.1 Descripción**

En esta parte se tiene dos objetivos principales. el primero presentar el desarrollo del modelo matemático del horno eléctrico y luego del sensor, el cual es necesario para analizar el sistema de control de temperatura del horno, el segundo es obtener y explicar el significado físico de algunos de los parámetros que definen la personalidad del sistema.

El trabajo consiste en diseñar un sistema de control de temperatura que pueda mantener la variable controlada en el punto de control (SET POINT), Cuando ya se ha logrado esto, debe ajustarse el controlador de manera que se reduzca al mínimo la operación de ensayo y error que se requiere para mantener el control. Para realizar un buen trabajo, debe conocerse; las características o "personalidad" del proceso que se va controlar luego una vez conocido la "personalidad del proceso" se puede diseñar el sistema de control y obtener la "personalidad del controlador" que mejor combine con la del proceso.

Para conocer la "personalidad" es conveniente aplicar una señal manipulada (actuante) al horno; Si la señal "I" de entrada al horno se modifica de repente de un valor a otro, la señal de salida "T(t)" denominada variable controlada no cambia en forma instantánea a su nuevo valor, la manera en que el horno responde a cambios súbitos de entrada se conoce como **características dinámicas** y estas se resumen por conveniencia utilizando una función de transferencia  $G(S)$ .

El comportamiento de un horno, en lo referente a su regulación de temperatura, varía notablemente de un tipo a otro, influyendo además considerablemente la carga dispuesta en su interior. Se puede hacer una clasificación de sus características:

- Estáticas
- Dinámicas

#### **6.4.2 Conceptos térmicos: Transferencia de calor**

Los sistemas térmicos son aquellos que comprenden la transferencia o transmisión de calor de una sustancia a otra. Estos se pueden analizar en términos de resistencia y capacitancia, aunque la capacitancia y resistencia térmica pueden no representarse con precisión como parámetros concentrados, ya que suelen estar distribuidos a lo largo de la sustancia, para el análisis se supondrá que un sistema térmico se puede representar por un modelo de parámetros concentrados.

Hay tres medios diferentes en que el calor fluye de una sustancia a otra: conducción, convección y radiación.

Para transferencia de calor por conducción o convección se tiene:

$$Q = \frac{1}{R}(T_1 - T_2) \quad \text{ó} \quad Q = K\Delta T \quad \text{ec.6.1}$$

donde :

$K$  = coeficiente , J/s/ °C

$Q$  = Flujo de energía calorífica :J/S

$R$  = Resistencia térmica : °C / (J/s)

$\Delta T$  =diferencia de Temperatura : °C

El flujo de energía neta a una sustancia afecta a la temperatura de la sustancia de acuerdo con :

$$T = \frac{1}{C}Q \quad \text{ec.6.2}$$

donde:

$C$  = capacidad térmica : J/ °C

Típicamente, debería haber varios caminos para que el calor fluya desde o hasta una sustancia, y el  $Q$  en la ecuación (6.2) debería ser la suma de los flujos de calor de acuerdo con la ecuación (6.1).

Normalmente las propiedades del material están dadas en tablas como:

1. El calor específico a volumen constante  $C_v$ , que es entonces convertido por.

$$C = m C_v$$

donde:

$C$  : capacidad térmica

$m$  : masa de la sustancia

$C_v$ : calor específico

2.- La conductividad térmica  $K$  la cual es convertida a  $R$  por

$$K = \frac{1}{R_t} = \frac{kA}{L}$$

donde :

**A** : área de la sección transversal

**L** : Longitud del paso del camino del flujo

**K** : coeficiente ,J/s/ °C

**k** : conductividad térmica ,en J/ m.s/ °C

### **6.4.3 Características de regulación del horno**

Sus características se clasifican en :

- Estáticas o de estado estable
- Dinámicas

#### **6.4.3.1 Características estáticas o de estado estable**

La principal característica estática de un horno es la que relaciona la temperatura límite ( $\theta_g$ ) que el horno puede alcanzar con la potencia media de calentamiento  $P_m$  (manteniéndose constante la temperatura ambiente ).

En un horno de calentamiento eléctrico  $P_m$  es la potencia aplicada a las resistencias.

No es posible establecer experimentalmente dicha curva por encima de la temperatura  $\theta_{max}$  de utilización del horno. ver figura 6.7

A partir de la curva característica (casi recta hasta  $\theta_{max}$ ) se puede calcular el factor de amplificación.

$$Vs=Tg \quad \alpha = \frac{d\theta_g}{dP_m} \quad \text{ec.6.3}$$

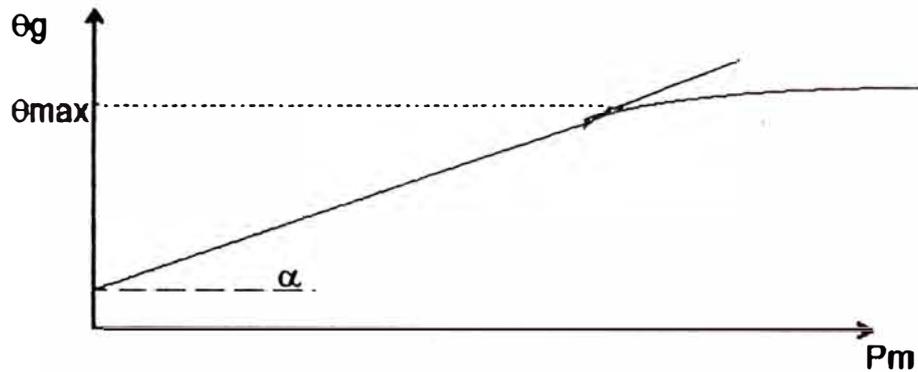


Figura 6.7: Característica Estática del horno

El factor de amplificación  $V_s$  disminuye ligeramente al aumentar la temperatura  $\theta_g$  (véase Fig.6.7) otras características estáticas permiten examinar la influencia sobre la temperatura del horno de magnitudes tales como: Temperatura ambiente, la apertura o cierre de válvulas de entrada de aire, etc.

#### 6.4.3.2 Características dinámicas

Hay que distinguir entre el calentamiento directo (sales, lecho fluidificado, resistencia directa, inducción) y el indirecto, cuando las piezas se calientan por radiación, convección o una combinación de ambas, procedente de las resistencias propiamente dichas dispuestas en las proximidades de la piezas, el cual es nuestro caso de estudio del horno a controlar.

En el horno de calentamiento indirecto, hay que tener en cuenta la resistencia térmica entre la carga y la fuente caliente que suponemos situada en las paredes del horno (resistencias), intervienen en un horno de calentamiento por resistencias las magnitudes siguientes:

$T_c$  : Temperatura de la carga

$T_a$  : Temperatura ambiente

$T_r$  : Temperatura de las resistencias

$T_p$  : Temperatura de la pared interior del horno

$Q_r$  : Calor generado por las resistencias

$Q_{pc}$  : Flujo de calor de las paredes hacia la carga

$Q_p$  : Perdidas de calor a través de las paredes del horno

$C_c$  : Capacidad térmica de la carga

$C_r$  : Capacidad térmica de los elementos de calentamiento

$C_p$  : Capacidad térmica de las paredes del horno

$RT_{pc}$  : Resistencia térmica entre pared y carga

$RT_{pb}$  : Resistencia térmica de las paredes y exteriores al horno

Descripción matemática de las *características dinámicas*

Se verificará el calor a la carga  $Q_r - (Q_{pc} + Q_p) :$

$$Q_r - (Q_{pc} + Q_p) = C_c \frac{dT_c}{dt} \quad \text{ec.6.4}$$

Si la conductividad térmica de la carga es elevada :

$$Q_{pc} = \frac{T_p - T_c}{RT_{pc}} = C_c \frac{dT_c}{dt} \quad \text{ec.6.5}$$

Flujo de calor por la paredes al medio ambiente :

$$Q_p = \frac{T_r - T_a}{Rt_{pe}} \quad \text{ec.6.6}$$

Cuando la capacidad térmica de la pared es muy reducida ,

$$Q_p = C_p \frac{dT_p}{dt} \dots\dots\dots \text{ec.6.7}$$

Cuando la conductividad térmica de la pared es muy elevada, aceptando esto último y sustituyendo (6.7) y (6.5) en (6.4) resulta:

$$\frac{V^2}{R} - C_c x \frac{dT_c}{dt} - C_p x \frac{dT_p}{dt} = C_2 x \frac{dT_r}{dt} \quad \text{ec.6.8}$$

Supongamos como una primera aproximación,  $T_r = T_p$ . Resulta

$$\frac{V^2}{R} - C_c \frac{dT_c}{dt} = (C_r + C_p) \frac{dT_p}{dt} \quad \text{ec.6.9}$$

de la ec. (6.5) deducimos que :

$$T_p = C_c x R T_{pc} x \frac{dT_c}{dt} \quad \text{ec.6.10}$$

y sustituyendo en ec.(6.9) queda :

$$\frac{V^2}{R} - C_c x \frac{dT_c}{dt} = (C_r + C_p) x (C_c + R T_{pc} x \frac{dT_c}{dt} + T_c)$$

donde ponemos ,  $\tau_1 = C_c x R T_{pc}$  ; se denomina constante de tiempo del horno.

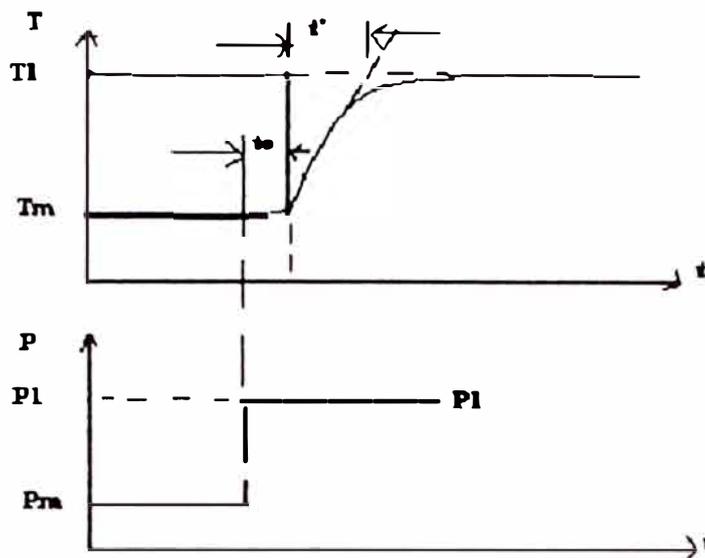
Luego dando la forma a la ecuación diferencial tenemos :

$$\tau_1 x \frac{d^2 T_c}{dt^2} + \frac{(C_c + C_r + C_p)}{C_r + C_p} x \frac{dT_c}{dt} = \frac{V^2}{R x (C_r + C_p)} \quad \text{ec.6.11}$$

(Ecuación diferencial del horno )

La solución de la ecuación diferencial nos permite obtener la curva característica del proceso .

Considerando la potencia media Pm ( temperatura límite Tm ) y aumentando dicha potencia a P1 ( temperatura límite T1 ) se puede observar la respuesta del horno :



**Figura 6.8: Constantes del horno** :  $\tau_1, t_0$

Trazando una tangente por el punto de inflexión de la curva de variación de T1 se definen dos constantes de tiempo:

$t_0$  : Tiempo muerto

$\tau_1$  : constante de tiempo

#### 6.4.3.3 Función de transferencia del horno

De la ecuación diferencial de segundo orden obtenido ec.6.11; donde Tc temperatura de la carga en el horno es aproximadamente igual a la temperatura del horno.

$$\tau \frac{d^2 T_c}{dt^2} + \frac{(C_c + C_r + C_p)}{(C_r + C_p)} \frac{dT_c}{dt} = \frac{V^2}{R(C_r + C_p)} ; \tau = \tau_1$$

La ecuación se puede escribir de otra forma:

$$\tau \frac{d^2 T_c}{dt^2} + \left( \frac{C_c}{C_r + C_p} + 1 \right) \frac{dT_c}{dt} = \frac{V^2}{R(C_r + C_p)}$$

haciendo :

$$K = \left( \frac{C_c}{C_r + C_p} + 1 \right)$$

$$u(t) = \frac{V^2}{R(C_r + C_p)} ; R = \text{resistencia eléctrica}$$

luego la ecuación se puede escribir :

$$\tau \frac{d^2 T_c(t)}{dt^2} + K \frac{dT_c(t)}{dt} = u(t) \quad \text{ec.6.12}$$

Con el fin de encontrar la función de transferencia para el horno se requiere la transformada de Laplace de la ec.6.12, utilizando tabla de transformaciones de Laplace de funciones comunes de tiempo  $f(t)$  se tiene.

$$\tau [s^2 T_c(s) - sT_c(0^-) - T_c'(0^-)] + K[sT_c(s) - T_c(0^-)] = U(s) \quad \text{ec.6.13}$$

Considerando:

$$T_c'(0^-) = T_c(0^-) = 0$$

La ec.6.13 se escribe

$$\tau s^2 T_c(s) + KsT_c(s) = U(s)$$

La función de transferencia del horno se puede expresar como:

$$\frac{T_c(s)}{U(s)} = \frac{1}{s(\tau s + K)} \quad \text{ec.6.14}$$

$$T_c(s) = T_h(s)$$

La señal actuante  $U(s)$  se puede expresar de la siguiente forma

$$U(s) = \frac{P(s)}{(C_r + C_p)} = P(s)K_1$$

luego la expresión ec.6.14 se puede expresar como :

$$\frac{T_c(s)}{P(s)} = \frac{K_1}{s(\tau s + K)} \quad \text{ec.6.15}$$

función de transferencia del horno

### Representación en variables de estado del horno de resistencias

El método de representar el horno en variables de estado, no solo comprende a las características de entrada - salida del horno sino también su comportamiento interno global.

De la ec.6.15, el empleo de la formulación de matriz - vector permite la representación en variables de estado en términos de dos ecuaciones:

$$X'(t) = A.X(t) + B.u(t)$$

$$Y(t) = C.X(t)$$

donde :

$X(t)$ : Vector de estado para la planta ( vector de 2x1 )

$P(t)$ : Señal de control ( escalar )

$A$  : Matriz constante de 2x2

$B$  : Matriz constante de 2x1

$C$  : Matriz constante de 1x2

$$\begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & -\frac{K}{\tau} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_1}{\tau} \end{bmatrix} p(t) \quad \text{ec 6.16}$$

(Representación del horno en forma de variable de estados)

## 6.5 Determinación de los parámetros del Horno

Se considera los métodos siguientes

### 6.5.1 Método teórico

Para obtener los parámetros del horno en forma teórico, requiere el uso de tablas de los componentes de construcción y el análisis teórico correspondiente, esto en algunos casos es una forma aproximada, debido a

que el presente trabajo está orientado para diferentes tipos de hornos de resistencias es conveniente utilizar el método experimental para obtener los parámetros correspondientes.

### 6.5.2 Método experimental

Hay varias razones para obtener un modelo de un sistema dinámico para controlares con datos experimentales

En primer lugar el mejor de los modelos teóricos construido a partir de ecuaciones es solo una aproximación de la realidad

Los datos de repuesta transitoria son rápidos y relativamente fácil de obtener. Son también representativos de las señales naturales a las cuales está expuesto el sistema y así un modelo basado sobre estos datos puede ser la base confiable para el diseño del sistema de control.

Experimentalmente se determino que existen cuatro intervalos de temperatura para los cuales las velocidades de calentamiento y enfriamiento son similares, esta información se resumen en la tabla siguiente:

<b>Rango de temperatura (°C)</b>	<b>Velocidad de calentamiento (°C /minuto)</b>	<b>velocidad de enfriamiento (°C/minuto)</b>
0 - 350	25	4
350 - 450	15	8
450 - 650	10	11
650 - 800	3	17

Tabla 1

Otra característica importante es la "inercia" térmica, que hace referencia al hecho de que la temperatura del horno continua aumentando luego de haber

cesado de aplicar tensión a la resistencia de calentamiento, originándose un sobreimpulso de temperatura, experimentalmente se determinaron los resultados que se presentan a continuación:

<b>Temperatura (°C)</b>	<b>Sobreimpulso de T (°C)</b>
200	33
300	24
400	18
500	9
600	6
700	1
800	1

Tabla 2

Los datos obtenidos para un horno determinado como prueba se tiene en la tabla Nro. 01 el cual, muestra la evolución de la temperatura en función del tiempo.

La respuesta del horno para una entrada escalón se tiene una curva denominada **curva de reacción del proceso**

Dicha **curva de respuesta** tiene la forma de una S. Según Ziegler y Nichols la forma de la curva es característica de los sistema de orden superior y el comportamiento entrada - salida de la planta se puede aproximar por:

$$G_p(s) = \frac{K}{(\tau s + 1)} e^{-\tau_1 s}$$

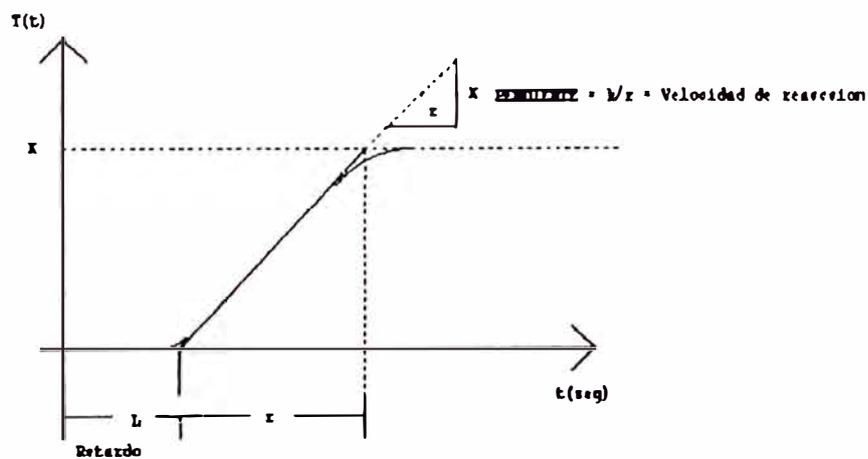
Es simplemente un sistema de primer orden más un retraso de transporte. Las constantes de la ecuación anterior se determinan a partir de la respuesta escalón.

Si se traza una tangente en el punto de inflexión de la curva de reacción, entonces la pendiente de la línea es aproximadamente

$$PENDIENTE = P = \frac{K}{\tau}$$

Y la intersección de la tangente con el eje de tiempo identifica el retraso de tiempo  $L = \tau_0$  ( tiempo muerto )

Luego de los datos se tiene :



**Figura 6.8.1: Curva de reacción del proceso**

$K_1$  ;  $K$  ; velocidad de reacción del proceso

$\tau_0$  ;  $\tau$  ; Retardo del proceso

Las tablas nro.01 y nro02 nos permite obtener estos parámetros para cada horno estudiado en particular

Estos parámetros es importante para el diseño y ajuste del sistema de control del controlador

## 6.6 Característica dinámica del sensor

La manera en que el sensor responde a cambios súbitos de entrada se conoce como características dinámicas del sensor y éstas se resumen por conveniencia utilizando una función de transferencia  $G(s)$ .

### 6.6.1 Función de transferencia del sensor

Para medir la variación de temperatura del horno, se utilizará como sensor de temperatura un termopar.

La transferencia de calor se efectúa como resultado de uno o más de tres tipos posibles de mecanismo: **Convección, conducción y radiación.**

La conducción es el mecanismo principal de transferencia de calor dentro de los cuerpos sólidos; en esta situación, el mecanismo principal de transferencia de calor es la convección. En la convección el calor se transfiere hacia el sensor

### 6.6.2 Sensor de temperatura contenido en una termofunda

Un sensor de temperatura, como el termopar por lo general está contenido en una termofunda para darle protección química y mecánica. La figura 6.9.a muestra una instalación típica de un termopar y la Figura 6.9.b es un modelo simplificado en el que el sensor y una termofunda representan las masas agrupadas individuales  $M_s$  y  $M_f$  respectivamente.

La fig.6.9.b es un modelo simplificado en el que un sensor y una termofunda representan las masas agrupadas individuales  $M_s$  y  $M_f$ , respectivamente.

Si se ignoran la capacidad térmica del espacio entre el sensor y la termofunda, las ecuaciones de equilibrio térmico son:

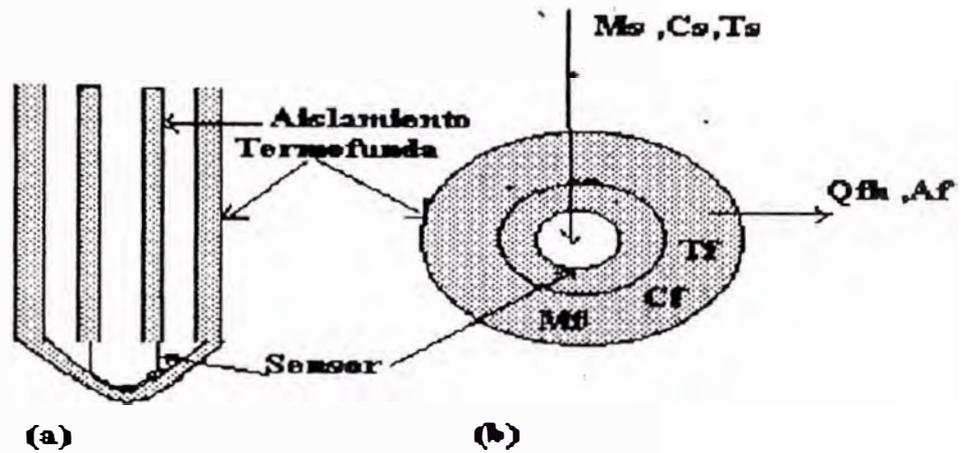


Figura.6.9 Sensor en la termofunda

$$\text{sensor: } M_s C_s \frac{dT_s}{dt} = U_{sf} A_s (T_f - T_s) \quad \text{ec 6.17}$$

$$\text{Termofunda: } M_f C_f \frac{dT_f}{dt} = -U_{sf} A_s (T_f - T_s) + U_{fh} A_f (T_h - T_f) \quad \text{ec 6.18}$$

Los cuales se transforma en

$$\tau_1 \frac{dT_s}{dt} = (T_f - T_s) \quad \text{ec.6.19}$$

$$\tau_2 \frac{dT_f}{dt} = -\delta(T_f - T_s) + (T_h - T_f)$$

$$\text{donde: } \tau_1 = \frac{M_s C_s}{U_{sf} A_s}, \tau_2 = \frac{M_f C_f}{U_{fh} A_f}, \delta = \frac{U_{sf} A_s}{U_{fh} A_f} \quad \text{ec.6.20}$$

$A_s, A_f$  = áreas de transferencia de calor entre el sensor y la termofunda

$C_s, C_f$  = Calores específicos entre el sensor y la termofunda

$U_{SP}$  = Coeficiente de transferencia de calor entre el sensor y la termofunda

$U_{h}$  = Coeficiente de transferencia de calor entre el horno y la termofunda

Definiendo  $\Delta T_h, \Delta T_f, \Delta T_s$  como desviaciones respecto de las condiciones estables iniciales, las transformaciones de Laplace de las ecuaciones ec.6.19 son:

$$[1 + \tau_1 s] \Delta T_s = \Delta T_f \quad \text{ec.6.21}$$

$$[(1 + \delta) + \tau_2 s] \Delta T_f = \Delta T_h + \delta \Delta T_s \quad \text{ec.6.22}$$

Al eliminar  $T_f$  entre estas ecuaciones, se obtiene la función de transferencia Integral.

$$\frac{\Delta T_s}{\Delta T_h}(s) = \frac{1}{\tau_1 \tau_2 s^2 + (\tau_1 + \tau_2 + \delta \tau_1) s + 1} \quad \text{ec.6.23}$$

La ecuación ec.6.23 es un modelo de segundo orden, donde la punta del elemento sensor no toca la termofunda

En una instalación normal correcta con la punta del sensor haciendo contacto con la termofunda, tiene un valor de  $u_f$  mucho mayor y de  $\tau_1$  mucho menor. En el límite de la transferencia perfecta de calor entre el sensor y la termofunda, ambos elementos están a la misma temperatura  $T_s$  y la ecuación de equilibrio térmico es ahora:

$$(M_s C_s + M_f C_f) \frac{dT_s}{dt} = u_{kf} A_f (T_k - T_s) \quad \text{ec.6.24}$$

Lo que resulta en una función de transferencia de primer orden

$$\frac{\Delta T_s}{\Delta T_k}(s) = \frac{1}{1 + \tau s}, \tau = \frac{M_s C_s + M_f C_f}{u_{kf} A_f} \quad \text{ec.6.25}$$

la ecuación función de transferencia del sensor :

$$\frac{dT_s(s)}{dT_k(s)} = \frac{1}{1 + \tau s} \quad \text{ec.6.25.1}$$

donde :

$\tau$  constante de tiempo del sensor

Este modelo es una buena representación de instalaciones donde se ha obtenido medidas especiales para mejorar la transferencia de calor .

Para obtener una respuesta rápida y el sensor debe estar protegido, entonces debe utilizarse un termopar aislado con minerales

El termopar consta de un alambre delgado dentro de un tubo estrecho de paredes delgadas: el tubo se llena con materiales mineral que sea un buen conductor de calor, pero aislante eléctrico. En este caso  $M_s$  y  $M_f$  son pequeños, lo cual resulta en constantes 100 veces menores que en las instalaciones de termofundas.

Se tiene:

los valores de la constante del sensor:

$\tau = 1.0$  ( sensor con termofunda)

$\tau = 0.7$  ( sensor con aislado con minerales)

## **6.7 Diseño del algoritmo del controlador**

### **6.7.1 Especificaciones de funcionamiento**

El presente trabajo, se realiza el diseño de un sistema digital que efectuará una regulación precisa de la temperatura en hornos eléctricos para el tratamiento térmico.

#### **6.7.1.1 Control programado**

Para realizar el control programable se realizará mediante la PC, el ingreso de datos como:

- Parámetros del controlador PID (  $K$  ,  $T_I$  ,  $T_d$  ,  $T$  ), de acuerdo al tipo de horno a controlar
- Control: Set - Point; temperatura de referencia (grados centígrados), tiempo en horas y minutos.
- Ingreso de clave para no variar los parámetros del controlador

Verificación de los datos Ingresados, Mostrar los parámetros y control

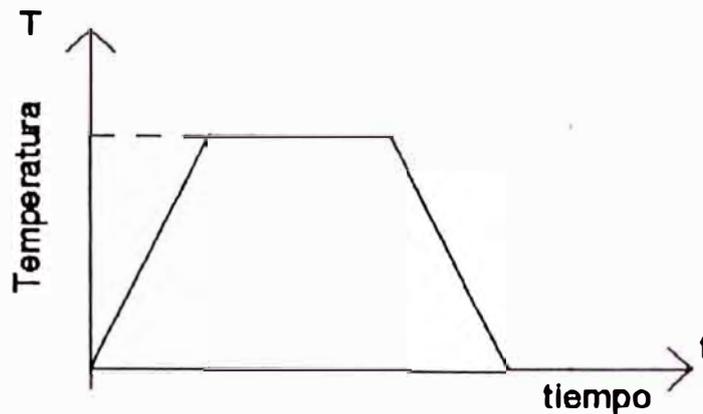
Enviar datos al controlador vía puerto serial.

El diseño del sistema se enfoca en orden a conseguir las siguientes especificaciones (prestaciones).

1.- Obtención de una secuencia de tramos las cuales serán

- Subidas lineales de la temperatura de consigna durante un tiempo prefijado.
- Estabilizaciones de temperatura de consigna durante un tiempo prefijado.
- Bajadas de la temperatura de consigna hacia una cota cualquiera

Ver Figura 6.10



**Figura 6.10 Secuencia térmica generada**

**2.- Rango de temperatura:**

Las evoluciones del punto de consigna serán desde de la temperatura ambiente hasta 1100 °C.

**3.- Error máximo de las regulaciones en cada tramo de estabilización será de 5° C. ( en el control )**

**4.- Precisión en las medidas de tiempo.**

**5.- Resolución en las medidas de tiempo : 1seg.**

**6.- Reducción al máximo el rebosamiento en las terminaciones de las rampas de las temperaturas con una determinada pendiente.**

**7.- Mantenimiento constante de las velocidades de ascenso de la temperatura del horno en los tramos lineales, según lo especificado por el programa.**

**8.- Presentación continua en la pantalla durante la fase de control, del tiempo programado y evolución de la temperatura.**

**9.- El sistema debe autovigilarse su funcionamiento en doble sentido:**

**a) Comprobando permanente que no se haya producido modificación alguna en el programa almacenado de usuario a causa de un fallo momentáneo de alimentación o ruido.**

- b) Vigilando en forma continua que la temperatura del horno no rebase una cuota máxima admisible establecida , como consecuencia de una avería en alguna de sus partes del sistema.
- c) En ambos casos anteriores el sistema bloqueará la etapa de potencia dando por finalizado el proceso térmico

### **6.7.2 Selección del tipo de control**

De acuerdo a las especificaciones establecido ; en el sistema de control se utilizará la técnica digital para la regulación se puede resumir en tres funciones generales :

- 1.- ' Logging ": Recogida, tratamiento, almacenamiento y presentación de datos.
- 2.- Supervisión : Asistencia o guía al operador para que ejecute las acciones de control.
- 3.- Control : Cálculos de las acciones de control que deben ejecutarse tipo PID .

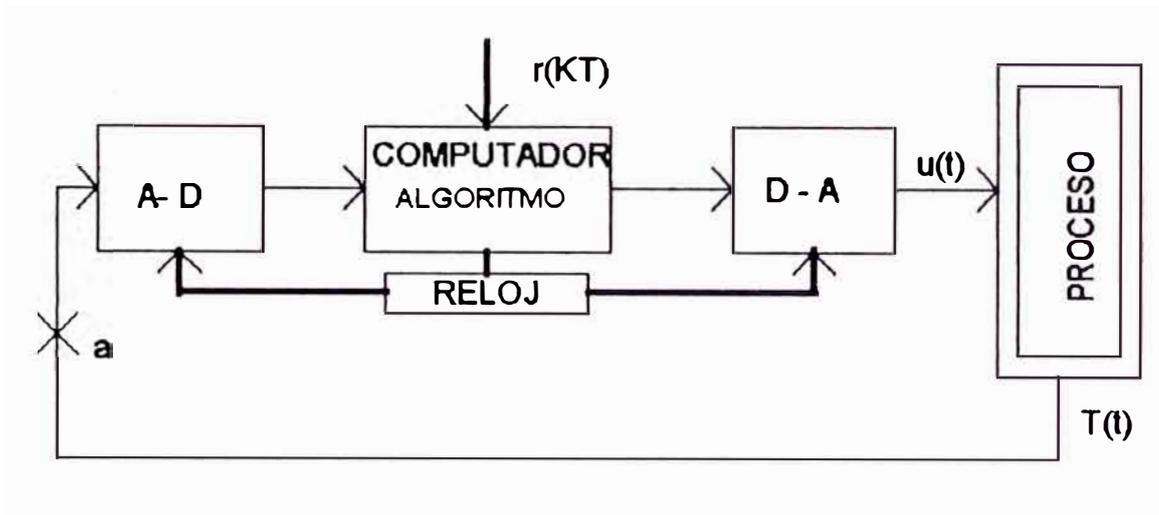
Considerando el tercer punto podemos indicar que se utilizará el control clásico ' Control Digital Directo' ( DDC ).

Con el objeto de poder efectuar una regulación precisa durante los tramos de subidas, tramos de estabilización y bajadas, se optado por un control de tres términos: PID ( Proporcional - Integral - Derivativo ).

El DDC se ha definido como un sistema en el cual la acción reguladora es determinado por un Microcomputador 8051 - Intel. Es decir el cálculo ejecutado por el clásico regulador (PID) es realizado por el Microcontrolador : El regulador como tal desaparece.

### 6.7..2.1 Esquema del bucle de regulación

En la figura 6.11 se muestra un diagrama de bloques del sistema de control digital ha considerar; que presenta la configuración del esquema de control básico.



**Figura 6.11 Diagrama de bloques del sistema de control digital**

donde:

$r(KT)$  = comando de entrada o referencia

$T(t)$  = Cantidad de salida ó variable controlada

$u(t)$  = Señal de entrada del actuador

Las variables asociadas al proceso evoluciona de forma continua en el tiempo, luego debido a que el sensor y actuador son analógicos es necesario utilizar el ADC para suministrar la señal al controlador digital y DAC para obtener una señal continua para el actuador. La actuación del Microcontrolador debe estar sincronizado con el proceso. Para ello es necesario el empleo de un reloj en tiempo real. Típicamente el convertidor ADC suministra un número al Microcomputador cada vez que le llega un pulso del reloj.

El algoritmo del computador es el siguiente:

1. Esperar hasta que ocurra un pulso de reloj
2. Realizar una conversión analógica a digital
3. Calcular las variables de control.
4. Realizar la conversión digital a analógica
5. Actualizar el estado del regulador
6. Volver de nuevo al punto 1

### 6.7.2.2 Algoritmo de control

Con la finalidad de realizar una precisa regulación del sistema de control para cumplir con los requisitos mencionados se optó por algoritmo PID, el algoritmo es de tipo **Incremental** se describe a continuación.

Una vez verificada la ausencia de alarmas se tiene. La señal del controlador:

$$\text{ACTUA} = \text{ACTUA} + \Delta \text{ACTUA}$$

### 6.7.2.3 Algoritmo PID

La ecuación diferencial de un regulador PID puede ser escrito mediante la siguiente ecuación

$$(\Delta \text{PID}) = \Delta \left[ K_p + K_i \left( \varepsilon + \frac{1}{T_i} \int \varepsilon dt + T_d \frac{d\varepsilon}{dt} \right) \right] \quad \text{ec. 6.26}$$

Siendo:

$$K_p, K_i = \text{ctes}$$

$$\varepsilon = - \left( T_{RMR} - T_{MBCIDA} \right) \frac{1}{T_{RMR}}$$

$\varepsilon$  : Error relativo

$T_i$  = Tiempo de acción integral

$T_D$  = Tiempo de acción derivativo

La ecuación 6.26 escrita en tiempo discreto tenemos

$$(\Delta PID) = K_1 \epsilon_N + K_1 \frac{T}{T_N} \sum \epsilon_i + K_1 \frac{T_D}{T} (\epsilon_N - \epsilon_{N-1}) \quad \text{ec.6.27}$$

Aplicando su aproximación por diferencias finitas se tiene ; La integral se ha aproxima por la suma de rectángulos y la derivada por la ecuación diferencia , si definimos :

$$(\Delta PID)I_N = K_1 \frac{T}{T_i} \sum \epsilon_i \quad \text{ec.6.28}$$

$$(\Delta PID)I_N = (\Delta PID)I_{N-1} + K_1 \frac{T}{T_i} \epsilon_N \quad \text{ec.6.29}$$

$$(\Delta PID)I_{N-1} = (\Delta PID)I_N - K_1 \frac{T}{T_i} \epsilon_N \quad \text{ec.6.30}$$

$$(\Delta PID)_{N-1} = K_1 \epsilon_{N-1} + (\Delta PID)I_{N-1} + K_1 \frac{T_D}{T} (\epsilon_{N-1} - \epsilon_{N-2}) \quad \text{ec.6.31}$$

Remplazando ec.6.29 en ec.6.30

$$(\Delta PID)_{N-1} = K_1 \epsilon_{N-1} + (\Delta PID)I_N - K_1 \frac{T}{T_i} \epsilon_N + K_1 \frac{T_D}{T} (\epsilon_{N-1} - \epsilon_{N-2}) \quad \text{ec.6.32}$$

De ec.6.27 y ec.6.28

$$(\Delta PID)_N = K_1 \epsilon_N + (\Delta PID)I_N + K_1 \frac{T_D}{T} (\epsilon_N - \epsilon_{N-1}) \quad \text{ec.6.33}$$

De ec.6.30

$$(\Delta PID)I_N = (\Delta PID)_{N-1} - K_1 \epsilon_{N-1} - K_1 \frac{T}{T_i} \epsilon_N - K_1 \frac{T_D}{T} (\epsilon_{N-1} - \epsilon_{N-2}) \quad \text{ec.6.34}$$

De ec.6.33 y 6.34 tenemos

$$(\Delta PID)_N = K_1 \varepsilon_N + (\Delta PID)_{N-1} - K_1 \varepsilon_{N-1} + K_1 \frac{T}{T_i} \varepsilon_N - K_1 \frac{T_D}{T} (\varepsilon_{N-1} - \varepsilon_{N-2}) + K_1 \frac{T_D}{T} (\varepsilon_N - \varepsilon_{N-1})$$

ec.6.35

luego se puede expresar :

$$(\Delta PID)_N = (\Delta PID)_{N-1} + K_1 \left[ 1 + \frac{T}{T_i} + \frac{T_D}{T} \right] \varepsilon_N + (-1 - 2 \frac{T_D}{T}) \varepsilon_{N-1} + \frac{T_D}{T} \varepsilon_{N-2} \quad \text{ec.6.36}$$

$$(\Delta PID)_N = (\Delta PID)_{N-1} + A \varepsilon_N + B \varepsilon_{N-1} + C \varepsilon_{N-2} \quad \text{ec.6.37}$$

donde:

$$\Delta ACTUA_N = \Delta PID_N$$

Es un algoritmo de control que expresa el valor enésimo de la variable manipulada en términos del valor anterior (n-1) y valores presentes y pasados del error.

Siendo :

"N" el número de muestreo considerado y A,B,C constantes dependientes de:

$$A = K_1 \left[ 1 + \frac{T}{T_i} + \frac{T_D}{T} \right]$$

$$B = K_1 \left[ -1 - 2 - \frac{T_D}{T} \right]$$

$$C = K_1 \frac{T_D}{T}$$

donde :

$K_1$ : constante del regulador

$T_D$ : Tiempo de acción derivativo

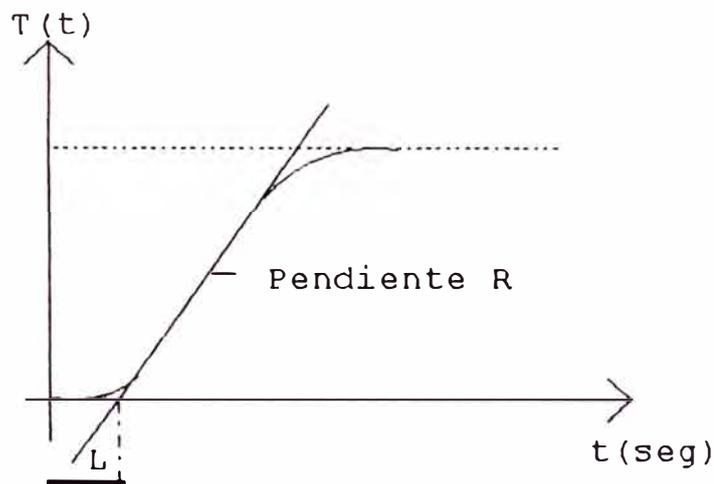
$T_i$ : Tiempo de acción integral

$T$  : Tiempo de muestreo

#### 6.7.2.4 Selección de los parámetros del controlador

Los controladores PID discretos tienen la ventaja que se comportan como controladores PID continuos cuando el periodo de muestreo es corto, para esto se puede considerar el método de la respuesta transitoria, se mide la máxima pendiente,  $R$  y el retardo,  $L$ , de la respuesta en lazo abierto ante una entrada escalón unitario.

Véase la figura 6.12



**Figura 6.12 Medida de las variables  $R$  y  $L$  sobre la respuesta en lazo abierto con el método de la respuesta transitoria**

Los parámetros para el controlador PID se obtienen con la ayuda de la tabla nro.03

	$K$	$T_i$	$T_D$
P	$1/RL$		
PI	$0.9/RL$	$3L$	
PID	$1.2 / RL$	$2L$	$0.5L$

**Tabla Nro.03 Parámetros del controlador cuando se emplea el método de la respuesta transitoria**

### 6.7.2.5 Selección del Intervalo de muestreo

Se recomienda que el periodo de muestreo ( T ) para las variables mas comunes, cuando se utiliza el control CDD sean:

Tipo de variable	Periodo de muestreo en seg
Flujo	1 - 3
Nivel	5 - 10
Presión	1 - 5
Temperatura	10 - 20

Se tiene también varias reglas que relacionan el intervalo de muestreo y el tiempo derivativo (constante derivativo). Es típica la sugerencia de que el periodo de muestreo se elija de modo que

$$\frac{T}{T_D} \approx 0.1 - 0.5$$

Para las reglas de sintonia de Ziegler - Nichols , esto implica que

$$\frac{T}{L} \approx 0.05 - 0.05$$

Con un experimento simple sobre el proceso, es posible comprobar si el periodo de muestreo es correcto.

## 6.8.1 Instrumentación del proceso

### 6.8.1.1 Descripción del hardware

Los bloques funcionales básicos del sistema de control consiste en la siguientes partes:

#### 1.MODULO I : Sección de supervisión

Las operaciones del sistema están supervisadas por una PC. que tiene la además la función de:

- Generación de puntos de consigna
- Clave de control de cambios de parámetros
- Ingreso de parámetros de control
- Ingreso de Set - Point de la temperatura y tiempo
- Mostrar parámetros y variable ha controlar a si mismo tiempo

## **2.MODULO II: Sección de control**

En esta sección se trata de la parte de control del proceso ,se utiliza un microcontrolador 8031 Intel como alternativa a las tarjetas microprocesador, el microcontrolador debe dar las ordenes oportunas de disparo de los tiristores que componen el sistema , de acuerdo con la referencia elegida y con la señal de Información del sensor térmico ( termocupla):

La sección de control podemos indicar lo siguiente:

- Microcontrolador 8051 Intel
- Memoria EEPROM : 2816
- Generador de reloj
- Interfase serie para comunicaciones
- Incluye etapa de Visualización

## **3.MODULO : Sección analógica**

Esta sección comprende la adquisición y acondicionamiento de las señales analógicas para su tratamiento digital Incluye las siguientes etapas.

- Sensor de temperatura
- Amplificación y acondicionamiento de la señal analógica

- Procesamiento de la señal analógica :Convertor Análogo - Digital (CAD)

#### **4.Modulo : Interfase sección de potencia**

En esta sección se describe la Interfase lógica, previa a la etapa de potencia propiamente dicha.

#### **5.Modulo: Control de potencia**

Se detalla el método utilizado en el control de potencia del horno Incluye en esta etapa:

- Salida de control del elemento actuador
- Visualización de señal de control
- Etapa de cruce por cero

#### **6.8.2 Selección del elemento sensor**

La realización de las medidas de temperatura a las que se encuentra el horno a lo largo del ciclo ,se efectúa a partir de señal proporcionada por un termopar medidor; dicha señal suele variar de forma no lineal entre cero y 10 ó 20mV dentro del margen de temperaturas que estamos considerando (25 °C a 1100°C).

Los factores que influyen en la selección del sensor son:

- Tecnología

Son elementos termoeléctricos. Si se empalman dos metales A y B diferentes, existe una diferencia de potencia eléctrica a través del empalme llamado potencial de contacto este potencial de contacto depende de los metales A y B y de la temperatura T°C del empalme y esta dado por una serie de potencias de la forma:

$$E_T^{AB} = a_1 T + a_2 T^2 + a_3 T^3 + \dots \quad \text{ec.6.38}$$

Los valores de las constantes  $a_1, a_2, a_3, \dots$  etc. dependen de los metales A y B

- Comportamiento Funcional :

Un termopar es un circuito cerrado que consta de dos empalmes a diferentes temperaturas  $T_1^\circ C..y...T_2^\circ C$ . Si se introduce en el circuito un voltímetro de alta impedancia, de manera que el flujo de corriente sea despreciable, entonces la F.E.M medida es aproximadamente, la diferencia de los potenciales de contacto, esto es:

$$E_{T_1, T_2}^{AB} = a_1(T_1 - T_2) + a_2(T_1^2 - T_2^2) + a_3(T_1^3 - T_2^3) + \dots \quad \text{ec.6.39}$$

por lo tanto, la F.E.M medida depende de las temperaturas  $T_1^\circ C..y...T_2^\circ C$  de ambos empalmes.

$T_1^\circ C$  : temperatura a medir

$T_2^\circ C$  : temperatura del empalme de referencia

A fin de deducir con precisión el valor el valor de  $T_1^\circ C$  a partir de la FEM. medida , debe conocerse la temperatura del empalme de referencia  $T_2^\circ C$  ver figura 6.11

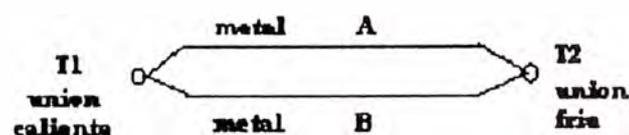


Figura 6.11 Circuito de un termopar

También se debe considerar otros factores para la elección del sensor como:

- Propiedades Físicas : Peso, tamaño, resistencia
- Factores de calidad : Fiabilidad, durabilidad y mantenimiento
- Costo : Gastos, disponibilidad, instalaciones para prueba y mantenimiento.

Finalmente en la tabla Nro. 4 resume el alcance de medición, los valores de FEM, las tolerancias y características de cuatro termopares en uso industrial común. La tabla puede considerarse para cuantificar la no linealidad

Luego finalmente se puede concluir que el termopar adecuado para nuestro sistema y amplificación se considero el de tipo " K ".

- Sensibilidad  $\frac{dE}{dT} = 41.269 \frac{\mu V}{^{\circ}C}$

- Margen de temperatura: 0 - 1250 °C

- Tolerancia : 0.75 %

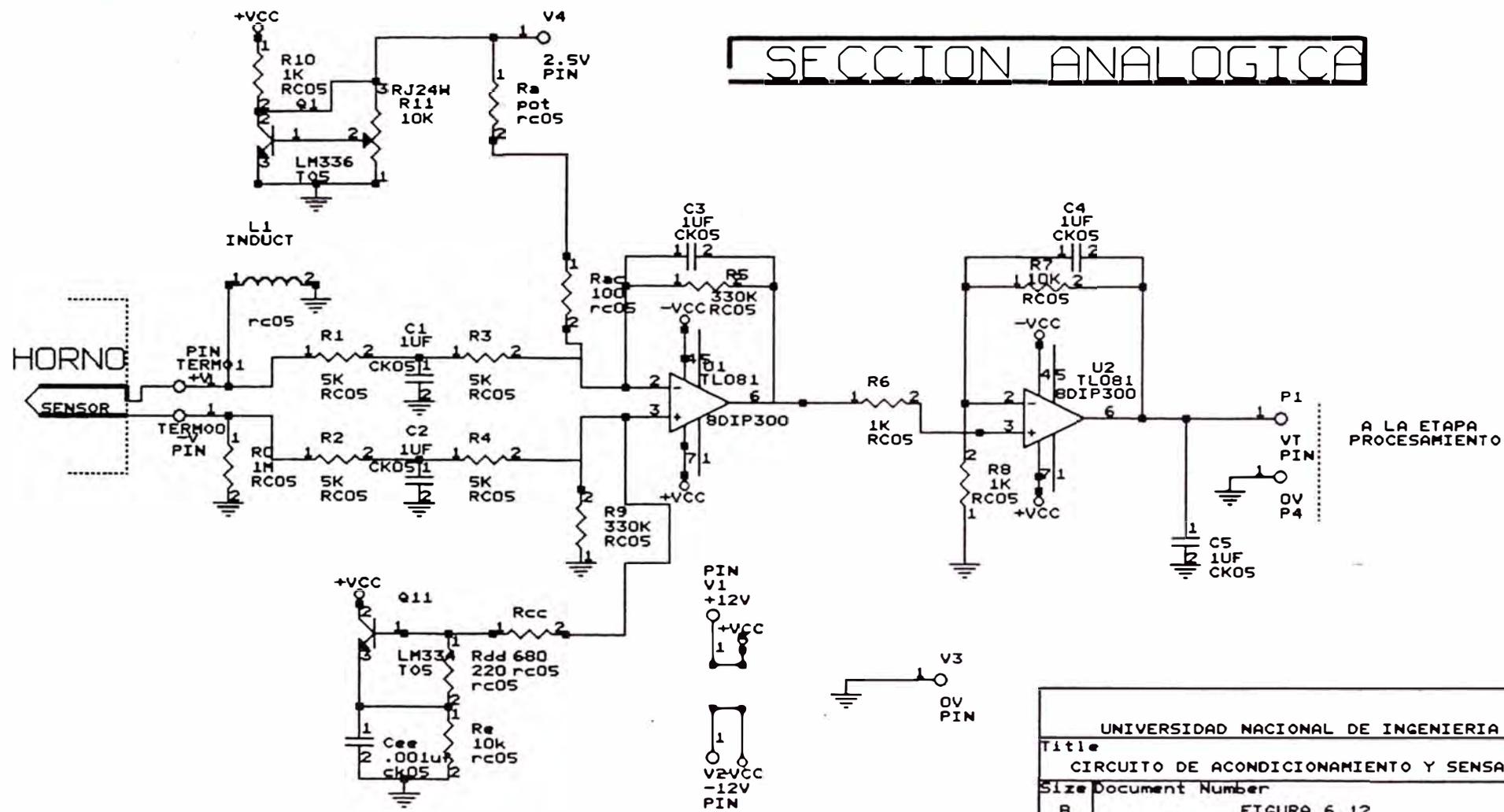
### 6.8.3. Tratamiento de la señal analógica

#### 6.8.3.1 Filtro de entrada

La termocupla se conecta mediante un filtro de entrada pasa bajo con frecuencia de corte sobre 50 Hz , para eliminar en lo posible el ruido inducido de alta frecuencia . figura.6.12

Se observa que se ha fraccionado una resistencia R en R/2 y luego se ha colocado un condensador en el punto medio, como si se tratase de típico filtro tipo " T ".

# SECCION ANALOGICA



A LA ETAPA PROCESAMIENTO

UNIVERSIDAD NACIONAL DE INGENIERIA		
Title CIRCUITO DE ACONDICIONAMIENTO Y SENSADO		
Size Document Number B	FIGURA 6.12	REV
Date: December 28, 1999 Sheet		of 30

### 6.8.3.2 Amplificador

Como la tensión suministrada por el termopar utilizado hasta temperaturas de  $1250^{\circ}\text{C}$  (cromel - alumel) está entre 0 y 41.269 mV, se hace preciso un amplificador previo de ganancia próxima a 31.5.

El amplificador debe cumplir los siguientes requisitos:

- Baja deriva térmica
- Bajo " deriva offset ".

Considerando los requisitos es preciso efectuar el diseño bajo condiciones desfavorables ello ha conducido a utilizar un amplificador de , precisión . se ha seleccionado TL081 cuya especificación es.

- Deriva térmica es de  $3\mu\text{V}/^{\circ}\text{C}$
- Offset de entrada de 0.7mV

#### Configuración Adoptada:

Se ha considerado el OPAM como amplificador diferencial, ver FIG.6.12

- El conjunto  $R/2$  C  $R/2$  constituyen un filtro pasa bajo sobre 50 Hz
- Mientras que las resistencias partidas  $R/2$  ,  $R/2$  y  $33R$  establece la ganancia del amplificador.

### 6.8.3.3 Compensación

Para evitar que se formen nuevos termopares, se utiliza el llamado cable de compensación que esta formado por materiales iguales a los de la termocupla. Con ello el equipo ve una tensión proporcional a la diferencia de temperatura entre el punto objeto de medición y la temperatura ambiente del equipo; esta ultima se denomina dentro del argot " soldadura fría ". La

temperatura del equipo es variable según el ambiente en consecuencia la temperatura de la soldadura fría, al no ser constante, es una fuente de error considerable; existen varios sistemas de compensación. Una solución alternativa que emplea la ley de temperatura intermedias se muestra en la fig.6.13. donde la FEM. del termopar  $E_{T_1,T_2}$ , para una temperatura de empalme de medición  $T_1^{\circ}C$  y una temperatura de referencia  $T_2^{\circ}C$ . Si se introduce una segunda fuente de F.E.M de magnitud  $E_{T_2,0}$  en el circuito en serie con  $E_{T_1,T_2}$ , entonces el voltímetro mide  $E_{T_1,T_2} + E_{T_2,0}$ , que es igual  $E_{T_1,0}$ , por lo tanto el voltímetro mide una F.E.M respecto de una aparente temperatura fija de  $0^{\circ}C$  aunque la temperatura de referencia real varíe respecto de una medida de  $20^{\circ}C$ . La fuente de F.E.M que produce  $E_{T_2,0}$  se conoce como circuito automático compensatorio del empalme de referencia. de la ec.6.38 se tiene

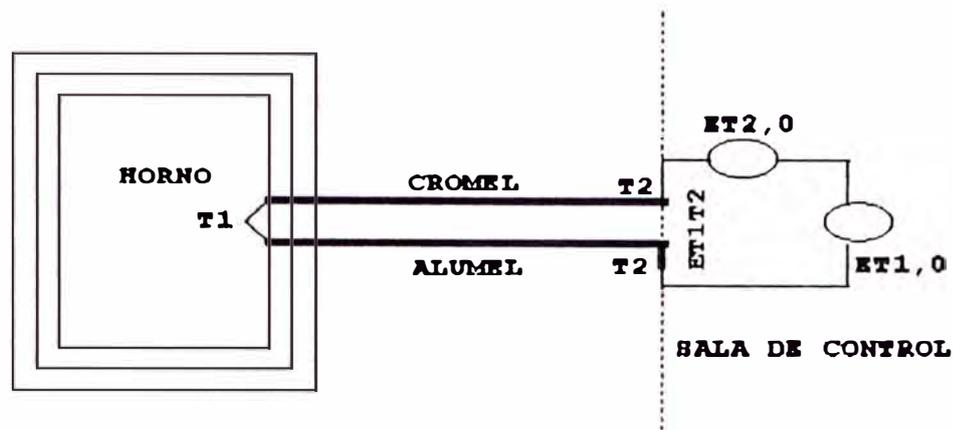
$$E_{T_2,0} = a_1 T_2 + a_2 T_2^2 + a_3 T_2^3 + \dots \quad \text{ec6.40}$$

Pero como  $T_2^{\circ}C$  es pequeña, puede obtenerse una solución aproximada a través de  $E_{T_2,0} \approx a_1 T_2$ . Por lo tanto se requiere un circuito que genere una señal de salida en milivolts proporcional a la temperatura de referencia  $T_2$ .

Donde :

$E_{T_1,T_2}$  : FEM. del termopar

$T_1^{\circ}C$  : Temperatura de empalme de medición



**Fig.6.13 Instalación termopar**

$T_2^{\circ}C$  : Temperatura de referencia (alrededor de 20 a 25°C)

$E_{T1,0} = E_{T1,T2} + E_{T2,0}$  : Lectura del voltímetro

0°C : Temperatura aparente de referencia fija

$E_{T2,0}$  : Fuente automática compensación del empalme de referencia

**Consideración Práctica :**

Por lo tanto , se requiere un circuito que genere una señal de salida en mili - voltios proporcional a la temperatura de referencia  $T_2$ . El voltaje de compensación de la temperatura ambiente es implementada con el LM334 , que es un generador de corriente en función de la temperatura se usa un potenciómetro para calibrar el voltaje a 25°C. A si cualquier cambio en  $T_2$  que altere la FEM del termopar ,es susceptible a la sensibilidad del LM334 , produciendo un cambio compensatorio en el voltaje de salida del

circuito compensación

### Temperatura de referencia 0°C

La temperatura de referencia de 0°C ,esta en la sala de control o en el equipo, una solución clara a este problema consiste en colocar una referencia en un ambiente a temperatura controlada como puede ser por ejemplo un refrigerador de 0°C .

Donde la referencia lo derivaremos mediante el LM336 ,una tensión regulada necesario para la referencia de 0°C.

Ver Fig. 6.12

### Voltaje medido del termopar

La salida de la etapa de amplificación y acondicionamiento de la señal cumplirá la siguiente función para tener una señal proporcional a la temperatura medida

$$V_{SAL} = G(V_T + V_{TAMB} - V_{RBF}) \quad \text{ec. 6.41}$$

donde:

$G$  : Ganancia del amplificador =  $33R/R = 33$

$V_T$  : Voltaje generado por la termocupla

$V_{RBF}$  : Es el voltaje correspondiente a 0°C

$T_{AMB}$  : Voltaje de compensación generado por el sensor de temperatura de la unión fría.

Amplificador: Configuración sumador - restador. Conexión de la termocupla:

Forma diferencial

### Implementación:

El voltaje de la temperatura de referencia para  $0^{\circ}\text{C}$  ( $V_{TRBF}$ ) se medio con una termocupla tipo 'K', y fue de 11 mV, en el circuito se deriva el voltaje referencia mediante el LM336

El voltaje de compensación de temperatura ambiente es implementada con el sensor LM 334, se calibra el voltaje para obtener 11mV a  $25^{\circ}\text{C}$

Debido que la termocupla genera  $41269 \frac{\mu\text{V}}{^{\circ}\text{C}}$ , la entrada al amplificador a una temperatura de  $1100^{\circ}\text{C}$  es.

$$V_T = 41.269 \frac{\mu\text{V}}{^{\circ}\text{C}} \times 1100^{\circ}\text{C} = 45.396 \text{ mV}$$

Salida del amplificador

$$V_{SAL} = 45.396 \times 33 = 14980.068 \text{ mV} = 1.498 \text{ Voltios}$$

### 6.8.4. Conversor analógico - digital ( A - D )

Un ADC común de bajo costo, es el ADC0804 que pertenece a una familia de convertidores que son casi iguales excepto en la exactitud este convertidor es compatible con una amplia gama de microprocesadores, existe ADC más rápido y algunos con mayor resolución que 8 bits, pero este convertidor es ideal para muchas aplicaciones que no requieren un alto grado de exactitud, debido a que en control de procesos térmicos, no se requiere un alto grado de precisión se consideró el ADC0804 dicha etapa de procesamiento se tiene el modulo entrenador. Usando la equivalencia de  $5^{\circ}\text{C}$  por bit de definición podremos medir hasta  $1275^{\circ}\text{C}$  de temperatura. ( $255 \times 5 = 1275^{\circ}\text{C}$ ).

### **6.8.5 Etapa del controlador o regulador**

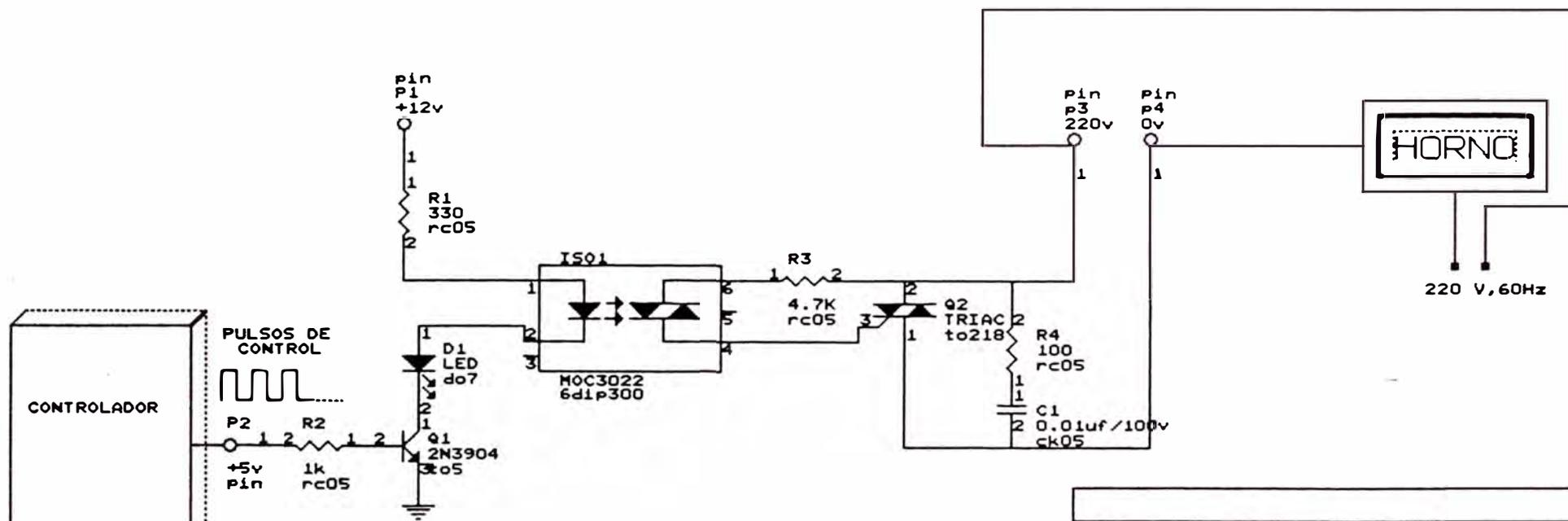
La etapa del controlador o etapa de regulación esta basado entorno al las unidades microcontroladoras , se describe el elemento base de este control de cargas de potencia y la unidad MUC y se demuestra como se puede reducir en tamaño y se puede simplificar considerablemente el diseño y realización practica de un sistema de control; en nuestro caso se tiene el modulo entrenador basado en un microcontrolador 8031 , 8051 ó 8051 INTEL elegido por reunir los bloques necesarios para realizar el control, así como por su bajo costo

### **6. 8.6 Control de potencia**

Describimos la Interfase lógica, previa a la etapa de potencia propiamente dicha . Según se detalla en el circuito de la Fig. **.6.13**

Debido a que existe el peligro de que existan fugas de corriente del circuito de potencia que puedan dañar a los circuitos de pequeña señal y de control (digitales y analógicos), se aisló la señal de control proveniente del microcontrolador, de la entrada de compuerta del triac, mediante un modelo MOC3021. Dependiendo de la resistencia calefactora que incorpore el horno que se desea controlar ,la etapa final de se realizo mediante un triac de 41 amperios y cuenta con un voltaje máximo de trabajo de 600V para un horno de resistencias de 10KW.

# INTERFACE DE POTENCIA CON TRIAC



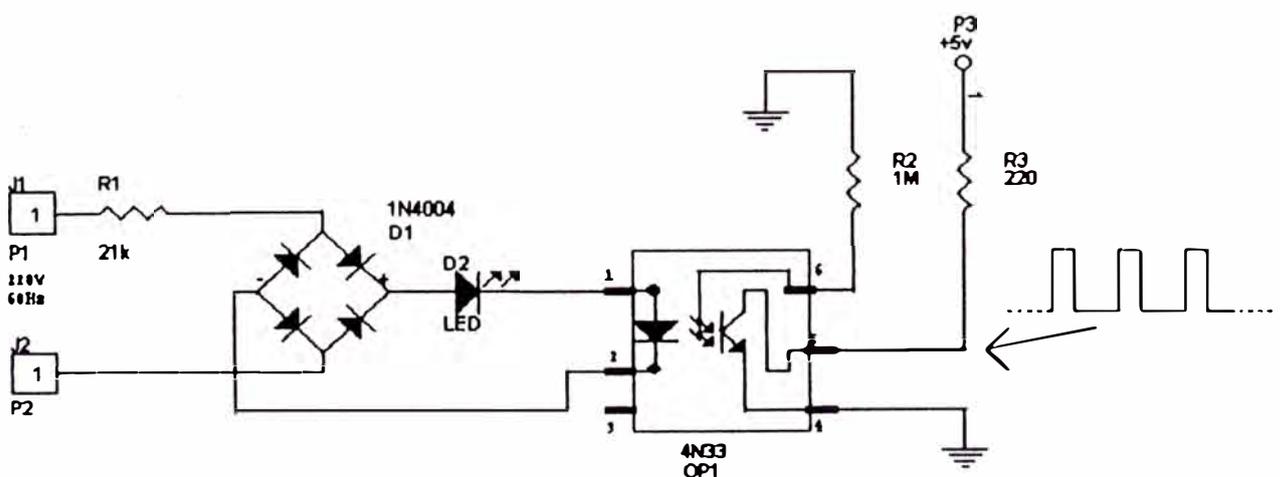
Title		ETAPA DE POTENCIA	
Size	Document Number	REV	
B	FIG.6.13		
Date:	December 28, 1999	Sheet	of 2

### 6.8.7 Circuito de cruce por cero

El circuito de cruce por cero nos permite detectar el paso por cero de la tensión de la red ,para lo cual se ha previsto un opto - acoplador con los diodos rectificador, en serie con la resistencia (el valor de esta resistencia determina la anchura del pulso). Solamente cuando la tensión valga cero, la salida del opto - Transistor pasará a un estado de no conducción, dejando que el colector suba a 5 voltios .

Conociendo el instante en que la red pasa por cero bastará con temporizar el tiempo del disparo sobre el triac, para obtener la regulación deseada .

La tarea de detectar los pasos por cero de la red está reservado a la interrupción externa INT0 encargado de detectar los pasos por cero de la red. Ver figura.6.14



**Fig.6.14 Detector de paso por cero**

## **6.9 Software del sistema de control automático para hornos de tratamiento térmico**

El sistema de control desarrollado para realizar las diferentes acciones se consideró básicamente dos programas en lenguaje assembler para uP8088 y el assembler para uC8051.

- Programa para el usuario, así mismo para Ingresar parámetros y variables , etc. mediante la PC; vía comunicación serial, conectado con el módulo de prácticas para sistemas de control automáticos
- Programa para el algoritmo de control de temperatura y tiempo, interrupciones , visualización , etc.

### **6.9.1 Estructura del software de control del proceso térmico**

Este programa consta de varias subrutinas

- Medición de temperatura
- Selección de parámetros
- Subrutina de visualización de temperatura y tiempo
- Subrutina de regulación
- Subrutina de comunicación serial con la PC y interrupciones

Ver diagrama de flujo N. 1

### **6.9.2 Estructura del software para supervisión del sistema**

Este Software permite presentar en pantalla de la PC, un menú para que el usuario puede hacer Ingresar los parámetros considerados y evaluados para cada horno ha utilizar , y las variables a controlar como son:

Parámetros del controlador PID:

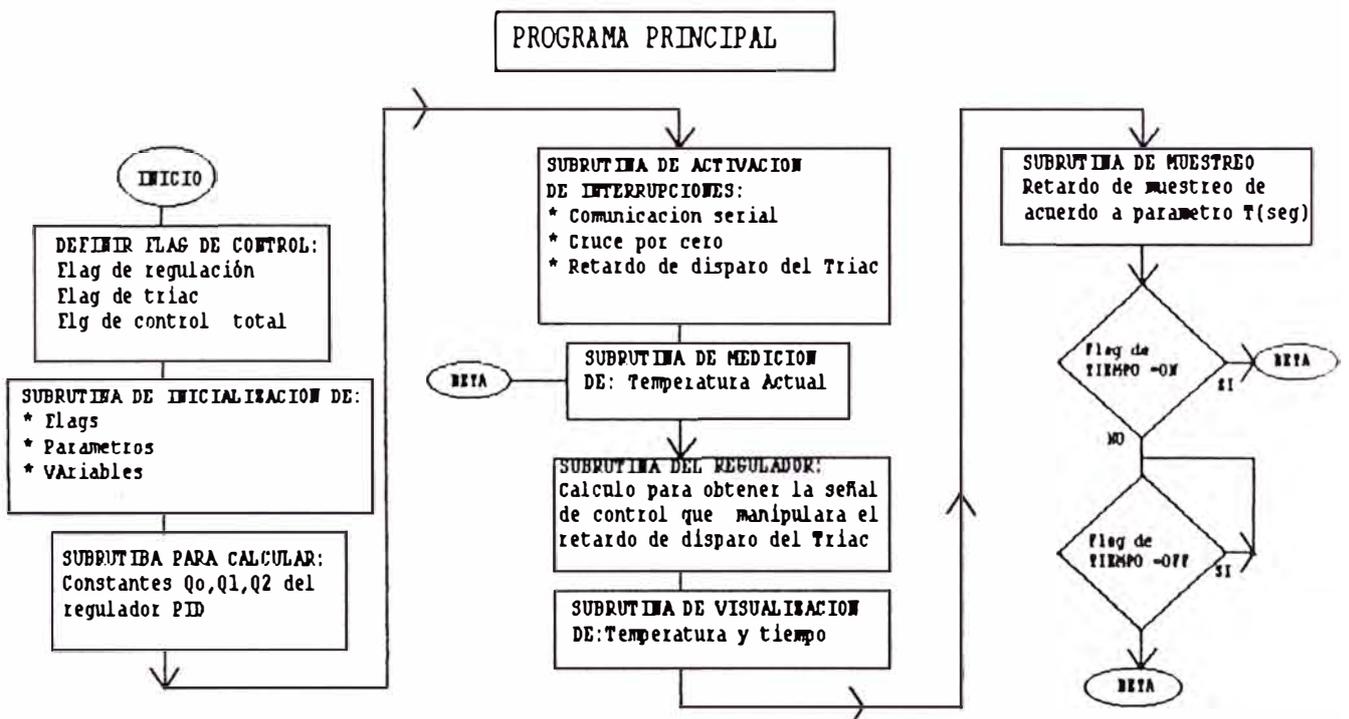
- Constante proporcional

- Constante tiempo integral
- Constante de tiempo derivativo
- Tiempo de muestreo
- Set point
- Tiempo

El menú así mismo presenta una clave para guardar los valores ingresados esta clave de acceso para realizar cualquier cambio se denomina "UNI".

Finalmente este programa nos permite realizar la comunicación de la PC para enviar los valores Ingresados, a la memoria RAM del entrenador de Control automático, para posteriormente comenzar ha realizar el proceso correspondiente. Ver diagrama de flujo N. 2

### Diagrama de flujo nro01: Programa de control del proceso térmico



**Subrutina del proceso ( Regulador PID) ,** nos permite obtener la señal  $u(k)$  para generar el retardo de disparo del triac.; una vez obtenido el valor  $u(K)$  , esto da lugar a que se obtenga valores de carga de los registros de conteo del Timer 0 como temporizador , luego generar la interrupción del Timer 0 y producir el disparo del triac

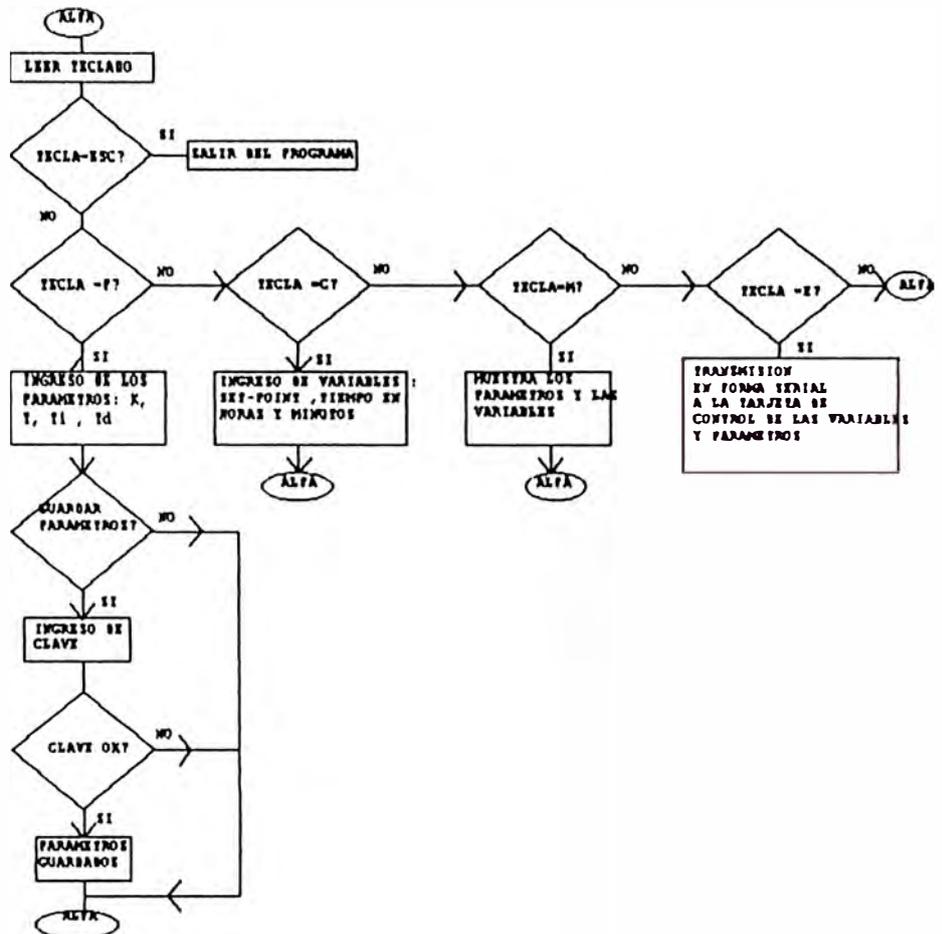
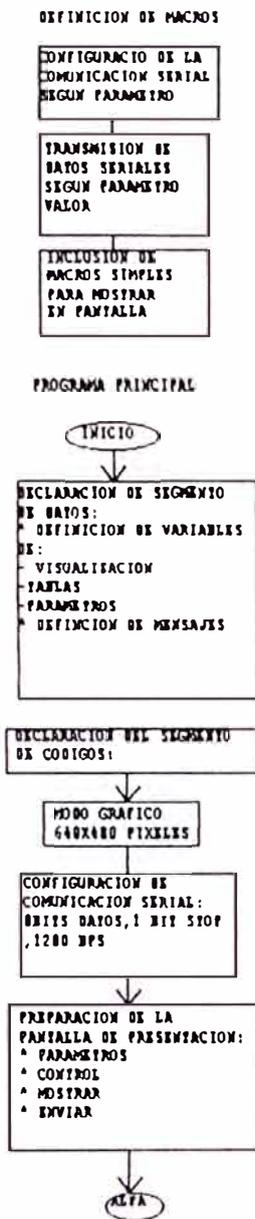
**Subrutina de comunicación serial** Esta subrutina recibe los parametros  $T, T_d, T_i, K$  , luego el tiempo en horas y minutos , finalmente el set -point , estos valores lo almacena en la memoria RAM interna del microcontrolador , en una tabla desde la posición de memoria 43H hasta 4AH

43H	44H	45H	46H	47H	48H	49H	4AH	4Bh
T	Td	Ti	K	HORAS		MINUTOS		S- POINT

**Subrutina de cruce por cero** sirve para programar y activar temporizador para el disparo del triac de acuerdo a los valores obtenidos en la rutina Proceso , a si mismo para realizar el control final de tiempo del proceso ha realizar

**Subrutina de visualización** , está subrutina nos permite mostrar la temperatura medida y luego el tiempo programado para realizar el proceso esto se realiza en cuatro display cada dos segundos ; primero se muestra la temperatura medida después de dos segundos luego el tiempo en horas y minutos .

Diagrama de flujo Nro.02 del programa: Consta de las siguientes partes



## **CONCLUSIONES**

Una vez concluida cada etapa del entrenador estas cumplieron con los objetivos planteados en el desarrollo inicial de ésta investigación:

- Construcción de un sistema entrenador con el microcontrolador 8031.
- Control de un proceso de tratamiento térmico en hornos de resistencias.
- Los puertos añadidos aumentan la potencialidad, se manejan más variables E/S.
- Uso del sistema entrenador en otros problemas reales.

Se logró obtener una herramienta de apoyo didáctico que se ve ampliada por la característica de poder establecer comunicación serial con una computadora personal a una velocidad de transmisión bastante aceptable, resaltando la función de programación desde la PC, con esto la programación se vuelve más rápida y fácil.

Es importante mencionar que la aplicación al proceso de tratamiento térmico en hornos de resistencias demuestra su versatilidad para usarse en control automático. Se comprobó, de acuerdo a un análisis costo/beneficios, que el entrenador resultó una excelente opción con un costo muy por debajo de otros equipos de control.

**ANEXO A**

**PROGRAMA MONITOR DEL MODULO DEL CONTROL**

**DIGITAL**

```

1 ;*****PROGRAMA DE GRABACION PARA MEMORIA RAM *****
2 ;
3
4 FCONT      BIT 2FH.0      ;indica que se inicia la llegada del program
5 FDISPLAY  BIT 2FH.1      ;indica que llego o que esta llegando el pro
6 FINTS      BIT 2FH.2      ;indica que se esta o no ejecutando el progr
7
8 ORG 0300H
9 DISPF: DB 0F7H,0F7H,0FBH,0FDH,0FEH,0FEH,0FEH,0FDH,0FBH,0F7H
10         DB 0F7H,0F7H,0FBH,0FDH,0FEH,0FEH,0FEH,0FDH,0FBH,0F7H
11         ;tabla que contiene los valores a mostrar en los displays
12         ;esta llegando el programa
13 ORG 0400H
14 DISPC: DB 0DFH,0FEH,0FEH,0FEH,0FEH,0FDH,0BFH,0BFH,0BFH,0BFH
15         DB 0EFH,0F7H,0F7H,0F7H,0F7H,0FBH,0BFH,0BFH,0BFH,0BFH
16         ;tabla que contiene los valores de activacion de los displ
17         ;mientras esta llegando el programa
18 ORG 0500H
19 DPG:    DB 0FFH,0FFH,0FFH,0FFH,0CH,0FFH,82H,0AFH,88H,83H,88H
20         DB 0A1H,0C0H,0FFH,0FFH,0FFH,0FFH
21         ;tabla que contiene los valores a mostrar en los displays
22         ;cuando ya llego el programa
23 ORG 0600H
24 DPC:    DB 0F7H,0FBH,0FDH,0FEH
25         ;tabla que contiene los valores de activacion de los displ
26         ;cuando ya llego el programa
27
28 ORG 0000H      ;
29 JMP ARESET     ;
30 ORG 0003H      ;
31 JMP AINT0      ;
32 ORG 000BH      ;
33 JMP ATIM0      ;
34 ORG 0013H      ;
35 JMP AINT1      ;
36 ORG 001BH      ;
37 ;CLR TF1      ;
38 JMP AT1        ;
39 ORG 0023H      ;
40 JMP ATR        ;direcciones de salto para todas las interrupciones
41
42 ORG 002BH
43 INICIO: MOV P1,#00      ;apaga los leds del puerto 1
44         CLR T0          ;desactiva disparo del primer triac
45         CLR T1          ;desactiva disparo del segundo triac
46         MOV IE,#00      ;desactiva todas las interrupciones
47         MOV IP,#00      ;desactiva la prioridad de las interrupc
48         CLR TRO         ;stop para el timer0
49         CLR TR1         ;stop para el timer1
50         SETB ES         ;activa interrupcion de la comunicacion s
51         SETB ET1        ;activa interrupcion del timer1
52         SETB EA         ;activa las interrupciones
53         MOV SCON,#50h   ;comunicacion serial en modo 1 y recepc
54         MOV TMOD,#20h   ;timer 1 en modo 2 y temporizador
55         MOV TH1,#239    ;velocidad a 1200 baudios
56         MOV TL1,#239    ;velocidad a 1200 baudios
57         SETB TR1        ;activa trabajo de timer 1
58         CLR FINTS      ;

```

```

59          CLR FDISPLAY          ;pone a cero fdisplay para espera de pr
60          CLR FCONT              ;
61
62 ADISPLAY: JNB FDISPLAY,DDISPLAY ;salta a ddisplay mientras llega
63             JMP PING             ;salta a ping cuando llego todo e
64 DDISPLAY: MOV R0,#00            ;inicializa desplazamiento en tab
65 NEWD:     MOV DPH,#04H          ;apunta a tabla con los valores p
66             MOV DPL,R0          ;segmentos del display
67             MOV A,#00           ;
68             MOVC A,@A+DPTR      ;obtiene valor de tabla
69             MOV DPTR,#4000H     ;apunta a los segmentos de los di
70             MOVX @DPTR,A        ;envia valor a los segmentos
71             MOV DPH,#03H        ;apunta a tabla con valores para
72             MOV DPL,R0          ;los comunes de los displays
73             MOV A,#00           ;
74             MOVC A,@A+DPTR      ;obtiene valor de la tabla
75             MOV DPTR,#6000H     ;apunta a los comunes de los disp
76             MOVX @DPTR,A        ;envia valor a los comunes
77             CALL DELAY          ;retardo de 1/4 para visualizacio
78             MOV A,#0FFH        ;
79             MOVX @DPTR,A        ;apaga displays
80             INC R0              ;incrementa desplazamiento en la
81             CJNE R0,#20,NEWD     ;salta a newd si aun no acaba la
82             JMP ADISPLAY        ;salta a display a inicializar de
83
84 PING:     MOV 40H,#00           ;iniciliza desplazamiento de mens
85 CNEW:     MOV R2,#50            ;# de veces que se mostrara el mi
86 DNEW:     MOV R0,40H           ;carga inicio de desplazamiento
87             MOV R1,#00         ;inicializa desplazamiento de com
88 NEW:      MOV DPL,R0           ;apunta a tabla que contiene el m
89             MOV DPH,#5          ;los segmenntos de los displays
90             MOV A,#00           ;
91             MOVC A,@A+DPTR      ;obtiene valor de la tabla
92             MOV DPTR,#4000H     ;aputa a segmentos de displays
93             MOVX @DPTR,A        ;envia valor a segmentos de los d
94             MOV DPL,R1          ;apunta a tabla que conntiene los
95             MOV DPH,#6          ;activacion para los comunes de l
96             MOV A,#00           ;
97             MOVC A,@A+DPTR      ;obtiene valor de la tabla
98             MOV DPTR,#6000H     ;apunta a comunes de los displays
99             MOVX @DPTR,A        ;envia valor a comunes de los dis
100            CALL DELAYM         ;retardo de 1 ms para visualizaci
101            MOV A,#0FFH        ;
102            MOVX @DPTR,A        ;apaga displays
103 SMENSAJE: INC R0              ;aumenta desplazamiento en tabla
104            INC R1              ;aumenta desplazamiento en tabla
105            CJNE R1,#04,NEW      ;salta a new si no se activo todo
106            DJNZ R2,DNEW         ;salta a dnew para mostrar el mis
107            INC 40H             ;aumenta desplazamiento de mensaj
108            MOV R0,40H          ;
109            CJNE R0,#14,CNEW     ;salta cnew si falta mostrar el m
110            SETB FINITS         ;activa flag indicador de program
111            MOV 70H,#03         ;numero de resets para nuevo prog
112            MOV 71H,#55H        ;valor para ejecucion de programa
113            JMP ARESET          ;salta a ejecutar programa recibi
114
115 DELAY:   MOV R3,#1            ;
116 DEL1:    MOV R4,#40h         ;

```

```

117 DELO:    MOV R5,#80h      ;
118 DELX:    DJNZ R5,DELX    ;
119          DJNZ R4,DELO    ;
120          DJNZ R3,DEL1    ;retardo de 1/4 segundo aproximad
121          RET              ;
122
123 DELAYM:   MOV R3,#3       ;
124 DEL1M:   MOV R4,#10      ;
125 DELOM:   MOV R5,#10      ;
126 DELXM:   DJNZ R5,DELXM   ;
127          DJNZ R4,DELOM   ;
128          DJNZ R3,DEL1M   ;retardo de 1 ms aproximadamente
129          RET              ;
130
131 ARESET:   MOV R0,71H      ;carga valor de ejecucion o no de
132          CJNE R0,#55H,SINI ;salta a sini si recien se encien
133          JB  FINTS,SPROG   ;salta si flag de resets aun esta
134          CLR FINTS        ;desactiv flags de resets
135 SINI:     JMP INICIO      ;salta a ejecutar programa bios
136
137 SPROG:    MOV R0,#10      ;
138 STOP:     CALL DELAY      ;
139          DJNZ R0,STOP      ;tiempo de espera para evitar rui
140          MOV R7,70H        ;carga numero de reset para el pr
141          DJNZ R7,FGRE      ;decrementa y ejecuta nuevamente
142          CLR FINTS        ;desactiva indicador de ejecucion
143          JMP INICIO        ;salta a ejecutar programa bios
144 FGRE:     MOV 70H,R7      ;guarda numero de resets
145          MOV IE,#00       ;
146          MOV IP,#00       ;
147          CLR TR0          ;
148          CLR TFO          ;
149          CLR TR1          ;
150          CLR TF1          ;
151          CLR RI           ;
152          CLR TI           ;
153          CLR TO           ;
154          CLR T1           ;
155          MOV P1,#00       ;
156          MOV DPTR,#6000H  ;
157          MOV A,#0FFH      ;desactiva todo para ejecucion de
158          MOVX @DPTR,A     ;programa recibido
159          SETB FINTS       ;activa indicador de ejecucion de
160          JMP EJPG         ;salta a ejecutar nuevo programa
161
162 AT1:      JNB FINTS,NOPS   ;salta a nops si indicador no est
163          JMP ATIM1        ;salta a subrutina de timer 1
164 NOPS:     CLR TF1         ;
165          RETI             ;
166
167 ATR:      JNB FINTS,GRABA  ;salta a graba si indicador no es
168          JMP ATXRX        ;salta a subrutina de comunicacion
169 GRABA:    CLR RI          ;desactiva indicador de recepcion
170          JNB FCONT,RCONT  ;salta a rcont si indicador esta
171          MOV 40H,A         ;salva valor de A
172          mov 42h,dpl       ;salva valor de dpl
173          mov 41h,dph       ;salva valor de dph
174          MOV A,SBUF        ;lee valor del puerto serial

```

```

75      mov dph,50h          ;dptr se carga para
76      mov dpl,51h          ;puntar a las celdas de la memori
77      MOVX @DPTR,A        ;carga valor en celda direccionad
78      inc dptr             ;incrementa puntero de memoria ra
79      mov 50h,dph         ;
80      mov 51h,dpl         ;guarda valor incrementado
81      MOV A,40H           ;recupera el valor de A
82      mov dpl,42h         ;recupera el valor de dpl
83      mov dph,41h         ;recupera el valor de dph
84      DJNZ R6,voc         ;decrementa contador de capacidad
85      CJNE R7,#00H,SALIR  ;salta si no se grabo aun el prog
86      CLR FCONT           ;desactiva indicador de cuenta
87      MOV P1,#55H         ;indica en p1 que se grabo el pro
88      SETB FDISPLAY       ;activa indicador de programa gra
89 SALIR: RETI
90 voc:  cjne r6,#0ffh,salir ;
91      dec r7              ;actualiza contador mas significat
92      reti
93
94 RCONT: setb fcont        ;activa indicador de llegada de p
95      mov r7,#07h         ;
96      mov r6,#0ffh       ;inicializa contador de 2k de cap
97      mov 50h,#20h       ;
98      mov 51h,#00h       ;apunta a la posicion de inicio d
99      RETI
200
201 ORG 2000H                ;
202 EJPG:                    ;
203 ORG 2003H                ;
204 AINT0:                   ;
205 ORG 200BH                ;
206 ATIM0:                   ;
207 ORG 2013H                ;
208 AINT1:                   ;
209 ORG 201BH                ;
210 ATIM1:                   ;
211 ORG 2023H                ;etiquetas y direcciones de las inte
212 ATXRX:                   ;del programa recibido
213 END^Z

```

## **ANEXO B**

**PROGRAMA DE CONTROL DE TEMPERATURA DE UN  
HORNO PARA TRATAMIENTO TERMICO DE CONTROL PID**

```
1 ;*****PROGRAMA DE CONTROL DE TEMPERATURA DE UN HORNO
2 ;***** PARA TRATAMIENTO TERMICO CON CONTROL PID***
3
4 ;30h   almacena temporalmente el valor leido del adc
5       para visualizacion
6
7   *****constantes de trabajo q0. q1 y q2 ***
8
9 ;31h   constante q0
10 ;32h   constante q1
11 ;33h   constante q2
12
13 ;**errores en los puntos de muestreo k, k-1, k-2; y
14     signos respectivos***
15
16 ;34h   error actual ek
17 ;35h   error anterior e(k-1)
18 ;36h   error santerior e(k-2)
19 ;20h.0  signo de e(k)
20 ;20h.1  signo de e(k-1)
21 ;20h.2  signo de e(k-2)
22
23 ;*****salidas de control en k y k-1 y signos****
24
25 ;37h,38H  salida anterior u(k-1)
26 ;39H,3AH  salida actual uk
27 ;20h.3    signo de u(k-1)
28 ;20h.4    signo de u(k)
29
30 ;operadores q2*e(k-2)  q1*e(k-1) y q0*ek
31 ;3bh,3ch   q2*e(k-2)
32 ;3dh,3eh   q1*e(k-1)
33 ;3fh,40h   q0*ek
34
35 ;*****variable para el calculo de operadores*****
36
37 ;41h,42h   valor que inicia con 8000h y se va sumando o restando
38           los
39           operadores
40 ;*****constantes de trabajo*****
41 ;43h   T constante de muestreo
42 ;44h   Td tiempo derivativo
43 ;45h   Ti tiempo de integracion
44 ;46h   K ganancia del regulador
45 ;4bh   set point
46
47 ;tiempo de proceso y flag de on
48 ;47h,48h  horas
49 ;49h,4ah  minutos
50 ;20h.5    flag de activacion del tiempo de duracion
51
52           Programa principal*****
53 ftrans  bit 21h.0
54 fctriac bit 20h.7
55 allcont bit 21h.1
56 org 0000H
57 jmp ini
58 org 0003H
59 jmp aint0
60 org 000bh
61 jmp timer00
62 org 001bh
63 jmp timer01
64 org 0023h
65 jmp comser
66 org 0600h
```

```
67 tabla: db 0c0h.0f9h.0a4h.0b0h.99h.92h.82h.0f8h.80h.90h
68 org 0700h
69 val:    db 00
70 kq0:   db 4
71 kq1:   db 5
72 kq2:   db 3
73 ek0:   db 5
74 ek1:   db 0
75 ek2:   db 0
76 uk1:   db 0.0
77 uk0:   db 0.0
78 op2:   db 0.0
79 op1:   db 0.0
80 op0:   db 0.0
81 vw:    db 80h.0
82 T:     db 2
83 Td:    db 2
84 Tii:   db 3
85 k:     db 2
86 hora:  db 0.4
87 minu:  db 3.0
88 setp:  db 120
89
90 org 002bh
91 ini:   mov a,#0c0h
92        mov dptr,#8000h
93        movx @dptr,a
94        mov dptr,#6000h
95        movx @dptr,a
96        mov dptr,#4000h
97        movx @dptr,a
98        mov dptr,#2000h
99        movx @dptr,a
100      call inival
101      call calq0q1q2
102      call actint
103      setb 20h.5
104
105 new:   mov dptr,#0ffffh
106        movx @dptr,a
107        call dladc ;retardo para lectura del adc
108        mov a,p1
109        mov 30h,a
110        call proceso
111        call visual ;muestra tiempo y/o temperatura
112        call delay ;retardo de muestreo
113        jb 20h.5,temp
114 hor:   setb 20h.5
115        jmp new
116 temp:  clr 20h.5
117        jmp new
118
119 proceso:mov 36h.35h
120        clr 20h.2
121        jnb 20h.1,actek1
122        setb 20h.2 ;asigna valor y signo de e(k-1) a e(k-2)
123 actek1: mov 35h.34h
124        clr 20h.1
125        jnb 20h.0,inip
126        setb 20h.1 ;asigna valor y signo de e(k) a e(k-1)
127
128 inip:  mov a,4bh
129        subb a,30h
130        mov 34h,a
131        clr 20h.0
132        jnc contop
133        mov a,30h
```

```
134          subb a,4bh
135          mov 34h,a
136          setb 20h.0
137
138 contop:   mov a,33h      q2
139          mov b,36h      e(k-2)
140          mul ab
141          mov 3bh,b
142          mov 3ch,a      q2*e(k-2)
143          mov a,32h      q1
144          mov b,35h      e(k-1)
145          mul ab
146          mov 3dh,b
147          mov 3eh,a      q1*e(k-1)
148          mov a,31h      q0
149          mov b,34h      e(k)
150          mul ab
151          mov 3fh,b
152          mov 40h,a      q0*e(k)
153          mov 41h,#80h
154          mov 42h,#00h   ;inicializa valor de partida
155
156 oper2:    jb 20h.2,resta2;q2*e(k-2)
157 suma2:    clr c
158          mov a,42h
159          add a,3ch
160          mov 42h,a
161          mov a,41h
162          addc a,3bh
163          mov 41h,a
164          jmp oper1
165 resta2:   clr c
166          mov a,42h
167          subb a,3ch
168          mov 42h,a
169          mov a,41h
170          subb a,3bh
171          mov 41h,a
172 oper1:    jnb 20h.1,resta1;q1*e(k-1)
173 suma1:    clr c
174          mov a,42h
175          add a,3eh
176          mov 42h,a
177          mov a,41h
178          addc a,3dh
179          mov 41h,a
180          jmp oper0
181 resta1:   clr c
182          mov a,42h
183          subb a,3eh
184          mov 42h,a
185          mov a,41h
186          subb a,3dh
187          mov 41h,a
188
189 oper0:    jb 20h.0,resta0;q0*e(k)
190 suma0:    clr c
191          mov a,42h
192          add a,40h
193          mov 42h,a
194          mov a,41h
195          addc a,3fh
196          mov 41h,a
197          jmp operx
198 resta0:   clr c
199          mov a,42h
200          subb a,40h
```

```
201      mov 42h,a
202      mov a,41h
203      subb a,3fh
204      mov 41h,a
205
206 operx:  jb 20h.3,restax;q0*e(k)
207 sumax:  clr c
208      mov a,42h
209      add a,38h
210      mov 42h,a
211      mov a,41h
212      addc a,37h
213      mov 41h,a
214      jmp salok
215 restax: clr c
216      mov a,42h
217      subb a,38h
218      mov 42h,a
219      mov a,41h
220      subb a,37h
221      mov 41h,a
222
223 salok:  clr c
224      mov a,41h
225      subb a,#80h
226      jnc respos
227 resneg: setb 20h.3
228 maximo: mov 39h,#0ffh
229      mov 3ah,#0d0h      ;valor de trabajo para el timer0
230      mov 37h,#00h
231      mov 38h,#00h
232      ret
233 respos: clr c
234      mov 41h,a
235      mov a,#60h
236      subb a,42h
237      mov a,#09h
238      subb a,41h
239      jc minimo
240 ctrol:  mov 38h,42h      ;actualizando valor de 1/2 u(k-1)
241      mov 37h,41h      ;actualizando valor de 1/2 u(k-1)
242      mov a,41h
243      cpl a
244      mov 39h,a
245      mov a,42h
246      cpl a
247      mov 3ah,a
248      clr 20h.3
249      ret
250
251 minimo: mov 39h,#0f7h
252      mov 3ah,#9fh      ;valor de trabajo para el timer0
253      mov 37h,#09h
254      mov 38h,#60h
255      clr 20h.3
256      ret
257
258 visual: jb 20h.5,tempera
259 tiempo: mov dptr,#0600h
260      mov a,4ah
261      movc a,@a+dptr
262      mov dptr,#8000h
263      movx @dptr,a
264      mov dptr,#0600h
265      mov a,49h
266      movc a,@a+dptr
267      mov dptr,#6000h
```

```
268      movx @dptr,a
269      mov dptr,#0600h
270      mov a,48h
271      movc a,@a+dptr
272      mov dptr,#4000h
273      movx @dptr,a
274      mov dptr,#0600h
275      mov a,47h
276      movc a,@a+dptr
277      mov dptr,#2000h
278      movx @dptr,a
279      ret
280
281 tempera:  mov a,30h      ;a=variable medida
282 exx:      mov b,#10
283           div ab
284           mov 60h,b
285           mov b,#10
286           div ab
287           mov 61h,b
288           mov 62h,a      r2r1r0=variable medida en decimal.
289           mov a,60h
290           mov b,#5
291           mul ab
292           mov 60h,a
293           mov a,61h
294           mov b,#5
295           mul ab
296           mov 61h,a
297           mov a,62h
298           mov b,#5
299           mul ab
300           mov 62h,a
301           mov a,60h
302           subb a,#10
303           jnc dig1
304           mov a,#0
305           jmp dig2
306 dig1:    mov a,60h
307           mov b,#10
308           div ab
309           mov 60h,b
310 dig2:    add a,61h
311           mov 61h,a
312           subb a,#10
313           jnc dig2a
314           mov a,#0
315           jmp dig3
316 dig2a:   mov a,61h
317           mov b,#10
318           div ab
319           mov 61h,b
320 dig3:    add a,62h
321           mov 62h,a
322           subb a,#10
323           jnc dig3a
324           mov 63h,#0
325           jmp display
326
327 dig3a:   mov a,62h
328           mov b,#10
329           div ab
330           mov 62h,b
331           mov 63h,a
332 display: mov dptr,#0600h
333           mov a,60h
334           movc a,@a+dptr
```

```
335      mov dptr,#8000h
336      movx @dptr,a
337      mov dptr,#0600h
338      mov a,61h
339      movc a,@a+dptr
340      mov dptr,#6000h
341      movx @dptr,a
342      mov dptr,#0600h
343      mov a,62h
344      movc a,@a+dptr
345      mov dptr,#4000h
346      movx @dptr,a
347      mov dptr,#0600h
348      mov a,63h
349      movc a,@a+dptr
350      mov dptr,#2000h
351      movx @dptr,a
352      ret
353
354 delay:  mov r7,#1
355 delx:   mov r6,#80h
356 dely:   mov r5,#0ffh
357 delw:   djnz r5,delw
358         djnz r6,dely
359         djnz r7,delx
360         ret
361
362 dladc:  mov r7,#2
363 delu:   mov r6,#0ffh
364 delc:   mov r5,#0ffh
365 delg:   djnz r5,delg
366 delt:   djnz r6,delc
367         djnz r7,delu
368         ret
369
370 calq0q1q2:  mov a,46h
371             mov b,44h
372             mul ab
373             mov b,43h
374             div ab
375             mov 33h,a      :q2=k*Td/T
376
377             mov a,44h
378             mov b,#2
379             mul ab
380             mov b,43h
381             clr c
382             add a,b
383             mov b,43h
384             div ab
385             inc a
386             mov b,46h
387             mul ab
388             mov 32h,a      :q1=k(1 + 2 Td/T )
389
390             mov a,43h
391             mov b,43h
392             mul ab
393             mov 31h,a
394             mov a,45h
395             mov b,44h
396             mul ab
397             clr c
398             add a,31h
399             mov 31h,a
400             mov a,43h
401             mov b,45h
```

```
:
: k=46h
: Td=44h
: T=43h
: q2=33h
:
:
:
:
: k=46h
: Td=44h
: T=43h
: q1=32h
:
:
:
:
:
:
:
:
:
:
:
: k=46h
```

```
402          mul ab          ;Td=44h
403          mov b,a         ;T=43h
404          mov a,31h       ;Ti=45h
405          div ab          ;q0=31h
406          inc a
407          mov b,46h
408          mul ab
409          mov 31h,a       ;q0=k( 1 + T/Ti + Td/T )
410          ret
411 ;-----
```

```
412 aint0:    clr ex0        ;desactiva interrupcion
413          clr ie0        ;borra flag de interrupcion de
414          jnb allcont,stim ;CRUCE POR CERO
415          mov th0,39h     ;programando
416          mov tl0,3ah     ;tiempo de trabajo
417          setb tr0       ;del timer1
418          jnb ftrans,contime
419          mov r4,34h
420 col:     cjne r4,#00,co2
421          setb ftrans
422          jmp contime
423 co2:     cjne r4,#01,stim
424          setb ftrans
425 contime: mov r4,51h
426          cjne r4,#00,stimx
427          mov r4,50h
428          cjne r4,#00,stimx
429          clr fc triac
430          clr ftrans
431          clr allcont
432          jmp stim
433
434 stimx:   dec 53h
435          mov r4,53h
436          cjne r4,#0ffh,stim
437 ver1:   mov 53h,#118
438          dec 52h
439          mov r4,52h
440          cjne r4,#0ffh,stim
441          mov 52h,#59
442          dec 51h
443          mov r4,51h
444          cjne r4,#0ffh,stim
445          mov 51h,#59
446          dec 50h
447
448 stim:   setb ex0        ;se activa la interrupcion de
449          ; cruce por cero
450          reti
451 ;-----
```

```
452 timer00: clr tr0
453          clr tf0
454          jnb fc triac,salt
455          setb t0         ;pulso
456          mov r3,#15     ;de
457 deltri: djnz r3,deltri  ;disparo
458 salt:   clr t0         ;del triac
459          reti
460
461 timer01: clr tf1
462          reti
463
464 ;::recibe los parametros T,Td,Ti,K,Horas,Minutos,Setp:
465 comser:  clr es
466          clr ri
467          mov 70h,a
```

```
468         jnb 20h.6,recibe
469         mov a,sbuf
470         cjne a,#55h,salir
471         setb 20h.6
472         mov r0,#43h
473 salir:    mov a,70h
474         setb es
475         reti
476 recibe:  mov a,sbuf
477         cjne a,#0aah,newv
478         mov a,47h
479         mov b,#10
480         mul ab
481         clr c
482         add a,48h
483         mov 50h,a ;horas
484         mov a,49h
485         mov b,#10
486         mul ab
487         clr c
488         add a,4ah
489         mov 51h,a ;minutos
490         mov 52h,#59
491         mov 53h,#118
492         setb fc triac
493         clr allcont
494         clr ftrans
495         clr 20h.6
496         jmp salir
497 newv:    mov @r0,a
498         inc r0
499         jmp salir
500 ;-----
501 actint:  setb ex0 ;activa interrupcion int0 (cruce por cero)
502         setb it0
503         setb es
504         clr tr0
505         clr tr1
506         setb et0 ;activa interrupcion timer0
507         setb et1 ;activa interrupcion timer1
508         mov tmod,#21h ;timer1 en modol 16 bits, temporizador.
509         ;timer0 en modo 2, autorecarga
510         mov scon,#50h ;comunicacion serial modol. recepcion
511         mov th1,#248 ;
512         mov tl1,#248 ;valor de trabajo para que la c. serial
513         setb tr1 ;sea de 1200 baudios
514         setb ea
515         ret
516 ;-----
517 inival:  clr 20h.0
518         clr 20h.1
519         clr 20h.2
520         clr 20h.3
521         clr 20h.4
522         clr 20h.5 ; pone a cero todos los signos (+)
523         clr 20h.6
524         clr fc triac
525         clr ftrans
526         clr allcont
527         clr t0
528         clr t1
529         mov dptr,#0700h
530         mov r0,#30h
531         mov r1,#00
532 sini:   mov a,r1
533         movc a,@a+dptr
534         mov @r0,a ;
```

```
535         inc r1
536         inc r0
537         cjne r0,#4ch,sini:
538         mov 35h,#00      :inicializa e(k-1)
539         mov 36h,#00      :inicializa e(k-2)
540         mov 34h,4bh      :inicializa e(k)
541         mov 37h,#00
542         mov 38h,#00      :inicializa u(k-1)
543         ret
544
545         end
```

## **ANEXO C**

### **PROGRAMACION DE PARAMETROS INTERFACE CON PC**

```
1 ;**** PROGRAMA DE PROGRAMACION DE PARAMETROS Y INTERFACE SERIAL
  CON PC*****
2 ,,,. PROGRAMA PARA MOSTRAR LA INTERFASE CON EL USUARIO Y REALIZAR
  SU PROGRAMACION;;;
3 ,,,, DE PARAMETROS Y INTERFACE SERIAL CON EL SISTEMA DE CONTROL
  .....
4 @pcserial macro parm
5     @salva
6     mov ah,00
7     mov dx,01
8     mov al,parm
9     int 14h
10    @recu
11    endm
12
13 @dserial macro valor
14    @salva
15    mov ah,1
16    mov dx,01
17    mov al,valor
18    int 14h
19    @recu
20    endm
21
22 include c:\electron\8088\mac1.asm
23 include c:\electron\8088\macc2.asm
24 include c:\electron\8088\mac2b.asm
25 include c:\electron\8088\mac3.asm
26 datos segment
27 include c:\electron\8088\vari1.asm
28 cops      db 00
29 cops1     db 00
30 cops2     db 00
31 copb      db 00
32 copx      db 00
33 copm      db 00
34 copk      db 00
35 copt      db 00
36 copi      db 00
37 copd      db 00
38 date      db 00
39 copg      db 00
40 copp      db 00
41 coph      db 00
42 copn      db 00
43 copct     db 00
44 vsetp     dw 00
45 dirtabt   dw 00
46 dirtabm   dw 00
47 tablat    db 10 dup (00)
48 tablam    db 20 dup (00)
49 clavic    db 'UNI$'
50 clavi     db 3 dup (00)
51 scape     db 'Presione Esc para salir$'
52 cotem     db 'CONTROL DE TEMPERATURA$'
```

```
53 param db 'Parametros$'  
54 contr db 'Control$'  
55 mostr db 'Mostrar$'  
56 envia db 'Enviar$'  
57 kkmm db 'K :$'  
58 ttmm db 'T :$'  
59 timm db 'Ti :$'  
60 tdmm db 'Td :$'  
61 spmm db 'Sp :$'  
62 homm db 'H :$'  
63 mimm db 'M :$'  
64  
65 spm db 'Sp:$'  
66 temper db 'CONTROLADOR DE TEMPERATURA DE HORNOS DE TRATAMIENTO TE  
RMICO$'  
67 men db 'Desea guardar las modificaciones? (s/n): $'  
68 errors db 'clave erronea!!!$'  
69 ingcla db 'ingrese clave: $'  
70 NULL DB '  
71 tabces db 60 dup (00)  
72 kkp db 'K : $'  
73 ttp db 'T : $'  
74 tip db 'Ti : $'  
75 tdp db 'Td : $'  
76 spo db 'Setp: $'  
77 hor db 'Horas: $'  
78 min db 'Minutos: $'  
79 datos ends  
80  
81 codigo segment  
82 assume cs:codigo,ds:datos  
83 inicio: mov ax,datos  
84 mov ds,ax  
85 @video 12h  
86 nbv: @pcserial 83h  
87 @CAJVID 0,0,639,479,2,0BH  
88 mov cops,00  
89 mov copb,00  
90 mov copt,00  
91 @cajaf 40,40,154,133,07  
92 @cajap 40,40,154,133,0eh  
93 @boton 50,50,143,62,2,param,0eh,00,2,2,cops,1,0ah  
94 @boton 50,70,143,82,2,contr,0eh,00,2,2,copb,1,0ah  
95 @boton 50,90,143,102,2,mostr,0eh,00,2,2,copm,1,0ah  
96 @boton 50,110,143,122,2,envia,0eh,00,2,2,copx,1,0ah  
97 @CAJAF 38,398,602,418,07  
98 @boton 40,400,600,416,2,temper,0eh,00,12,3,copct,1,0ah  
99  
100 tec: @wkey  
101 cmp al,27  
102 jne cmpar  
103 jmp salir  
104 cmpar: cmp al,'p'  
105 je parametro  
106 cmp al,'P'
```

```
107         je parametro
108         cmp al,'c'
109         je controla
110         cmp al,'C'
111         je controla
112         cmp al,'m'
113         je mostra
114         cmp al,'M'
115         je mostra
116         cmp al,'e'
117         je enviar
118         cmp al,'E'
119         je enviar
120         jmp tec
121 parametro: jmp par
122 controla:  jmp con
123 mostra:   jmp mos
124 enviar:   jmp env
125
126 par:      @boton 50,50,143,62,2,param,0eh,00,2,2,cops,1,0ah
127           @cajaf 240,40,354,133,07
128           @cajap 240,40,354,133,01h
129           mov bx,offset tablat
130 kkk:      @boton 250,50,343,62,2,kkp,0eh,00,2,2,copk,1,0ah
131           @boton 250,50,343,62,2,kkp,0eh,00,2,2,copk,1,0ah
132           @cajaf 300,60,305,61,01
133           @wkeyn
134           mov [bx],al
135           inc bx
136           @letg 300,53,date,01
137           @cajaf 300,60,305,61,07
138           @cajaf 310,60,315,61,01
139           @wkeyn
140           mov [bx],al
141           inc bx
142           @letg 310,53,date,01
143           @cajaf 310,60,315,61,07
144
145 ttt:      @cajaf 300,80,305,81,01
146           @boton 250,70,343,82,2,ttp,0eh,00,2,2,copt,1,0ah
147           @boton 250,70,343,82,2,ttp,0eh,00,2,2,copt,1,0ah
148           @wkeyn
149           mov [bx],al
150           inc bx
151           @letg 300,73,date,01
152           @cajaf 300,80,305,81,07
153           @cajaf 310,80,315,81,01
154           @wkeyn
155           mov [bx],al
156           inc bx
157           @letg 310,73,date,01
158           @cajaf 310,80,315,81,07
159
160 tti:      @cajaf 300,100,305,71,01
161           @boton 250,90,343,102,2,tip,0eh,00,2,2,copi,1,0ah
```

```
162         @boton 250,90,343,102,2.tip,0eh,00,2,2,copi,1,0ah
163         @wkeyn
164         mov [bx],al
165         inc bx
166         @letg 300,93,date,01
167         @cajaf 300,100,305,101,07
168         @cajaf 310,100,315,101,01
169         @wkeyn
170         mov [bx],al
171         inc bx
172         @letg 310,93,date,01
173         @cajaf 310,100,315,101,07
174
175 ttd:     @cajaf 300,120,305,121,01
176         @boton 250,110,343,122,2,tdp,0eh,00,2,2,copd,1,0ah
177         @boton 250,110,343,122,2,tdp,0eh,00,2,2,copd,1,0ah
178         @wkeyn
179         mov [bx],al
180         inc bx
181         @letg 300,113,date,01
182         @cajaf 300,120,305,121,07
183         @cajaf 310,120,315,121,01
184         @wkeyn
185         mov [bx],al
186         inc bx
187         @letg 310,113,date,01
188         @cajaf 310,120,315,121,07
189
190         @cajaf 145,155,535,176,07
191         @boton 150,160,530,172,2,men,0eh,00,2,2,copg,1,0ah
192         @boton 150,160,530,172,2,men,0eh,00,2,2,copg,1,0ah
193         @cajaf 520,170,525,171,01
194
195 stec:   @wkey
196         cmp al,'s'
197         jne seno
198         jmp guarda
199 seno:   cmp al,'n'
200         jne stec
201         jmp borra
202
203 guarda: mov date,al
204         @letg 520,163,date,01h
205         @cajaf 160,190,350,202,07
206         @textg 162,192,ingcla,01
207         @cajaf 320,201,325,202,01
208
209         mov bx,offset clavi
210         mov cx,3
211 sclav:  @wkey
212         mov [bx],al
213         INC BX
214         loop sclav
215         mov bx,offset clavic
216         mov dirtabt,bx
```

```
217         mov bx,offset clavi
218         mov dirtabm,bx
219         mov cx,3
220 scomp:   mov al,[bx]
221         mov dl,al
222         mov bx,dirtabt
223         mov al,[bx]
224         cmp al,dl
225         jne error
226         inc dirtabt
227         inc dirtabm
228         mov bx,dirtabm
229         loop scomp
230         jmp okk
231 error:   @cajaf 160,190,350,202,07h
232         @textg 162,192,errors,0eh
233
234         @retardo 2500,5000
235
236         @cajaf 160,190,350,202,00
237         @cajvid 158,188,352,204,2,0bh
238         jmp borra
239
240 okk:     @cajaf 160,190,350,202,00
241         @cajvid 158,188,352,204,2,0bh
242         mov bx,offset tablam
243         mov dirtabm,bx
244         mov cx,8
245         mov bx,offset tablat
246         mov dirtabt,bx
247
248 sgu:     mov al,[bx]
249         mov bx,dirtabm
250         mov [bx],al
251         inc dirtabt
252         inc dirtabm
253         mov bx,dirtabt
254         loop sgu
255 borra:   @cajaf 145,155,535,176,00
256         @CAJVID 144,154,536,178,2,0BH
257         @cajaf 240,40,354,133,00
258         @CAJVID 238,38,556,132,2,0BH
259         @boton 50,50,143,62,2,param,0eh,00,2,2,cops,1,0ah
260         jmp tec
261
262 con:     @cajaf 240,40,354,133,07
263         @cajap 240,40,354,133,04
264 setpo:   @boton 250,50,343,62,2,spo,0eh,00,2,2,copp,1,0ah
265         @boton 250,50,343,62,2,spo,0eh,00,2,2,copp,1,0ah
266         @cajaf 300,60,305,61,01
267         mov bx,offset tablat
268         @wkeyn
269         mov [bx],al
270         inc bx
271         @letg 300,53,date,01
```

```
272      @cajaf 300,60,305,61,07
273      @cajaf 310,60,315,61,01
274      @wkeyn
275      mov [bx],al
276      inc bx
277      @letg 310,53,date,01
278      @cajaf 310,60,315,61,07
279      @cajaf 320,60,325,61,01
280      @wkeyn
281      mov [bx],al
282      inc bx
283      @letg 320,53,date,01
284      @cajaf 320,60,325,61,07
285      @cajaf 330,60,335,61,01
286      @wkeyn
287      mov [bx],al
288      inc bx
289      @letg 330,53,date,01
290      @cajaf 330,60,335,61,07
291
292 hora:  @boton 250,70,343,82,2,hor,0eh,00,2,2,coph,1,0ah
293        @boton 250,70,343,82,2,hor,0eh,00,2,2,coph,1,0ah
294        @cajaf 325,80,330,81,01
295        @wkeyn
296        mov [bx],al
297        inc bx
298        @letg 325,73,date,01
299        @cajaf 325,80,330,81,07
300        @cajaf 335,80,340,81,01
301        @wkeyn
302        mov [bx],al
303        inc bx
304        @letg 335,73,date,01
305        @cajaf 335,80,340,81,07
306
307 minu:  @boton 250,90,343,102,2,min,0eh,00,2,2,copn,1,0ah
308        @boton 250,90,343,102,2,min,0eh,00,2,2,copn,1,0ah
309        @cajaf 325,100,330,101,01
310        @wkeyn
311        mov [bx],al
312        inc bx
313        @letg 325,93,date,01
314        @cajaf 325,100,330,101,07
315        @cajaf 335,100,340,101,01
316        @wkeyn
317        mov [bx],al
318        inc bx
319        @letg 335,93,date,01
320        @cajaf 335,100,340,101,07
321
322      mov bx,offset tablam
323      add bx,8
324      mov dirtabm,bx
325      mov cx,8
326      mov bx,offset tablat
```

```
327         mov dirtabt,bx
328
329 sgus:    mov al,[bx]
330         mov bx,dirtabm
331         mov [bx],al
332         inc dirtabt
333         inc dirtabm
334         mov bx,dirtabt
335         loop sgus
336
337         jmp tec
338
339 mos:     @cajaf 240,40,354,193,07
340         @cajap 240,40,354,193,01h
341
342         @textg 260,53,kkmm,1
343         @textg 260,73,ttmm,1
344         @textg 260,93,ttmm,1
345         @textg 260,113,tdmm,1
346         @textg 260,133,spmm,1
347         @textg 260,153,homm,1
348         @textg 260,173,mimm,1
349         mov bx,offset tablam
350         mov al,[bx]
351         mov date,al
352         @letg 300,53,date,01
353         inc bx
354         mov al,[bx]
355         mov date,al
356         @letg 310,53,date,01
357
358         inc bx
359         mov al,[bx]
360         mov date,al
361         @letg 300,73,date,01
362         inc bx
363         mov al,[bx]
364         mov date,al
365         @letg 310,73,date,01
366
367         inc bx
368         mov al,[bx]
369         mov date,al
370         @letg 300,93,date,01
371         inc bx
372         mov al,[bx]
373         mov date,al
374         @letg 310,93,date,01
375
376         inc bx
377         mov al,[bx]
378         mov date,al
379         @letg 300,113,date,01
380         inc bx
381         mov al,[bx]
```

```
382         mov date,al
383         @letg 310,113,date,01
384
385 setpp:    inc bx
386         mov al,[bx]
387         mov date,al
388         @letg 300,133,date,01
389         inc bx
390         mov al,[bx]
391         mov date,al
392         @letg 310,133,date,01
393         inc bx
394         mov al,[bx]
395         mov date,al
396         @letg 320,133,date,01
397         inc bx
398         mov al,[bx]
399         mov date,al
400         @letg 330,133,date,01
401
402 horr:    inc bx
403         mov al,[bx]
404         mov date,al
405         @letg 300,153,date,01
406         inc bx
407         mov al,[bx]
408         mov date,al
409         @letg 310,153,date,01
410
411 minn:    inc bx
412         mov al,[bx]
413         mov date,al
414         @letg 300,173,date,01
415         inc bx
416         mov al,[bx]
417         mov date,al
418         @letg 310,173,date,01
419         @wkey
420         @cajaf 240,40,354,193,00
421         @cajvid 238,38,356,194,2,0bh
422         jmp tec
423
424
425 env:     @boton 50,110,143,122,2,envia,0eh,00,2,2,copx,1,0ah
426         mov date,55h
427         @dserial date
428         @RETARDO 500,10 ;TRANSMISION DE INICIO DE PARAMETROS
429         mov bx,offset tablam
430         mov vsetp,0
431         mov ax,0
432         MOV AL,[BX+2]
433         SUB AL,30H
434         MOV CX,10
435         MUL CX
436         MOV VSETP,AX
```

```
437      MOV AX,0
438      MOV AL,[BX+3]
439      SUB AL,30H
440      ADD AX,VSETP
441      MOV DATE,AL
442      @dserial date      ;
443      @RETARDO 500,10    ;TRANSMICION DE "T"
444
445      MOV AL,[BX+6]
446      SUB AL,30H
447      MOV CX,10
448      MUL CX
449      MOV VSETP,AX
450      MOV AX,0
451      MOV AL,[BX+7]
452      SUB AL,30H
453      ADD AX,VSETP
454      MOV DATE,AL
455      @dserial date      ;
456      @RETARDO 500,10    ;TRANSMICION DE "Td"
457
458      MOV AL,[BX+4]
459      SUB AL,30H
460      MOV CX,10
461      MUL CX
462      MOV VSETP,AX
463      MOV AX,0
464      MOV AL,[BX+5]
465      SUB AL,30H
466      ADD AX,VSETP
467      MOV DATE,AL
468      @dserial date      ;
469      @RETARDO 500,10    ;TRANSMICION DE "Ti"
470
471      MOV AL,[BX]
472      SUB AL,30H
473      MOV CX,10
474      MUL CX
475      MOV VSETP,AX
476      MOV AX,0
477      MOV AL,[BX+1]
478      SUB AL,30H
479      ADD AX,VSETP
480      MOV DATE,AL
481      @dserial date      ;
482      @RETARDO 500,10    ;TRANSMICION DE "K"
483
484      MOV AL,[BX+12]
485      SUB AL,30H
486      MOV DATE,AL
487      @dserial date      ;
488      @RETARDO 500,10    ;TRANSMICION DE "Horas"
489
490      MOV AL,[BX+13]
491      SUB AL,30H
```

```
492         MOV DATE,AL
493         @dserial date           ;
494         @RETARDO 500,10        ;TRANSMICION DE "Horas"
495
496         MOV AL,[BX+14]
497         SUB AL,30H
498         MOV DATE,AL
499         @dserial date           ;
500         @RETARDO 500,10        ;TRANSMICION DE "Minutos"
501
502         MOV AL,[BX+15]
503         SUB AL,30H
504         MOV DATE,AL
505         @dserial date           ;
506         @RETARDO 500,10        ;TRANSMICION DE "Minutos"
507
508         mov vsetp,0
509  asetp:   mov al,[bx+8]
510         sub al,30h
511         mov cx,1000
512         mul cx
513         add ax,vsetp
514         mov vsetp,ax
515         mov ax,0
516         mov al,[bx+9]
517         sub al,30h
518         mov cx,100
519         mul cx
520         add ax,vsetp
521         mov vsetp,ax
522         mov ax,0
523         mov al,[bx+10]
524         sub al,30h
525         mov cx,10
526         mul cx
527         add ax,vsetp
528         mov vsetp,ax
529         mov ax,0
530         mov al,[bx+11]
531         sub al,30h
532         add ax,vsetp
533         mov cx,5
534         div cx
535         mov date,al
536         @dserial date ;setp
537         @retardo 500,10
538         @boton 50,110,143,122,2,envia,0eh,00,2,2,copx,1,0ah
539         mov date,0aah
540         @dserial date ;setp
541         @retardo 500,10
542         jmp tec
543
544  salir:   @video 03
545         mov ah,4ch
546         int 21h
```

c:\electron\8051\cotf.asm  
Printed 08:31 22 Dec 99

Page

547  
548 codigo ends  
549 end inicio

**ANEXO D**

**CONJUNTO DE INSTRUCCIONES DEL MICROCONTROLADOR**

**8051**

# Juego de instrucciones del microcontrolador (8051)

ACALL addr11 .....	309	CPL <bit> .....	317
ADD A,<byte-fuente> .....	309	CPL C .....	318
ADDC A,Rn .....	310	CPL bit .....	318
ADDC A,direct .....	310	DA A .....	318
ADDC A,@Ri .....	310	DEC <byte> .....	319
ADDC A,#data .....	310	DEC A .....	319
ADDC A,<byte-fuente> .....	311	DEC Rn .....	319
ADDC A,Rn .....	311	DEC direct .....	320
ADDC A,direct .....	311	DEC @Ri .....	320
ADDC A,@Ri .....	311	DIV AB .....	320
ADDC A,#data .....	311	DJNZ <byte>,rel .....	321
AJMP addr11 .....	312	DJNZ Rn,rel .....	321
ANL <byte-destino>,<byte-fuente> ..	312	DJNZ direct,rel .....	321
ANL A,Rn .....	312	INC <byte> .....	322
ANL A,direct .....	312	INC A .....	322
ANL A,@Ri .....	313	INC Rn .....	322
ANL A,#data .....	313	INC direct .....	322
ANL direct,A .....	313	INC @Ri .....	322
ANL direct,#data .....	313	INC DPTR .....	323
ANL C,<bit-fuente> .....	313	JB bit,rel .....	323
ANL C,bit .....	314	JBC bit,rel .....	323
ANL C,/bit .....	314	JC rel .....	324
CJNE <byte-destino>,<byte-fuente>,rel.	314	JMP @A+DPTR .....	324
CJNE A,direct,rel .....	315	JNB bit,rel .....	325
CJNE A,#data,rel .....	315	JNC rel .....	325
CJNE Rn,#data,rel .....	315	JNZ (rel .....	326
CJNE @Ri,#data,rel .....	316	JZ bit,rel .....	326
CLR A .....	316	LCALL addr16 .....	326
CLR <bit> .....	316	LJMP addr16 .....	327
CLR C .....	317	MOV <byte-destino>,<byte-fuente> ..	327
CLR bit .....	317	MOV A,Rn .....	328
CPL A .....	317	MOV A,direct .....	328

MOV A,@ Ri	328	ORL <A>,<bit-fuente>	337
MOV A,# data	328	ORL C,bit	337
MOV Rn,A	329	ORL C,'bit	338
MOV Rn,direct	329	POP direct	338
MOV Rn,# data	329	PUSH direct	338
MOV direct,A	329	RET	339
MOV direct,Rn	329	RETI	339
MOV direct,direct	330	RL A	340
MOV direct.@ Ri	330	RLC A	340
MOV direct,# data	330	RR A	340
MOV @ Ri,A	330	RRC A	340
MOV @ Ri,direct	331	SETB <bit>	341
MOV @ Ri.# data	331	SETB C	341
MOV <bit-destino>,<bit-fuente>	331	SETB bit	341
MOV C,bit	331	SJMP rel	341
MOV bit,C	331	SUBB A,<byte-fuente>	342
MOV DPTR,# data16	332	SUBB A,Rn	342
MOVC A,@ A + <base-reg>	332	SUBB A,direct	342
MOVC A,@ A + DPTR	332	SUBB A,@ Ri	343
MOVC A,@ A + PC	333	SUBB A,# data	343
MOVX <byte-destino>,<byte-fuente>	333	SWAP A	343
MOVX A,@ Ri	333	XCH A,<byte>	344
MOVX A,@ DPTR	334	XCH A,Rn	344
MOVX @ Ri,A	334	XCH A,direct	344
MOVX @ DPTR,A	334	XCH A,@ Ri	344
MUL AB	334	XCHD A,@ Ri	345
NOP	335	XRL <byte-destino>,<byte-fuente>	345
ORL <byte-destino>,<byte-fuente>	335	XRL A,Rn	345
ORL A,Rn	336	XRL A,direct	345
ORL A,direct	336	XRL A,@ Ri	346
ORL A,@ Ri	336	XRL A,data	346
ORL A,# data	336	XRL direct,A	346
ORL direct,A	337	XRL direct,# data	346
ORL direct,# data	337		

## ■ NOTA

Rn	Registro R7-R0 del banco actual.
direct	Dirección 0-127 de RAM Interna o un SFR (128-255).
@ Ri	Dirección de RAM Interna apuntada por R1-R0.
# data	Dato inmediato de 8 bits.
# data16	Dato inmediato de 16 bits.
# addr16	Dirección de 16 bits (LCALL,LJMP).
# addr11	Dirección de 11 bits (ACALL,AJMP).
rel	Desplazamiento en saltos relativos de -128 a +127.
bit	Bit de RAM Interna o SFR Direcciónamiento directo.

■ INSTRUCCIONES QUE MODIFICAN LOS FLAGS

Instrucción	Flags			Instrucción	Flags		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,BIT	X		
RLC	X			CJNE	X		
SETB C	1						

■ JUEGO DE INSTRUCCIONES DEL 8051

• **ACALL addr11**

Llamada incondicional a un subprograma situado en la dirección indicada. El PC (Program Counter), o Contador de Programa, se incrementa hasta obtener la dirección de la siguiente instrucción a ejecutar, guarda dicha dirección en la PILA (el byte de menor peso primero) e incrementa el Stack Pointer (SP) o Apuntador de Pila. La dirección de llamada se obtiene encadenando a los 5 bits de mayor peso que ya tiene el PC con los bits 7-5 del código de operación y los 8 bits del segundo byte de la instrucción. De modo que el subprograma llamado debe comenzar dentro del mismo segmento de 2 Kbytes de memoria.

Bytes : 2  
Ciclos : 2

Código : 

A10 A9 A8 1	0001	A7 A6 A5 A4 A3 A2 A1 A0
-------------	------	-------------------------

Operación :  
 (PC) ← (PC) + 2  
 (SP) ← (SP) + 1  
 ((SP)) ← (PC 7 - 0)  
 (SP) ← (SP) + 1  
 ((SP)) ← (PC 15 - 8)  
 (PC A10-10) ← dirección de página

• **ADD A, <byte-fuente >**

Suma la variable indicada al acumulador, dejando el resultado en el acumulador. El carry y el auxiliar-carry se activan si se produce llevada en los bits 7 y 3, respectivamente. Cuando sumamos enteros sin signo, el flag de carry nos indica que se ha producido un rebosamiento.

El Overflow se activa si se produce llevada en el bit 7 pero no en el 6, o si ésta se produce en el 6 pero no en el 7. Cuando sumamos enteros con signo, OV nos indica que se ha producido un resultado negativo de la suma de ambos operandos, o un resultado positivo si ambos eran negativos.

Se pueden emplear cuatro modos de direccionamiento: por registro, directo, indirecto por registro, inmediato.

• **ADDC A,Rn**

Bytes : 1

Ciclos : 1

Código : 

0010	1rrr
------	------

Operación :  $(A) \leftarrow (A) + (Rn)$

• **ADDC A,direct**

Bytes : 2

Ciclos : 1

Código : 

0010	0101	direct
------	------	--------

Operación :  $(A) \leftarrow (A) + (\text{direct})$

• **ADDC A,@Ri**

Bytes : 1

Ciclos : 1

Código : 

0010	011i
------	------

Operación :  $(A) \leftarrow (A) + ((Ri))$

• **ADDC A,#data**

Bytes : 2

Ciclos : 1

Código : 

0010	0100	data
------	------	------

Operación :  $(A) \leftarrow (A) + \#data$

• **ADDC A, <byte-fuente>**

Opera de modo similar a la instrucción ADD, salvo que en ésta, además, se le suma el valor de carry.

• **ADDC A, Rn**

Bytes : 1  
Ciclos : 1

Código : 

0010	1rrr
------	------

Operación :  $(A) \leftarrow (A) + (C) + (Rn)$

• **ADDC A, direct**

Bytes : 2  
Ciclos : 1

Código : 

0010	0101	direct
------	------	--------

Operación :  $(A) \leftarrow (A) + (C) + (\text{direct})$

• **ADDC A, @Ri**

Bytes : 1  
Ciclos : 1

Código : 

0011	011i
------	------

Operación :  $(A) \leftarrow (A) + (C) + ((Ri))$

• **ADDC A, #data**

Bytes : 2  
Ciclos : 1

Código : 

0011	0100	data
------	------	------

Operación :  $(A) \leftarrow (A) + (C) + \#data$

**• AJMP addr11**

Realiza un salto incondicional a la dirección indicada. La forma en que obtiene la dirección es igual a la empleada por la instrucción ACALL, y del mismo modo la dirección de salto debe estar dentro del mismo segmento de 2 Kbytes de memoria.

Las instrucciones de salto lo único que hacen es modificar el contenido del PC.

Bytes : 2  
Ciclos : 2

Código : 

A10 A9 A8 0	0001	A7 A6 A5 A4 A3 A2 A1 A0
-------------	------	-------------------------

Operación :  $(PC) \leftarrow (PC) + 2$   
 $(PC\ A10-A0) \leftarrow \text{dirección de página}$

**• ANL <byte-destino>, <byte-fuente>**

Realiza la función lógica AND entre las variables indicadas y almacena el resultado en la variable destino.

Ambos operandos permiten seis modos de direccionamiento.

Cuando el destino es el acumulador, el fuente puede ser: registro, indirecto por registro, inmediato.

Cuando el destino es una dirección direct, el fuente puede ser: acumulador, inmediato.

Cuando esta instrucción se emplea para modificar el valor de un puerto, el valor del puerto no se lee de las patillas, sino que lo hace del «latch» de salida.

**• ANL A,Rn**

Bytes : 1  
Ciclos : 1

Código : 

0101	1rrr
------	------

Operación :  $(A) \leftarrow (A) \wedge (Rn)$

**• ANL A,direct**

Bytes : 2  
Ciclos : 1

Código : 

0101	0101	direct
------	------	--------

Operación :  $(A) \leftarrow (A) \wedge (\text{direct})$

• ANL A,@Ri

Bytes : 1

Ciclos : 1

Código : 0101 011i

Operación :  $(A) \leftarrow (A) \wedge ((Ri))$

• ANL A,#data

Bytes : 2

Ciclos : 1

Código : 0101 0100 data

Operación :  $(A) \leftarrow (A) \wedge \#data$

• ANL direct,A

Bytes : 2

Ciclos : 1

Código : 0101 0010 direct

Operación :  $(direct) \leftarrow (direct) \wedge (A)$

• ANL direct,#data

Bytes : 3

Ciclos : 2

Código : 0101 0011 direct data

Operación :  $(direct) \leftarrow (direct) \wedge \#data$

• ANL C,<bit-fuente>

Es la función lógica AND a nivel de «bit», donde el carry hace de acumulador. La barra o slash (/) puede preceder al operador indicando el «complemento lógico» del bit-fuente, pero éste no resulta afectado. Solamente se permiten bit-fuente con direccionamiento directo.

• ANL C,bit

Bytes : 2  
Ciclos : 2

Código : 

1000	0010
------	------

bit address
-------------

Operación :  $(C) \leftarrow (C) \wedge (\text{bit})$

• ANL C,/bit

Bytes : 2  
Ciclos : 2

Código : 

1011	0000
------	------

bit address
-------------

Operación :  $(C) \leftarrow (C) \wedge (\text{bit})$

• CJNE <byte-de origen>,rel

Esta instrucción compara la magnitud de ambos operandos y salta si ambos no son iguales. La dirección de salto se obtiene sumando un desplazamiento (rel) que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción.

El carry se activa si el contenido de <byte-destino> es menor que el <byte-fuente>, de lo contrario será borrado.

Los dos primeros operandos disponen de cuatro modos de direccionamiento: el acumulador puede ser comparado con el contenido de cualquier posición accesible con direccionamiento directo o dato inmediato, y el contenido de cualquier posición de memoria accesible mediante direccionamiento indirecto o por registro, con un dato inmediato.

Ejemplo:

```

                CJNE R7,#60H,NO_IGUAL
;
; NO_IGUAL: JC     ES_MENOR           ; R7 = 60H
;                               ; SALTA SI R7 < 60H
;                               ; R7 > 60H
;

```

Ejemplo:

```

ESPERA:  CJNE A,P1,ESPERA

```

Si el dato presente en el Port 1 es igual al contenido del acumulador, el programa continúa con la siguiente instrucción; de lo contrario, buclea en la misma línea.

• **CJNE A,direct,rel**

Bytes : 3  
Ciclos : 2

Código : 

1011	0101	direct	rel
------	------	--------	-----

Operación : (PC) ← (PC) + 3  
SI (A) <> (direct)  
ENTONCES  
(PC) ← (PC) + relative offset  
SI (A) < (direct)  
ENTONCES  
(C) ← 1  
CONTRARIO  
(C) ← 0

• **CJNE A,#data,rel**

Bytes : 3  
Ciclos : 2

Código : 

1011	0100	data	rel
------	------	------	-----

Operación : (PC) ← (PC) + 3  
SI (A) <> #data  
ENTONCES  
(PC) ← (PC) + relative offset  
SI (A) < #data  
ENTONCES  
(C) ← 1  
CONTRARIO  
(C) ← 0

• **CJNE Rn,#data,rel**

Bytes : 3  
Ciclos : 2

Código : 

1011	lrrr	data	rel
------	------	------	-----

Operación : (PC) ← (PC) + 3  
SI (Rn) <> #data

ENTONCES  
 $(PC) \leftarrow (PC) + \text{relative offset}$   
 SI  $(Rn) < \# \text{data}$   
 ENTONCES  
 $(C) \leftarrow 1$   
 CONTRARIO  
 $(C) \leftarrow 0$

• **CJNE @Ri, #data,rel**

Bytes : 3  
 Ciclos : 2

Código : 

1011	011i	data	rel
------	------	------	-----

Operación :  $(PC) \leftarrow (PC) + 3$   
 $SI ((Ri)) < > \# \text{data}$   
 ENTONCES  
 $(PC) \leftarrow (PC) + \text{relative offset}$   
 $SI ((Ri)) < \# \text{data}$   
 ENTONCES  
 $(C) \leftarrow 1$   
 CONTRARIO  
 $(C) \leftarrow 0$

• **CLR A**

Todos los bits del acumulador se ponen a cero sin afectar a ningún señalizador (flag).

Bytes : 1  
 Ciclos : 1

Código : 

1111	0100
------	------

Operación :  $(A) \leftarrow 0$

• **CLR bit**

El bit indicado se pone a cero. Ningún otro flag se ve afectado. CLR puede operar con el carry y cualquier bit direccionable.

• CLR C

Bytes : 1  
Ciclos : 1

Código : 

1100	0011
------	------

Operación : (C) ← 0

• CLR bit

Bytes : 1  
Ciclos : 1

Código : 

1100	0010	bit address
------	------	-------------

Operación : (bit) ← 0

• CPL A

Complementa el contenido del acumulador. Cada bit del acumulador que esté a «1» se pondrá a «0», y a la inversa. Ningún flag queda afectado.

Bytes : 1  
Ciclos : 1

Código : 

1111	0100
------	------

Operación : (A) ←  $\neg$  (A)

• CPL bit

Complementa el bit especificado. Puede operar sobre el carry o sobre cualquier bit direccionable de forma directa.

Cuando usamos esta instrucción para modificar una patilla de salida de un puerto, la lectura no se realiza de la patilla, se realiza del latch de salida de datos.

• CPL C
---------

Complementa el indicador de Carry.

Bytes : 1  
Ciclos : 1

Código : 

1011	0011
------	------

Operación :  $(C) \leftarrow \neg (C)$

• CPL bit
-----------

Complementa el bit indicado.

Bytes : 2  
Ciclos : 1

Código : 

1011	0010	bit address
------	------	-------------

Operación :  $(\text{bit}) \leftarrow \neg (\text{bit})$

• DA A
--------

Ajuste a decimal del acumulador para la suma. DA A ajusta los 8 bits del acumulador tras la suma producida (en formato BCD) al sumar dos variables via instrucción ADD o ADDC.

Si el nibble bajo del acumulador contiene un número mayor que 9, o si está a 1 el auxiliar-carry, se le suma 6 al acumulador con objeto de obtener el dígito BCD apropiado en el nibble bajo. Esta suma interna puede llegar a activar el carry si la propagación que se produce hacia el nibble alto produce llevada, pero de ningún modo puede borrar el carry.

Si el carry está ahora activado, o si el nibble alto excede ahora de 9, este nibble alto se incrementa en 6 para obtener el dígito BCD apropiado en el nibble alto. De nuevo es posible que en esta operación interna de adición se active el carry, indicando que el resultado obtenido es mayor que 100, permitiendo precisión múltiple en la suma decimal. El flag de overflow OV no se ve afectado.

Todo esto ocurre en un único ciclo de instrucción. En esencia, esta instrucción realiza la conversión decimal sumando 00H, 06H, 60H o 66H al acumulador, dependiendo del estado inicial del acumulador y del PSW (Program State Word) o palabra de estado de programa.

DA A no permite la conversión de un número en hexadecimal que está en el acumulador a formato BCD. Además no debe emplearse DA A en la resta decimal.

**Ejemplo:** El acumulador tiene 56H representando al número 56 en BCD.  
R3 tiene 67H representando al número 67 en BCD.  
El carry está a 1.

La secuencia de instrucciones a ejecutar es:

```
ADDC A,R3
DA A
```

El resultado tras efectuar la suma es de 0BEH, y se borran tanto el carry-auxiliar como el carry.

El ajuste a decimal de dicho valor da como resultado 24H, que representa al número 24 en BCD, y el carry se pone a 1 indicando que el número es mayor que 100. Efectivamente,  $56 + 67 + 1 = 124$ .

Bytes : 1  
Ciclos : 1

Código : 

1101	0111
------	------

Operación : — El número en el acumulador está en BCD.  
SI  $[(A3-0) > 9 \vee [(AC) = 1]]$   
ENTONCES  $(A3-0) \leftarrow (3-0) + 6$   
Y SI  $[A7-4] > 9 \vee [(C) = 1]$   
ENTONCES  $(A7-4) \leftarrow (A7-4) + 6$

• DEC byte

Decrementa en una unidad la variable indicada. Si el valor a decrementar es 00H, entonces el resultado será 0FFH. Ningún flag se verá afectado. Permite cuatro formas de direccionamiento: acumulador, registro, directo o indirecto por registro.

Cuando se utiliza esta instrucción para modificar la salida de un puerto, el valor utilizado para decrementar no se lee de las patillas, se lee del latch de salida.

• DEC A

Bytes : 1  
Ciclos : 1

Código : 

0001	0100
------	------

Operación :  $(A) \leftarrow (A) - 1$

• DEC Rn

Bytes : 1  
Ciclos : 1

Código : 

0001	1rrr
------	------

Operación :  $(Rn) \leftarrow (Rn) - 1$

• DEC direct

Bytes : 2  
Ciclos : 1

Código : 

0001	0101	direct
------	------	--------

Operación :  $(\text{direct}) \leftarrow (\text{direct}) - 1$

• DEC @Ri

Bytes : 1  
Ciclos : 1

Código : 

0001	011i
------	------

Operación :  $((Ri)) \leftarrow ((Ri)) - 1$

• DIV AB

Divide un entero sin signo de 8 bits situado en el acumulador, entre un entero sin signo de 8 bits situado en el registro B. Tras efectuarse la división, el acumulador recibirá la parte entera de la división, y el registro B recibirá el resto resultante de la división. Los flags carry y OVERflow se borran.

Si el registro B tuviera 00H, el resultado obtenido en el acumulador y el registro B sería indeterminado y se activaría el flag OV. El carry sería igualmente borrado.

Ejemplo: El acumulador tiene 251 (0FBH).

El registro B tiene 18 (12H).

Tras ejecutar DIV AB quedarían con:

El acumulador: 13 (0DH).

El registro B: 17 (11H).

Dado que:  $251 = 13 \times 18 + 17$ . CY y OV borrados.

Bytes : 1  
Ciclos : 4

Código : 

1000	0100
------	------

Operación :  $(A)_{15-8} \leftarrow \text{Parte entera de } (A) / (B)$   
 $(B)_{7-0} \leftarrow \text{Resto de } (A) / (B)$

• **DJNZ <byte>,rel**

Decrementa y Jump (salta) si No Zero. Esta instrucción decrementa en una unidad la variable indicada y salta a la dirección indicada por el segundo operando si el valor resultante del decremento es diferente de cero. Si el valor fuera 00H, la variable pasaría a valer 0FFH. Ningún flag se ve afectado.

La dirección de salto se obtiene sumando un desplazamiento (rel) que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción.

La variable decrementada puede ser: registro o direct.

Cuando se utiliza esta instrucción para modificar la salida de un puerto, el valor utilizado para la modificación no se lee de las patillas, se lee del latch de salida.

**Ejemplo:** Ejecución de un bucle un número determinado de veces.

```

MOV R2,#8
CAMBIA: CPL P1.7
        DJNZ R2,CAMBIA
    
```

Estas líneas de código hacen que por la patilla 7 del puerto 1 aparezcan cuatro pulsos.

• **DJNZ Rn,rel**

Bytes : 2  
Ciclos : 2

Código : 

1101	1rrr	rel
------	------	-----

Operación :  $(PC) \leftarrow (PC) + 2$   
 $(Rn) \leftarrow (Rn) - 1$   
 SI  $(Rn) > 0$  O  $(Rn) < 0$   
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

• **DJNZ direct,rel**

Bytes : 3  
Ciclos : 2

Código : 

1101	0101	direct	rel
------	------	--------	-----

Operación :  $(PC) \leftarrow (PC) + 2$   
 $(direct) \leftarrow (direct) - 1$   
 SI  $(direct) > 0$  O  $(direct) < 0$   
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

**• INC <byte>**

INC incrementa en una unidad la variable indicada. Si ésta contiene 0FFH, pasará a valer 00H. Ningún flag será afectado.

Los modos de direccionamiento son: registro, directo, o indirecto por registro.

**• INC A**

Bytes : 1  
Ciclos : 1

Código : 

0000	0100
------	------

Operación :  $(A) \leftarrow (A) + 1$

**• INC Rn**

Bytes : 1  
Ciclos : 1

Código : 

0000	1rrr
------	------

Operación :  $(Rn) \leftarrow (Rn) + 1$

**• INC direct**

Bytes : 1  
Ciclos : 1

Código : 

0000	0101	direct
------	------	--------

Operación :  $(\text{direct}) \leftarrow (\text{direct}) + 1$

**• INC @Ri**

Bytes : 1  
Ciclos : 1

Código : 

0000	011i
------	------

Operación :  $((Ri)) \leftarrow ((Ri)) + 1$

### • INC DPTR

Incrementa en una unidad el apuntador de 16 bits DPTR. Un desbordamiento en la parte baja DPL de 0FFH a 00H hará que la parte alta DPH se incremente en una unidad. Ningún flag se verá afectado.

Bytes : 1  
Ciclos : 2

Código : 

1010	0011
------	------

Operación :  $(DPTR) \leftarrow (DPTR) + 1$

### • JB bit,rel

Salta si el bit indicado está a 1, de lo contrario continuará la ejecución en la siguiente instrucción. La dirección de salto se obtiene sumando un desplazamiento (rel), que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción.

No se modifican los flags ni el bit en cuestión.

Bytes : 3  
Ciclos : 2

Código : 

0010	0000	bit address	rel
------	------	-------------	-----

Operación :  $(PC) \leftarrow (PC) + 3$   
SI (bit) = 1  
ENTONCES  
 $(PC) \leftarrow (PC) + rel$

### • JBC bit,rel

Salta si el bit indicado está a 1 y lo borra, de lo contrario continuará la ejecución en la siguiente instrucción. La dirección de salto se obtiene sumando un desplazamiento (rel), que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción.

No se modifican los flags ni el bit en cuestión.

Cuando se utiliza esta instrucción para modificar la salida de un puerto, el valor utilizado para la modificación no se lee de las patillas, se lee del latch de salida.

Bytes : 3  
Ciclos : 2

Código : 

0001	0000	bit address	rel
------	------	-------------	-----

Operación :  $(PC) \leftarrow (PC) + 3$   
 SI (bit) = 1  
 ENTONCES  
 (bit)  $\leftarrow$  0  
 (PC)  $\leftarrow$  (PC) + rel

• **JC rel**

Salta si el flag carry está activado, de lo contrario continuará la ejecución en la siguiente instrucción. La dirección de salto se obtiene sumando un desplazamiento (rel) que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción.

No se modifican los flags.

Cuando se utiliza esta instrucción para modificar la salida de un puerto, el valor utilizado para la modificación no se lee de las patillas, se lee del latch de salida.

Bytes : 3  
 Ciclos : 2

Código : 

0001	0000	bit address	rel
------	------	-------------	-----

Operación :  $(PC) \leftarrow (PC) + 3$   
 SI (C) = 1  
 ENTONCES  
 (PC)  $\leftarrow$  (PC) + rel

• **JMP @A + DPTR**

Salto indirecto. Suma el entero sin signo de 8 bits contenido en el acumulador con los 16 bits del DPTR y coloca su resultado en el PC. Esta instrucción suele utilizarse para bifurcación múltiple.

No se modifican los flags, ni el DPTR, ni el acumulador.

Ejemplo: Debe cargarse un número entre 0, 2, 4 y 6 en el acumulador.

La siguiente secuencia de instrucciones desviará la ejecución del programa a una de las cuatro etiquetas de la tabla de saltos.

```

MOV DPTR, # TABLA_SALTOS
JMP @A + DPTR
TABLA_SALTOS: AJMP ETIQUETA_1
               AJMP ETIQUETA_2
               AJMP ETIQUETA_3
               AJMP ETIQUETA_4
  
```

ETIQUETA\_1: .  
 .  
 .  
 ETIQUETA\_2: .  
 .  
 .  
 ETIQUETA\_3: .  
 .  
 .  
 ETIQUETA\_4: .  
 .  
 .

Bytes : 1  
 Ciclos : 2

Código : 

0111	0011
------	------

Operación :  $(PC) \leftarrow (A) + (DPTR)$

• **JNB bit,rel**

Salto relativo si el bit «no está» activado, de lo contrario se continúa con la siguiente instrucción. Ni el bit comprobado ni los flags resultan afectados.

Bytes : 3  
 Ciclos : 2

Código : 

0011	0000
------	------

bit address
-------------

rel
-----

Operación :  $(PC) \leftarrow (PC) + 3$   
 SI (bit) = 0  
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

• **JNC rel**

Salto relativo si el carry «no está» activado, de lo contrario se continúa con la siguiente instrucción. Ni el bit comprobado ni los flags resultan afectados.

Bytes : 2  
 Ciclos : 2

Código : 

0101	0000	rel
------	------	-----

Operación :  $(PC) \leftarrow (PC) + 2$   
 SI (C) = 0  
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

• **JNZ rel**

Salto relativo si alguno de los bits del acumulador no está a cero, de lo contrario se continúa con la siguiente instrucción. Ni el bit comprobado ni los flags resultan afectados.

Bytes : 2  
 Ciclos : 2

Código : 

0011	0000	rel
------	------	-----

Operación :  $(PC) \leftarrow (PC) + 2$   
 SI (A)  $\neq$  0  
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

• **JZ bit,rel**

Salto relativo si todos los bits del acumulador están a cero, de lo contrario se continúa con la siguiente instrucción. Ni el bit comprobado ni los flags resultan afectados.

Bytes : 2  
 Ciclos : 2

Código : 

0011	0000	rel
------	------	-----

Operación :  $(PC) \leftarrow (PC) + 2$   
 SI (A) = 0  
 ENTONCES  
 $(PC) \leftarrow (PC) + rel$

• **LCALL addr16**

Llamada a un subprograma situado en la dirección indicada de 16 bits. Esta instrucción suma 3 al contenido del PC y lo guarda en la PILA; a continuación se carga la nueva dirección en el PC para

que el microcontrolador ejecute el subprograma hasta encontrar una instrucción de retorno que coloque la dirección guardada en la PILA de nuevo en el PC. La llamada puede realizarse a cualquier dirección de los 64 K de memoria de programa. Ningún flag se verá afectado.

Bytes : 3  
Ciclos : 2

Código : 

0001	0010	addr 15-8	addr 7-0
------	------	-----------	----------

Operación :  
 $(PC) \leftarrow (PC) + 3$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC7-0)$   
 $(SP) \leftarrow (SP) + 1$   
 $((SP)) \leftarrow (PC15-8)$   
 $(PC) \leftarrow \text{addr}16$

• **LJMP addr16**

Salto incondicional a la dirección de 16 bits indicada. El salto puede realizarse a cualquier dirección de los 64 K de memoria de programa. No se verá afectado ningún flag.

Bytes : 3  
Ciclos : 2

Código : 

0000	0010	addr 15-8	addr 7-0
------	------	-----------	----------

Operación :  $(PC) \leftarrow \text{addr}16$

• **MOV <byte-destino>, <byte-fuente>**

Mueve (copia) el contenido del byte fuente hacia el byte destino. El byte fuente no se modifica, así como los flags.

Esta es la operación de mayor flexibilidad, soportando 15 combinaciones diferentes de direccionamiento entre fuente y destino.

**Ejemplo:** La RAM interna 30H tiene 40H.  
 La RAM interna 40H tiene 10H.  
 El dato presente en el Puerto 1 es 0CAH.

```

MOV R0, # 30H ; R0 ← 30H
MOV A, @R0    ; A ← 40H
MOV R1, A     ; R1 ← 40H
MOV B, @R1    ; B ← 10H
MOV @R1, P1   ; RAM (40H) ← 0CAH
MOV P2, P1    ; (P2) ← (P1)
    
```

**• MOV A,Rn**

Bytes : 1  
 Ciclos : 1  
 Código : 

1110	1rrr
------	------

  
 Operación : (A) ← (Rn)

**• MOV A,direct**

MOV A,ACC no es una instrucción válida.

Bytes : 2  
 Ciclos : 1  
 Código : 

0000	0010
------	------

direct
--------

  
 Operación : (A) ← (direct)

**• MOV A,@Ri**

Bytes : 1  
 Ciclos : 1  
 Código : 

1110	011i
------	------

  
 Operación : (A) ← ((Ri))

**• MOV A,#data**

Bytes : 2  
 Ciclos : 1  
 Código : 

0111	0100
------	------

data
------

  
 Operación : (A) ← #data

**• MOV Rn,A**

Bytes : 1  
 Ciclos : 1  
 Código : 

1111	1rrr
------	------

  
 Operación : (Rn) ← (A)

**• MOV Rn,direct**

Bytes : 2  
 Ciclos : 2  
 Código : 

1010	1rrr
------	------

direct
--------

  
 Operación : (Rn) ← (direct)

**• MOV Rn,#data**

Bytes : 2  
 Ciclos : 1  
 Código : 

0111	1rrr
------	------

data
------

  
 Operación : (Rn) ← #data

**• MOV direct,A**

Bytes : 2  
 Ciclos : 1  
 Código : 

1111	0101
------	------

direct
--------

  
 Operación : (direct) ← (A)

**• MOV direct,Rn**

Bytes : 2  
 Ciclos : 2

Código : 

1000	1rrr	direct
------	------	--------

Operación : (direct) ← (Rn)

• MOV direct, direct

Bytes : 3

Ciclos : 2

Código : 

1000	0101	direct destino	direct fuente
------	------	----------------	---------------

Operación : (direct) ← (direct)

• MOV direct, @Ri

Bytes : 2

Ciclos : 2

Código : 

1000	011i	direct
------	------	--------

Operación : (direct) ← ((Ri))

• MOV direct, # data

Bytes : 3

Ciclos : 2

Código : 

0111	0101	direct	data
------	------	--------	------

Operación : (direct) ← # data

• MOV @Ri, A

Bytes : 1

Ciclos : 1

Código : 

1111	011i
------	------

Operación : ((Ri)) ← (A)

• **MOV @Ri,direct**

Bytes : 2  
 Ciclos : 2  
 Código : 

1010	011i	direct
------	------	--------

  
 Operación : ((Ri)) ← (direct)

• **MOV @Ri,#data**

Bytes : 2  
 Ciclos : 1  
 Código : 

0111	011i	data
------	------	------

  
 Operación : ((Ri)) ← #data

• **MOV <bit-destino>, <bit-fuente>**

Mueve (copia) el contenido del bit fuente hacia el bit destino. Ni el bit comprobado ni los flags resultan afectados.

Ejemplo: MOV P1.3,C ; P1.3 ← Carry  
 MOV C,P3.3 ; Carry ← P3.3

• **MOV C,bit**

Bytes : 2  
 Ciclos : 1  
 Código : 

1010	0010	bit address
------	------	-------------

  
 Operación : (C) ← (bit)

• **MOV bit,C**

Bytes : 2  
 Ciclos : 2

Código : 

1001	0010	bit address
------	------	-------------

Operación : (bit) ← (C)

**• MOV DPTR, #data16**

Carga el apuntador a la memoria de datos DPTR con un dato de 16 bits. La parte alta en DPH, y la baja en DPL.

Bytes : 3  
Ciclos : 2

Código : 

1001	0000	data 15-8	data 7-0
------	------	-----------	----------

Operación : (DPTR) ← #data16  
o DPH & DPL ← #data 15-8 & #data 7-0

**• MOVC A,@A + <base-reg>**

Las instrucciones MOVC se emplean para transferir al acumulador un byte de la memoria de código (indicado por la «C» de Code). La dirección del byte viene dada por la suma del contenido del acumulador con el doble registro base-reg que puede ser el PC o el DPTR. El PC es incrementado para apuntar a la siguiente instrucción antes de que se realice la suma con el contenido del acumulador. El base-reg o registro base no se altera nunca. Los flags no resultan afectados.

Ejemplo: Un valor comprendido entre 0 y 3 cargado en el acumulador permitirá realizar la lectura de uno de los cuatro valores definidos con la directiva de ensamblado DB.

```
REL_PC: INC A
        MOVC A,@A + PC
        RET
```

```
DB 66H,77H,88H,99H
```

**• MOVC A,@A + DPTR**

Bytes : 1  
Ciclos : 1

Código : 

1001	0011
------	------

Operación : (A) ← ((A) + DPTR))

• **MOVC A,@A+PC**

Bytes : 1  
 Ciclos : 1  
 Código : 

1000	0011
------	------

  
 Operación :  $(PC) \leftarrow (PC) + 1$   
 $(A) \leftarrow ((A) + (PC))$

• **MOVX <byte-destino>, <byte-fuente>**

Las instrucciones MOVX permiten el intercambio de datos entre el acumulador y la memoria externa de datos (indicado por la «x» de external).

De los dos operandos, uno siempre es el acumulador, el otro operando es @PTR o @Ri. Cuando un operando es @DPTR, los registros DPH y DPL proporcionan una dirección de 16 bits para acceder a los 64 Kbytes de memoria externa.

Cuando un operando es @Ri, el registro Ri proporciona una dirección de ocho bits para acceder a 256 posiciones de memoria externa, en este caso el puerto 2 saca el contenido del registro P2. (Direccionamiento paginado.)

En algunas situaciones es posible la mezcla de los dos tipos de instrucciones MOVX. Un array muy largo en RAM con la parte alta de la dirección controlada por P2 puede ser direccionado mediante el DPTR, o escribiendo en P2 la parte alta de la dirección y a continuación mediante instrucciones MOVX que utilicen R0 o R1.

**Ejemplo:** Una RAM externa de 256 bytes direccionada empleando las líneas de direcciones/datos multiplexados (por ejemplo un Intel 8155 RAM,I/O, Timer conectado a P0). El puerto P3 proporciona las líneas de control para memoria externa. P1 y P2 se utilizan como líneas de E/S. R0 y R1 contienen 12H y 34H, respectivamente. La posición 34H de RAM externa contiene el dato 56H.

```
MOVX A,@R1
MOVX @R0,A
```

La ejecución de las instrucciones anteriores habrá realizado la copia del dato 56H en el acumulador y en la dirección 12H de RAM externa.

• **MOVX A,@Ri**

Bytes : 1  
 Ciclos : 2

Código : 

1110	001i
------	------

Operación : (A) ← ((Ri))

**• MOVX A,@DPTR**

Bytes : 1

Ciclos : 2

Código : 

1110	0000
------	------

Operación : (A) ← ((DPTR))

**• MOVX @Ri,A**

Bytes : 1

Ciclos : 2

Código : 

1111	001i
------	------

Operación : ((Ri)) ← (A)

**• MOVX @DPTR,A**

Bytes : 1

Ciclos : 2

Código : 

1111	0000
------	------

Operación : ((DPTR)) ← (A)

**• MUL AB**

Multiplica un entero sin signo de 8 bits situado en el acumulador por otro situado en el registro B. El byte bajo del resultado se deja en el acumulador, y el byte alto en el registro B. Si el producto supera el valor 255 (0FFH), el flag de OVerflow se activa; de lo contrario se borra. El carry siempre se borra.

Ejemplo: El acumulador tiene 80 (50H).  
El registro B tiene 160 (0A0H).

MUL AB

Obtenemos un producto de 12.800 (3200H), con 32H cargado en el registro B, y 00H en el acumulador. El flag OVerflow está activado y el carry borrado.

Bytes : 1

Ciclos : 4

Código : 

1010	0100
------	------

Operación : (A7-0) ← Parte baja de (A) × (B)  
(A15-8) ← Parte alta de (A) × (B)

• **NOP**

No OPeración. El programa continúa en la siguiente instrucción, modificando tan sólo el PC.

Ejemplo: Vamos a generar un pulso de valor «0» en el bit 7 del puerto 2 durante cinco ciclos exactamente. Una única secuencia SETB/CLR nos generaría un pulso de sólo un ciclo de duración; para añadir los otros cuatro ciclos adicionales deberemos insertar cuatro instrucciones NOP. Se supone que no están permitidas las interrupciones ya que, en el caso de producirse alguna, el número de ciclos podría aumentar considerablemente mientras se atiende dicha interrupción.

```
CLR P2.7
NOP
NOP
NOP
NOP
SETB P2.7
```

Bytes : 1

Ciclos : 1

Código : 

0000	0000
------	------

Operación : (PC) ← (PC) + 1

• **ORL <byte-destino>, <byte-fuente>**

La instrucción ORL proporciona la «operación lógica» OR entre las dos variables, dejando el resultado de la operación en byte-destino. Ningún flag se verá afectado.

Los dos operandos pueden operar con seis modos diferentes de direccionamiento. Cuando el

destino es el acumulador, la fuente puede emplear direccionamiento directo, por registro, indirecto por registro o dato inmediato. Cuando el destino es una dirección la fuente puede ser un dato inmediato o el acumulador.

Ejemplo: El acumulador tiene 0C3H y R0 55H. Después de ejecutar:

ORL A,R0

El acumulador ahora tiene 0D7H.

• ORL A,Rn

Bytes : 1  
Ciclos : 1

Código : 

0100	1rrr
------	------

Operación :  $(A) \leftarrow (A) \vee (Rn)$

• ORL A,direct

Bytes : 2  
Ciclos : 1

Código : 

0100	0101	direct
------	------	--------

Operación :  $(A) \leftarrow (A) \vee (\text{direct})$

• ORL A,@Ri

Bytes : 1  
Ciclos : 1

Código : 

0100	011i
------	------

Operación :  $(A) \leftarrow (A) \vee ((Ri))$

• ORL A,#data

Bytes : 2  
Ciclos : 1

Código : 

1000	1rrr	data
------	------	------

  
 Operación :  $(A) \leftarrow (A) \vee \#data$

• **ORL direct,A**

Bytes : 2  
 Ciclos : 1  
 Código : 

0100	0010	direct
------	------	--------

  
 Operación :  $(direct) \leftarrow (direct) \vee (A)$

• **ORL direct,#data**

Bytes : 3  
 Ciclos : 2  
 Código : 

0100	0011	direct	data
------	------	--------	------

  
 Operación :  $(direct) \leftarrow (direct) \vee \#data$

• **ORL <A>,<bit-fuente>**

Activa el carry si el bit-fuente vale «1», de lo contrario lo deja como estaba. Si el bit-fuente va precedido del carácter «/», entonces se emplea el valor complementario del bit-fuente, pero éste no se verá afectado. No se verá afectado ningún otro flag.

Ejemplos: 

MOV	C,P1.0	;	Carry $\leftarrow$ P1.0
ORL	C,ACC.7	;	Carry $\leftarrow$ Carry $\vee$ Acc.7
ORL	C,/OV	;	Carry $\leftarrow$ Carry $\vee$ /OVerflow

• **ORL C,bit**

Bytes : 2  
 Ciclos : 2  
 Código : 

0111	0010	bit address
------	------	-------------

  
 Operación :  $(C) \leftarrow (C) \vee (bit)$

• **ORL C,/bit**

Bytes : 2  
Ciclos : 2

Código : 

1010	0000	bit address
------	------	-------------

Operación :  $(C) \leftarrow (C) \vee (\overline{\text{bit}})$

• **POP direct**

Se lee el contenido de una dirección de RAM interna apuntada por el Stack Pointer, y éste se decrementa en una posición. El valor leído se transfiere a la dirección indicada (direct). No se verá afectado ningún otro flag.

Ejemplo: El SP tiene el valor 32H, y la posición de RAM interna 30H hasta la 32H contienen los valores 20H, 23H y 01H, respectivamente.

POP DPH ; DPH  $\leftarrow$  01H y SP  $\leftarrow$  31H  
POP DPL ; DPL  $\leftarrow$  23H y SP  $\leftarrow$  30H  
POP SP ; SP  $\leftarrow$  20H

Bytes : 2  
Ciclos : 2

Código : 

1101	0000	direct
------	------	--------

Operación :  $(\text{direct}) \leftarrow ((\text{SP}))$   
 $(\text{SP}) \leftarrow (\text{SP}) - 1$

• **PUSH direct**

El SP se incrementa en una unidad y el contenido de la dirección indicada (direct) se copia en la posición de RAM apuntada por el SP. No se verá afectado ningún otro flag.

Ejemplo: El SP contiene 09H. El DPTR tiene 0123H.

PUSH DPL ; SP  $\leftarrow$  0AH y (0AH)  $\leftarrow$  01H  
PUSH DPH ; SP  $\leftarrow$  0BH y (0BH)  $\leftarrow$  23H

Bytes : 2  
Ciclos : 2

Código : 

1100	0000
------	------

direct
--------

Operación : (SP) ← (SP) + 1  
((SP)) ← (direct)

• RET

Retorno de subprograma. RET saca de la PILA (POP) el byte alto y el byte bajo que colocará en el PC, decrementando el SP en 2 unidades. La ejecución del programa continúa en dicha dirección, que por lo general es la correspondiente a la dirección siguiente a la instrucción ACALL o LCALL. No se verá afectado ningún otro flag.

Bytes : 1  
Ciclos : 2

Código : 

0010	0010
------	------

Operación : (PC 15-8) ← ((SP))  
(SP) ← (SP) - 1  
(PC 7-0) ← ((SP))  
(SP) ← (SP) - 1

• RETI

Retorno de Interrupción. RET saca de la PILA (POP) el byte alto y el byte bajo que colocará en el PC, decrementando el SP en dos unidades. Inicializa la lógica de atención a las interrupciones con objeto de que puedan ser atendidas otras interrupciones con el mismo o inferior nivel de prioridad que la que ha procesado.

La ejecución del programa continúa en la dirección obtenida de la PILA, que por lo general corresponde a la instrucción inmediatamente a continuación en que la petición de interrupción fue detectada y atendida.

Si una interrupción del mismo nivel o inferior se produce mientras la instrucción RETI está en proceso, se ejecutará otra instrucción antes de que la interrupción pendiente sea aceptada.

Bytes : 1  
Ciclos : 2

Código : 

0011	0010
------	------

Operación : (PC 15-8) ← ((SP))  
(SP) ← (SP) - 1  
(PC 7-0) ← ((SP))  
(SP) ← (SP) - 1

• RLA

Los ocho bits del acumulador serán rotados un bit a la izquierda (Left). El bit 7 será rotado sobre el bit 0. No se verá afectado ningún otro flag.

Bytes : 1  
Ciclos : 1

Código : 

0010	0011
------	------

Operación :  $(A.n + 1) \leftarrow (A.n) \quad n = 0 \rightarrow 6$   
 $(A.0) \leftarrow (A.7)$

• RLC A

Los ocho bits del acumulador serán rotados un bit a la izquierda (Left). El carry será rotado sobre el bit 0, y el bit 7 sobre el carry. No se verá afectado ningún otro flag.

Bytes : 1  
Ciclos : 1

Código : 

0011	0011
------	------

Operación :  $(A.n + 1) \leftarrow (A.n) \quad n = 0 \rightarrow 6$   
 $(A.0) \leftarrow (C)$   
 $(C) \leftarrow (A.7)$

• RRA

Los ocho bits del acumulador serán rotados un bit a la derecha (Right). El bit 0 será rotado sobre el bit 7. No se verá afectado ningún otro flag.

Bytes : 1  
Ciclos : 1

Código : 

0000	0011
------	------

Operación :  $(A.n) \leftarrow (A.n + 1) \quad n = 0 \rightarrow 6$   
 $(A.7) \leftarrow (A.0)$

• RRC A

Los ocho bits del acumulador serán rotados un bit a la derecha (Right). El bit carry será rotado sobre el bit 7, y el bit 0 sobre el carry. No se verá afectado ningún otro flag..

Bytes : 1  
 Ciclos : 1  
 Código : 

0011	0011
------	------

  
 Operación :  $(A.n) \leftarrow (A.n + 1) \quad n = 0 \rightarrow 6$   
 $(A.7 \leftarrow (C)$   
 $(C \leftarrow (A.0)$

**• SETB <bit>**

Pone a 1 el bit indicado. Este bit puede ser el carry o cualquier bit direccionable.

**• SETB C**

Bytes : 1  
 Ciclos : 1  
 Código : 

1101	0011
------	------

  
 Operación :  $(C) \leftarrow 1$

**• SETB bit**

Bytes : 2  
 Ciclos : 1  
 Código : 

1101	0010	bit address
------	------	-------------

  
 Operación :  $(\text{bit}) \leftarrow 1$

**• SJMP rel**

Realiza un salto relativo incondicional. La dirección de salto se obtiene sumando un desplazamiento (rel), que puede ser positivo o negativo (indicado en complemento a dos), a la dirección que tendrá el PC en la siguiente instrucción. De este modo puede realizarse un salto 128 bytes antes y 127 después de la instrucción SJMP.

Bytes : 2  
 Ciclos : 2  
 Código : 

1000	0000	rel
------	------	-----

  
 Operación :  $(PC) \leftarrow (PC) + 2$   
 $(PC) \leftarrow (PC) + rel$

• SUBB A, <byte-fuente>

Resta con carry negativo (Borrow). SUBB resta la variable y el carry del acumulador, dejando el resultado en el acumulador. SUBB activa el carry (Borrow), si se necesita un Borrow para el bit 7; de lo contrario borrará el carry. (Si el carry estaba activado antes de ejecutar la instrucción SUBB, esto indica que se necesitó un Borrow para el paso previo en una resta de precisión múltiple, de forma que el carry se restará del acumulador junto con el byte-fuente.) El auxiliar-carry se activará en el caso de necesitar un Borrow para el bit 7, de lo contrario se borrará. El flag OVERflow se activará si se necesitara un Borrow para el bit 6 pero no para el 7, o para el 7 pero no para el 6.

Cuando restamos enteros con signo, el flag OV indica resultados negativos cuando se resta un número negativo de otro positivo, o un resultado positivo cuando se resta un número positivo de otro negativo.

Ejemplo: El acumulador tiene 0C9H, R2 tiene 54H y el carry = 1.

SUBB A,R2

Una vez ejecutada la instrucción, el acumulador tendrá 74H, los flags C, y AC borrados, y el flag OV activado.

Dado que 0C9H menos 54H es 75H. La diferencia entre este valor y el resultado obtenido se debe al carry (Borrow), activado antes de haberse ejecutado la instrucción SUBB. Si no conociéramos el valor del carry antes de comenzar una resta en simple o múltiple precisión, debe borrarse con la instrucción CLR C.

• SUBB A,Rn

Bytes : 1  
 Ciclos : 1  
 Código : 

1001	1rrr
------	------

  
 Operación :  $(A) \leftarrow (A) - (C) - (Rn)$

• SUBB A,direct

Bytes : 2  
 Ciclos : 1

Código : 

1001	0101	direct
------	------	--------

  
 Operación :  $(A) \leftarrow (A) - (C) - (\text{direct})$

• **SUBB A,@Ri**

Bytes : 1  
 Ciclos : 1  
 Código : 

1001	011i
------	------

  
 Operación :  $(A) \leftarrow (A) - (C) - ((Ri))$

• **SUBB A,#data**

Bytes : 2  
 Ciclos : 1  
 Código : 

1001	0100	data
------	------	------

  
 Operación :  $(A) \leftarrow (A) - (C) - \#data$

• **SWAP A**

Intercambio de los nibbles del acumulador. Esta instrucción sustituye a cuatro instrucciones consecutivas de rotación.

Ejemplo: El acumulador tiene el valor 0C5H.

SWAP A

Ahora tiene el valor 5CH.

Bytes : 1  
 Ciclos : 1  
 Código : 

1100	0100
------	------

  
 Operación :  $(A\ 3-0) \leftrightarrow (A\ 7-4)$

• XCH A, < byte >

XCH carga en el acumulador el contenido de la variable indicada, y al mismo tiempo escribe el contenido original del acumulador en la variable indicada. Tenemos direccionamiento por registro, directo, e indirecto por registro.

Ejemplo: El acumulador tiene 3FH.  
El registro R0 contiene el valor 20H.  
La RAM 20H contiene el valor 75H.

```
XCH A, @R0
```

Esta instrucción nos pondrá en el acumulador el valor 75H, y el valor 3FH en la dirección de RAM 20H.

• XCH A, Rn

Bytes : 1  
Ciclos : 1  
Código : 

1100	1rrr
------	------

  
Operación : (A) ↔ (Rn)

• XCH A, direct

Bytes : 2  
Ciclos : 1  
Código : 

1100	0100	direct
------	------	--------

  
Operación : (A) ↔ (direct)

• XCH A, @Ri

Bytes : 1  
Ciclos : 1  
Código : 

1100	011i
------	------

  
Operación : (A) ↔ ((Ri))

• XCHD A,@Ri

XCHD intercambia el nibble bajo del acumulador (bits 3-0), que generalmente contendrá un dígito hexadecimal o BCD, con el nibble bajo de una posición de RAM interna apuntada por el registro Ri. Ni los nibbles altos (bits 7-4) de cada registro ni los flags se verán afectados.

Bytes : 1  
Ciclos : 1

Código : 

1101	011i
------	------

Operación : (A 3-0) ↔ ((Ri 3-0))

• XRL <byte-destino>,<byte-fuente>

La instrucción XRL proporciona la «operación lógica» XRL entre las dos variables, dejando el resultado de la operación en byte-destino. Ningún flag se verá afectado.

Los dos operandos pueden operar con seis modos diferentes de direccionamiento. Cuando el destino es el acumulador, la fuente puede emplear direccionamiento directo, por registro, indirecto por registro o dato inmediato. Cuando el destino es una dirección la fuente puede ser un dato inmediato o el acumulador.

• XRL A,Rn

Bytes : 1  
Ciclos : 1

Código : 

0110	1rrr
------	------

Operación : (A) ← (A) ✕ (Rn)

• XRL A,direct

Bytes : 2  
Ciclos : 1

Código : 

0110	0101	direct
------	------	--------

Operación : (A) ← (A) ✕ (direct)

• XRL A,@Ri

Bytes : 1  
Ciclos : 1

Código : 

0110	011i
------	------

Operación :  $(A) \leftarrow (A) \oplus ((Ri))$

• XRL A,#data

Bytes : 2  
Ciclos : 1

Código : 

1000	1rrr	data
------	------	------

Operación :  $(A) \leftarrow (A) \oplus \#data$

• XRL direct,A

Bytes : 2  
Ciclos : 1

Código : 

0110	0010	direct
------	------	--------

Operación :  $(direct) \leftarrow (direct) \oplus (A)$

• XRL direct,#data

Bytes : 3  
Ciclos : 2

Código : 

0110	0011	direct	data
------	------	--------	------

Operación :  $(direct) \leftarrow (direct) \oplus \#data$

## **ANEXO E**

### **PLANO DEL MODULO DE PRACTICAS**

## **BIBLIOGRAFÍA**

- [1] Boletín AB4-4, Vacuum Furnances Technology Instrumentation, Honeywell 1970
- [2] James W. Russell L. Pimmel. Sistemas Digitales Basados en Microprocesador. Mac Graw Hill 1983
- [3] Microcontroller User's Guide, Signetics 1989.
- [4] Embedded Control Applications Handbook, INTEL 1990
- [5] Manuel Torres P., Microprocesadores y Microcontroladores Aplicados a la Ind., Paraninfo. 1991
- [6] Alejandro Vega Salinas, Manual de Microcontrolador 8051, CINVESTAV IPN. 1993
- [7] Anibal Ollero Baturone ,Control por Computadora ,Alfaomega Marcombo 1991
- [8] Karl J.Astrom , Bjrn Wittenmark ,Sistemas de Controlados por Computador Paraninfo 1988
- [9] Julio Astigarraga Urquiza, Hornos Industriales de resistencias , McGRW-HILL, 1994
- [10] John P.Bentley ,Sistemas de medición , Compañía Editorial ,S.A. De C,V Mexico , 1993