

# *Contenido*

1

**Editorial**

---

**Como Direccionar Más Allá del primer MegaByte en el Modo Real de las PC**

M. en C. Gustavo A. Mas Levario

---

3

11

**Diseño, Simulación y Construcción de un Control PID Aplicado a un Sistema Térmico**

M. en C. Romeo Urbietta Parrazales

---

**Control Remoto de Formato Comercial**

Ing. Aquilino Cervantes Ávila

---

20

26

**Sistema de Control de Lógica Difusa Aplicado a un Motor de C. D.**

M. en C. Romeo Urbietta Parrazales

---

**Generador de Patrones de Televisión**

José A. Arias Aguilar, Osvaldo Espinoza Sosa

Alumnos de la Maestría del CINTEC

M. en C. Juan Carlos Gonzalez Robles

---

30

36

**Reporte de Actividades Realizadas en el CINTEC Durante el Período del 31 de Agosto de 1994 al 1o. de Septiembre de 1995**

---

---

## *Editorial*

**L**a revista polibits, desde sus inicios, ha tratado de cumplir con el principio de constituirse en un órgano de difusión de los resultados que se han logrado por la Investigación Científica y el Desarrollo Tecnológico realizado en el Instituto en materia de Cómputo. Como tal, sus inicios se dieron como órgano de difusión del CINTEC, y en los últimos números se ha iniciado el salto para la inclusión de participaciones externas. Sin embargo, consideramos en el CINTEC que el esfuerzo de divulgación realizado debe ampliar sus horizontes hacia las nuevas formas de comunicación y conectividad existentes.

Así como en otras áreas geográficas (Estados Unidos, Europa, Japón y Taiwan, principalmente) la interconectividad a través de la red Internet está instalada firmemente desde hace casi 4 años, en México y Latinoamérica se vive desde 1994 una expansión impresionante de este servicio, conectando tanto Instituciones Académicas como empresas Comerciales, Dependencias Gubernamentales, usuarios particulares, etc. Esta creciente conectividad no puede ser soslayada y debe ser aprovechada como un nuevo medio de divulgación a nuestro alcance.

El CINTEC, consciente de esta situación y de las ventajas de vinculación que este servicio ofrece, esta trabajando actualmente, al igual que muchas otras Dependencias y U.R's del Instituto Politécnico Nacional , en la creación de una pagina y un servidor Web, los cuales puedan servir como

medio de comunicación entre este Centro, la Comunidad Politécnica y todos aquellos que estén interesados en la Investigación y el Desarrollo Tecnológico en el campo de la computación (investigadores, Industriales, Dependencias Gubernamentales, etc.).

La revista polibits ve reforzado con esta acción el objetivo de expandir su campo de influencia, teniendo una forma mas de vinculación con el exterior y retroalimentarse con las aportaciones que en forma de comentarios o artículos nos envíen, ya sea en la forma tradicional o a través de la siguiente dirección de correo electrónico: **[cintec@vmredipn.ipn.mx](mailto:cintec@vmredipn.ipn.mx)**

# Cómo Direccionar Más Allá del Primer MegaByte en el Modo Real de las PC

*M. en C. Gustavo A. Mas Levario  
Profesor e Investigador del CINTEC-IPN.*

**L** La finalidad de este artículo es describir el método a seguir para acceder las direcciones de memoria superiores al primer megabyte en máquinas PC's que estén operando con el MS-DOS y en el modo real, desde un programa en C en interface con otro en ensamblador.

---

## Fundamentos

---

Antes de entrar en detalles con respecto al método es necesario que se analicen las capacidades de operación del direccionamiento de memoria de los microprocesadores Intel de la familia 80x86, en los modos real y protegido.

## Modo Real

El modo real se define [1] literalmente como sigue: En el Intel 286 y máquinas más avanzadas, estado operacional en el cual la computadora funciona como si fuera una 8086 u 8088. Se limita a direccionamientos de un megabyte de memoria.

La definición anterior se usará para detallar más algunos aspectos. En los 8088 y 8086 los programadores se enfrentan a una arquitectura de memoria segmentada en

bloques de 64KB que dificulta el manejo de ciertas estructuras de datos con longitudes mayores de 64KB. Otra restricción en el direccionamiento de memoria de mayor peso proviene por la cantidad de memoria que los 8088 y 8086 pueden direccionar, y la cual es de 1 MegaByte.

Debido a las restricciones del manejo de la memoria, cuando el MS-DOS se implantó como sistema operativo de las máquinas basadas en los microprocesadores 8088 y 8086, se le limitó a funcionar con un megabyte y como consecuencia el BIOS también se limitó a funcionar en un megabyte.

En la actualidad, aunque los microprocesadores de las computadoras personales sean 80386, 80486 o PENTIUM, cuya capacidad de direccionamiento es 4 Gigabytes, estas se siguen empleando con las restricciones del modo de operación de las máquinas 8088 y 8086.

La situación anterior fue ocasionada por la regla que adoptó Intel, la cual consiste en ofrecer una rigida compatibilidad de sus nuevos productos con los que ya tenía comercializados anteriormente [3]. Las máquinas 80286 y superiores ejecutan todo el software desarrollado para las máquinas 8088 y 8086. En este modo real de trabajo la única diferencia aparente consis-

te en la velocidad del proceso, siendo mucho mayor en las máquinas 80286 y superiores.

---

## El Direccionamiento en Modo Real

---

Las máquinas de Intel direccionan los objetos en la memoria en bytes y usan un modelo de direccionamiento de memoria conocido como segmentación, la cual consiste en particionar la memoria en segmentos. Dentro de cada segmento cada uno de los bytes tiene un desplazamiento (offset en computación) único.

Para direccionar un byte en la memoria tanto el segmento como el desplazamiento deben ser conocidos. Una dirección completa en modo real se especifica como SSSS:0000, donde SSSS representa el segmento y 0000 es el desplazamiento. La notación SSSS:0000 usa números en notación hexadecimal. Para obtener una dirección lineal o física en el modo real la parte SSSS se desplaza un lugar a la izquierda y en ese lugar vacío insertamos un cero, de modo que tendríamos lo siguiente SSSS0 y finalmente a SSSS0 le sumamos 0000; el resultado es una dirección lineal o física.

El valor de SSSS de la dirección SSSS:0000 es usada por uno de los registros de segmento, ya sea de

código, datos o pila, para direccionar junto con 0000 un campo dentro de un segmento de 64 Kbytes. Las direcciones van desde 0000:0000 pasando por A000:0000 hasta FFFF:000F, es decir, un rango de un megabyte.

Normalmente se indica con el número inicial de la notación SSSS:0000 (o sea la primera S leída de izquierda a derecha) el número del segmento al que se hace referencia. De este modo, el segmento cero, 0000:0000, es previo al segmento uno, 1000:0000, y así sucesivamente existiendo un espacio de 64 Kbytes entre cada uno.

Un fenómeno que se presenta en el modo real y que tiene que ver con el manejo de direcciones en el límite del megabyte en las máquinas 80286 y superiores es el llamado **"wrap-around"** [2].

Para el propósito de este artículo **wrap-around** tiene el siguiente significado: Hecho que se presenta en el cálculo de una dirección lineal el cual consiste en restar un megabyte (100000 en hexadecimal) a las direcciones que sean mayores del rango de un megabyte en el modo real. Por ejemplo, al calcular la dirección lineal de FFFF:0011 en un 80286 o superiores se presenta el fenómeno wrap-around y se generará la dirección uno, o sea 0000:0001. Los diseñadores de tarjetas madres de las máquinas 80286, 80386, 80486 han agregado una "puerta" para resolver esta restricción de compatibilidad en el modo real. Esta puerta permite que la línea de dirección 20 de las máquinas pueda ser usada para calcular una dirección lineal, en el modo real sólo se usan las líneas 0 a la 19 para un total de un megabyte. De modo que con la "puerta" activada el fenómeno wrap-around no ocurre, con la consecuencia de poder direccionar arriba del primer mega-

byte. Con la "puerta" desactivada wrap-around ocurre tal y como pasa en el modo real.

### Modo protegido

El modo protegido se define [1] literalmente como sigue: En el Intel 286 y superiores, un estado operacional que permite a la computadora direccionar toda la memoria. También previene a un programa introducirse en los límites de la memoria de otro, permitiendo de este modo ejecutarse en un entorno "protegido".

Además de la definición anterior necesitamos usar las siguientes definiciones, ya que en el modo protegido se usan términos nuevos con respecto al modo real:

**Selector [5].-** Una cantidad de 16 bits que especifica un descriptor de un segmento.

**Descriptor [5].-** Una cantidad de 8 bytes, especificando independientemente un objeto protegido.

**Objeto [1].-** Un módulo de datos relacionados y autónomos y su proceso asociado.

**Tabla de descriptores [5].-** Un arreglo de descriptores de segmento. Hay dos clases de tablas de descriptores: La Tabla Global de Descriptores (GDT) y un número arbitrario de Tablas Locales de Descriptores (LDT).

**Registro de la Tabla Global de Descriptores (GDTR) [5].-** Registro que guarda los 32 bits de la dirección base lineal y los 16 bits del límite de la GDT.

**Nivel de Privilegio [5].-** Un número en un rango predeterminado indicando el grado de protección del objeto de la memoria.

**Nivel de Privilegio del Descriptor (DPL) [5].-** Un campo en un descriptor indicando como está protegido el descriptor.

**Derechos de Acceso [5].-** Una cantidad de ocho bits contenida en algún descriptor, que indica las condiciones del objeto de la memoria. Las condiciones muestran si el objeto está presente en la memoria principal o no, su DPL, si es un objeto de datos o de código, y además si ha sido accedido o no.

**Granularidad .-** Los incrementos más pequeños que pueden ser diferenciados [7]. Con respecto al direccionamiento de memoria en los 80286 y superiores se puede decir que la granularidad es en byte o en páginas de 4 Kbytes [4]. La granularidad está relacionada con el campo límite del descriptor, de modo que con los 19 bits del campo límite podemos establecer el tamaño del objeto de la memoria, es decir, podemos tener hasta un megabyte de bytes si el bit G=0, o hasta un megabyte de páginas si el bit G=1 para dar un total de 4 Gigabytes.

**Unidad de Segmentación [5].-** Traduce direcciones segmentadas a direcciones lineales. La unidad de segmentación contiene una caché que guarda la información de la tabla de descriptores para cada uno de los seis registros de segmento.

La característica principal del mecanismo de protección en el modo protegido es el selector [4]. Todos los programas en el modo

protegido usan los mismos registros de segmento (CS, DS, ES y SS) que se usan en el modo real. La diferencia es lo que está contenido en los registros [8]. En el modo real los registros de segmento contienen la dirección inicial de un segmento en memoria. En el modo protegido los registros de segmento contienen un selector. Un selector consiste de un índice de 13 bits que apunta a un descriptor de alguna tabla de descriptors, un indicador de una tabla (ya sea GDT o LDT) de un bit y tres bits que indican el nivel de privilegio, que para la finalidad de este artículo no se utilizarán.

El descriptor es el otro nivel de protección, el cual contiene las siguientes características del objeto de la memoria: la dirección base de 32 bits, el límite de 19 bits, los derechos de acceso de 8 bits y otros cuatro bits de control de los cuales el que interesa es el bit llamado G (Granularidad) [4].

## Direccionamiento en el modo protegido

Para direccionar un byte, una palabra, o una doble palabra en el modo protegido [3], las máquinas realizan el siguiente proceso usando únicamente la unidad de segmentación: se obtiene el selector del registro de segmento involucrado para saber con que tabla de descriptors está relacionado (ya sea con la GDT o la LDT). Con el propósito de dar más sencillez a este artículo se empleó la GDT. Después de que el selector indica que la GDT se va a usar, se le localiza en la memoria física por medio de GDTR. Ya hecho lo anterior se localiza en la GDT al descriptor asociado con el selector usando su índice de 13 bits desplazado a la izquierda tres bits y así

obtener un índice de 16 bits para apuntar a una entrada de la GDT de 8 bytes que es el descriptor deseado. El contenido de este descriptor es usado para cargarlo en la unidad de segmentación en el registro de su caché asociado con el registro de segmento que contiene dicho selector. Finalmente la unidad de segmentación calcula la dirección lineal sumando la dirección base

00000000h usando los bytes 1, 2, 3 y 7, un límite de FFFFh usando los bytes 0, 1 y los cuatro bits menos significativos del byte 6, con G=1 y un valor de 92h en el byte 5, que son los derechos de acceso. Para la finalidad de este artículo se empleó ese valor en los derechos de acceso para indicar un segmento de datos de lectura/escritura de direccionamiento ascendente.

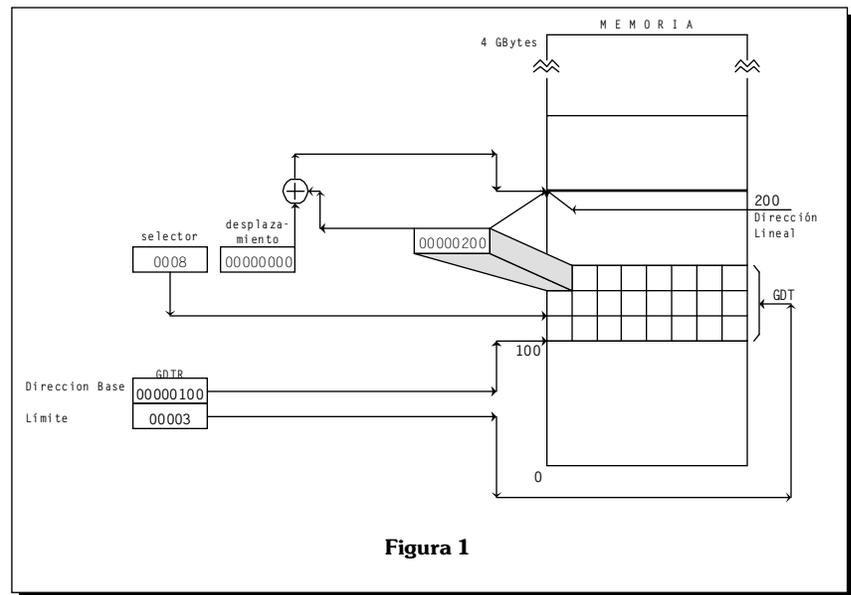


Figura 1

lineal del segmento que se desea acceder con el desplazamiento de 32 bits de alguna instrucción. En la figura 1 se muestra de un modo gráfico la traducción de dirección segmentada a lineal.

## El Modelo Plano de la Memoria [3]

Es una circunstancia en la cual el direccionamiento de la memoria física es visto sin segmentos, es decir la memoria es usada como un espacio lineal de 4 Gigabytes, empezando desde la dirección 00000000h hasta la dirección FFFFFFFFh.

Para lograr lo anterior se construye un descriptor en la GDT con una dirección base lineal de

Para implantar el modelo plano es necesario estar en el modo protegido, ya que los registros caché de la unidad de segmentación únicamente pueden ser reinicializados en su totalidad en ese modo de operación [6].

En el modo real, solamente el campo de la dirección base lineal se modifica directamente en los registros caché, de modo que los campos para el límite y los atributos permanecen fijos; y además, el campo límite se fija a FFFFh y la granularidad es en bytes (G=0) [6].

## Plan

Con todos los conceptos anteriores se debe idear un plan para direccionar un modelo plano desde el modo real usando el MS-DOS. El plan es el siguiente:

- 1.- Estar seguro de que no existe en operación otro programa que haga uso de la memoria extendida.
- 2.- Crear dos funciones, una que convierta las direcciones segmentadas a direcciones lineales y otra para que haga lo inverso, ya que cuando se involucre en el direccionamiento a GS y ES la unidad de segmentación estará inhibida.
- 3.- Crear una función que controle la "puerta" de la línea de dirección 20h del bus de direcciones.
- 4.- Crear una función que conmute al modo protegido desde el modo real e instale el modelo plano en los registros GS y ES, y además regrese al modo real.
- 5.- Hacer funciones que escriban y lean de cualquier parte del espacio de la memoria.

---

### Resultado

---

Con el plan ideado se logró un método factible que puede acceder los 4 Gigabytes que es capaz de manejar una máquina de Intel de 32 bits de un modo simple, ya que se sigue usando todo lo conocido de una PC.

---

### Restricciones

---

El programa anterior únicamente corre cuando no hay otro programa que haga uso de la memoria extendida, de modo que Windows,

EMM386.EXE, y otros no deben estar funcionando.

Por último, se debe reinicializar la computadora para prevenir una falla ocasionada por un conflicto de direccionamiento si otro programa (un tiempo después) hace uso de los registros de segmento de datos GS y ES en el modo real.

---

### El programa

---

El programa está hecho con una interface de lenguaje C y ensamblador de Borland Ver. 4.5 [2]. Está formado básicamente de una biblioteca con ocho funciones, las cuales se detallan a continuación.

Las funciones *seg\_a\_lineal()* y *lineal\_a\_seg()* se usan para emular la unidad de segmentación en el modelo plano, de modo que con estas funciones se tiene una conexión entre la segmentación del modo real y la linealidad del modelo plano. La expresión y el contenido de las funciones en lenguaje C es como sigue [2]:

```
LPTR seg_a_lineal(void far *p)
{
    return( ( (unsigned long) FP_SEG(p)
    << 4) + FP_OFF(p);
}
```

```
void far *lineal_a_seg(LPTR lin)
{
    void far *p;
    FP_SEG(p) = (unsigned int) (lin >> 4);
    FP_OFF(p) = (unsigned int) (lin & 0xF);
    return p;
}
```

El término LPTR indica un objeto de 32 bits sin signo, y los términos *FP\_SEG()* y *FP\_OFF()* son macros del C usados para generar la parte correspondiente al segmento y al desplazamiento de una dirección segmentada.

La función *seg\_a\_4Gigas()* es la parte principal del programa y efectúa dos acciones básicas:

- 1.- Conmuta al modo protegido para poner el modelo plano en los registros de segmento GS y ES, y ...
- 2.- regresar al modo real con los registros caché modificados.

Esta función llama a una rutina en ensamblador nombrada "prot-setup", para entrar desde el modo real al modo protegido y regresar de nuevo al modo real.

La función *a20()* desactiva y activa la "puerta" de la línea 20 del bus de direcciones para controlar el fenómeno "wrap-around" del modo real.

Las funciones *escribe\_8bits()* y *escribe\_32bits()* escriben un byte o una doble palabra respectivamente, en cualquier dirección deseada del rango de 0 a 4 Gigabytes.

Las funciones *lee\_8bits()* y *lee\_32bits()* leen un byte o una doble palabra, respectivamente.

Cada una de las últimas cuatro funciones anteriores llaman a una rutina escrita en ensamblador.

---

### Documentación del programa

---

El programa consta de cuatro archivos, tres de ellos están escritos en lenguaje C de Borland Ver. 4.5 y el restante está escrito en lenguaje ensamblador [2] para los procesadores de 32 bits.

El archivo BIBLI4G.H contiene las definiciones de todas las funcio-

nes y variables necesarias para la finalidad del artículo.

El archivo BIBLI4G.C contiene la implantación del contenido descrito en el archivo BIBLI4G.H.

El archivo MAIN4G.C contiene una aplicación que consiste en escribir y leer en la dirección FFFFCh (16 Mb-4 bytes) el dato BBBBAAAh. Esta aplicación se ejecuta adecuadamente en una máquina 80286 o superior con 16 Mbyte de RAM; si no se cuenta con 16 Mbyte de RAM, entonces se debe cambiar la dirección a otra deseada y que permita correr adecuadamente la aplicación.

El archivo BB4G.ASM contiene las rutinas del lenguaje ensamblador invocadas por las funciones del archivo BIBLI4G.C.

---

### Listados

---

En los siguientes cuadros se tiene el listado de la biblioteca de funciones desarrolladas para el manejo de la memoria.

```
// Archivo BIBLI4G.H
// Aquí están las definiciones de las funciones
// del archivo BIBLI4G.C.

/* el objeto LPTR es usado para guardar una dirección lineal */
typedef unsigned long LPTR;

/* funciones prototipo */
LPTR seg_a_lineal(void far *p);
void far *lineal_a_seg(LPTR lin);
void seg_a_4gigas(void);
void a20(int flag);
unsigned int lee_8bits(LPTR address);
void escribe_8bits(LPTR address, unsigned int byte);
void escribe_32bits(LPTR direccion, unsigned long
                    palabra32);
unsigned long lee_32bits(LPTR direccion);
// Fin del archivo BIBLI4G.H

// Archivo BIBLI4G.C
// Aquí está el desarrollo de las
// funciones del archivo BIBLI4G.H

#include <dos.h>
#include <conio.h>
#include <bibli4g.h>

/* se definen los controladores del teclado */
/* para habilitar la línea 20 de direcciones */
#define RAMPOR      0x70
#define KB_PORT     0x64
#define PCNMIPORT   0xA0
#define INBA20      0x60
#define INBA200N    0xDF
#define INBA200FF   0xDD

/* Convertir un apuntador far a una dirección lineal */
LPTR seg_a_lineal(void far *p)
{
    return (((unsigned long) FP_SEG(p)) << 4)
        +FP_OFF(p);
}

/* Convertir una dirección lineal a un apuntador far */
void far *lineal_a_seg(LPTR lin)
{
    void far *p;
    FP_SEG(p) =(unsigned int) (lin >> 4);
    FP_OFF(p) =(unsigned int) (lin & 0xF);
    return p;
}

/* estructura para hacer un descriptor para la GDT */
struct _GDT
{
    unsigned int limit;
    unsigned int base;
    unsigned int access;
    unsigned int hi_limit;
};

/* se inicializan dos descriptores para la GDT */
/* uno llamado nulo que es un default */
/* y el segundo para poner el modelo plano */
static struct _GDT GDT[2] =
{
    // descriptor no usado por el usuario y está por
    // default.
    {0, 0, 0, 0},
    // descriptor para poner un modelo plano en el
    // registro GS Y ES.
    {0xFFFF, 0, 0x9200, 0x8F}
}
```

## Cómo Direccionar Más Allá del Primer MegaByte en el Modo Real de las PC

```
};
/* estructura para inicializar el GDTR */
struct fword
{
    unsigned int limit;
    unsigned long linear_add;
};

/* definición del objeto gdtptr que es el */
/* medio para inicializar al GDTR */
static struct fword gdtptr;

/* se ajusta el limite de GS y ES a AGB */
void seg_a_4gigas()
{
    /* se calcula la direccion lineal y el limite de
    GDT y se introduce en el GDTR */
    gdtptr.linear_add = seg_a_lineal((void far *)
    GDT);
    gdtptr.limit = 15;

    /* se deshabilitan interrupciones comunes */
    _disable();

    /* se deshabilitan las interrupciones NMI */
    outp(PCNMIPORT, 0);

    /* codigo para llamar al modo protegido */
    // rutina del archivo BB4G.ASM
    // esta rutina pone el modelo plano
    // en GS y ES y regresa de nuevo al
    // modo real
    protsetup(&gdtptr);

    /* se habilitan las interrupciones de nuevo */
    _enable();

    /* rehabilitar NMI de nuevo */
    outp(PCNMIPORT, 0x80);
}

/* funcion de proposito general para habilitar */
/* A20 (flag = 1) o deshabilitar A20 (flag = 0) */
void a20(int flag)
{
    outp(INBA20, flag ? INBA20ON:INBA20OFF);
}

/* leer un simple byte de la memoria extendida */
/* dando una direccion lineal */
unsigned int lee_8bits(LPTR address)
{
    // rutina del archivo BB4G.ASM
    leer_lbyte(address);
}

/* escribir un simple byte hacia la memoria */
/* extendida dando una direccion lineal */
void escribe_8bits(LPTR address, unsigned int byte)
{
    // rutina del archivo BB4G.ASM
    escribir_lbyte(address, byte);
}

/* escribe 32 bits a la memoria extendida */
void escribe_32bits(LPTR direccion, unsigned
long palabra32)
{
    // rutina del archivo BB4G.ASM
    esc_32bits(direccion, palabra32);
}

/* se leen 32 bits de la memoria extendida */
unsigned long lee_32bits(LPTR direccion)
```

```
{
    void far * b;
    LPTR r = 0, t = seg_to_linear(b);

    // rutina del archivo BB4G.ASM
    leer_32bits(direccion, t);

    r = _AX;
    t = _DX;
    t = t<<16;
    t = t + r;
    return t;
}
// Fin del archivo BIBLI4G.C

// Archivo MAIN4G.C.
// Aquí está el programa principal, que usa las
// funciones del archivo BIBLI4G.C para acceder
// memoria en cualquier lugar.

#include <stdio.h>
#include <dos.h>
#include «bibli4g.h»

void main(void)
{
    // dirección FFFFFFFC (16 MBb- 4B]
    LPTR md = 0xFFFFFC;
    unsigned long d = 0;
    int i;
    char string[25];

    /* se hace un segmento de 4Gigas en
    GS y ES*/
    seg_a_4gigas();

    /* poner A20 en on */
    a20(1);

    /* se escribe el dato bbbbaaaa desde la
    dirección FFFFFFFC a la FFFFFFFF */
    escribe_32bits(md,0xBBBBAAAA);

    /* se lee el dato de la direccion FFFFFFFC */
    d = lee_32bits(md);

    // se convierte el dato leído en un string
    ltoa(d, string, 16);

    // se despliega el dato convertido en la pantalla
    printf(«En la direccion 0xFFFFFC Se
    escribio bbbbaaaa Y se leyo = %s\n», string);

    // mantiene el dato desplegado en la pantalla
    // hasta que recibe una interrupción del
    // teclado
    getchar();

    /* pone A20 en off de nuevo */
    a20(0);
}
// Fin del archivo MAIN4G.C

// Archivo BB4G.ASM
.MODEL LARGE,C

.386P
.CODE
PUBLIC protsetup, esc_32bits
PUBLIC leer_32bits, leer_lbyte
```

## Cómo Direccionar Más Allá del Primer MegaByte en el Modo Real de las PC

```
; rutina para ir al modo protegido luego poner
; el modelo plano en los registros GS y ES
IF @DataSize
protsetup proc fpointer:dword,c
    push ds
    lds bx,fpointer
ELSE
protsetup proc fpointer:word,c
    mov bx,fpointer
ENDIF
; se guarda el valor de fpointer en GDTR
    lgdt fword ptr [bx]
; Load GDT
IF @DataSize
    pop ds
ENDIF
; se prepara el procesador para meterlo
; al modo protegido
    mov eax,cr0
    or al,1
    mov cr0,eax
; se reinicializa el registro IP
    jmp short nxtlbl:
nxtlbl:
; se guarda en bx el selector que apunta
; al segundo arreglo de la estructura GDT[2]
; del archivo BIBLI4G.C
    mov bx,8
; se inicializan los registros GS y ES con el
; selector anterior para poner el modelo plano
; en los registros caché de GS y ES
    mov gs,bx
    mov es,bx
    and al,0feh
; el procesador se regresa al modo real de
; nuevo
    mov cr0,eax
    ret
protsetup endp

; leer un byte desde un LPTR
leer_lbyte proc address:dword,c
    xor ax,ax
; aquí se opera en modo real así que
; GS tiene el segmento 0000 que es la
; base del segmento de 4 Gigas
    mov gs,ax
    mov eax,address
; se lee el byte y se pone en al
    mov al,gs:[eax]
    xor ah,ah
    ret
; se regresa el byte leído en ax
leer_lbyte endp

// rutina que lee 32 bits en el modelo plano
leer_32bits proc direccion:dword, b:dword,c
    xor ax, ax
; aquí se opera en modo real así que
; GS tiene el segmento 0000 que es la
; base del segmento de 4 Gigas
    mov gs, ax
    mov ebx, direccion
    mov eax, dword ptr gs:[ebx]
    mov gs:[b], eax
    mov ax, word ptr gs:[ebx]
    mov dx, word ptr gs:[ebx+2]

    ret
; se regresa el dato leído en ax y dx
leer_32bits endp

; se escribe un byte en la dirección de LPTR
escribir_lbyte proc address:dword, byt:word,c
```

```
    xor ax,ax
; aquí se opera en modo real así que
; GS tiene el segmento 0000 que es la
; base del segmento de 4 Gigas
    mov gs,ax
    mov eax,address
    mov bx,byt
    mov byte ptr gs:[eax],bl
    ret
escribir_lbyte endp

; escribe una doble palabra a la dirección LPTR
esc_32bits proc direccion:dword,
    palabra32:dword,c
    xor ax, ax
; aquí se opera en modo real así que
; GS tiene el segmento 0000 que es la
; base del segmento de 4 Gigas
    mov gs, ax
    mov eax, direccion
    mov ebx, palabra32
    mov dword ptr gs:[eax], ebx
    ret
esc_32bits endp

    end
// Fin del archivo BB4G.ASM
```

---

### **Bibliografía**

---

- [1] Freedman Alan. "Diccionario de Computación". Quinta Edición Mc Graw Hill. 1993.
- [2] Williams Al. "DOS + 386 = 4 Gigabytes". Dr. Dobb's Journal. July 1990.
- [3] Angulo José Ma., Funke Enrique M. "386 y 486 Microprocesadores avanzados de 32 bits". Editorial Paraninfo.
- [4] Nelson Ross P. "80386/80486 Programming Guide". Second Edition. Microsoft Press. 1991.
- [5] Brumm Penn, Brumm Don, Scanlon Leo J. "80486 Programming". Windcrest Books. 1991.
- [6] Intel. "Microprocessors: Volume II". 1994.
- [7] Turner Rufus P., Gibilisco Stan. "The Illustrated Dictionary of Electronics". Fifth Edition. TAB. 1991.
- [8] Syck Gary. "Turbo Assembler Bible". SAMS. 1991.

# Diseño, Simulación y Construcción de un Control PID Aplicado a un Sistema Térmico

*M. en C. Romeo Urbieto Parrazales*  
*Profesor e Investigador del CINTEC-IPN.*

**E**ste artículo trata tres aspectos mayores que son: el diseño, la simulación, y la construcción de un control **PID** (proporcional más integral más derivativo) aplicado a un sistema térmico.

El diseño del sistema de control de temperatura comprende el modelado y cálculo de los parámetros del controlador, el proceso de temperatura, y las interfaces entre la computadora y el proceso. El comportamiento de la temperatura del agua en el recipiente es un modelo de primer orden expresado con la transformada  $s$ . En este modelo se muestra una curva de la variable de temperatura con respecto al tiempo. En la curva del proceso de temperatura de lazo abierto se obtuvieron tres parámetros de interés que son: la zona muerta, ganancia del proceso, y la constante de tiempo. El algoritmo de control empleado es un sistema clásico **PID** cuyos parámetros **K<sub>p</sub>**, **K<sub>i</sub>**, **K<sub>d</sub>** se obtuvieron por el método de Ziegler & Nichols. El algoritmo de control de temperatura implantado en la PC viene expresado en una ecuación de diferencias.

La simulación trata de representar la variable de temperatura del agua de un recipiente con capacidad de un litro. El rango de temperatura del agua en el recipiente es de 25°C

a 75°C con una precisión de 0.1°C. La representación de la variable de temperatura se llevó a cabo por medio de la creación de un programa gráfico estructurado en lenguaje C++, el cual contiene las rutinas de graficación y de cálculo de la variable de temperatura.

La construcción del sistema de control **PID** aplicado a la temperatura del agua contenida en un recipiente de un litro consiste de tres módulos mayores: el módulo de cálculo, consistente en una computadora PC 386; el módulo de conversión y de potencia, consistente en una interface electrónica conteniendo un puerto paralelo para capturar y enviar datos, conversión de datos de A/D y D/A, amplificador de corriente y voltaje para excitar la resistencia térmica que calienta al agua, y amplificador para sensor de temperatura; y el módulo físico, consistente de un recipiente de porcelana con capacidad de un litro, equipado con resistencia térmica y sensor de temperatura.

---

## 1. Introducción

---

El objetivo del sistema de control de temperatura del agua contenida en un recipiente es mantener la temperatura del agua a un valor deseado ("set point") **T<sub>o</sub>**.

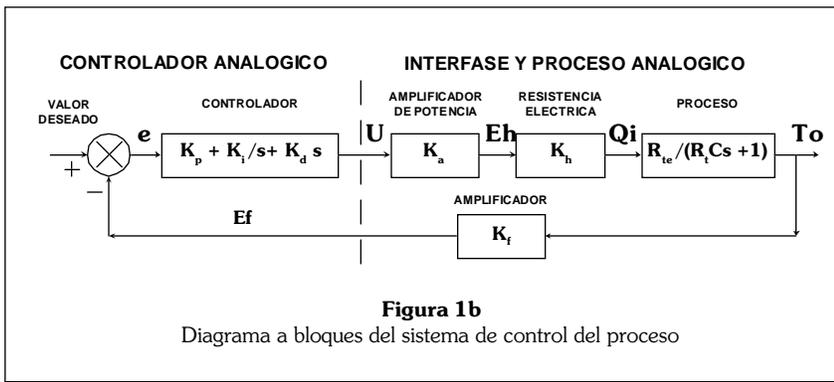
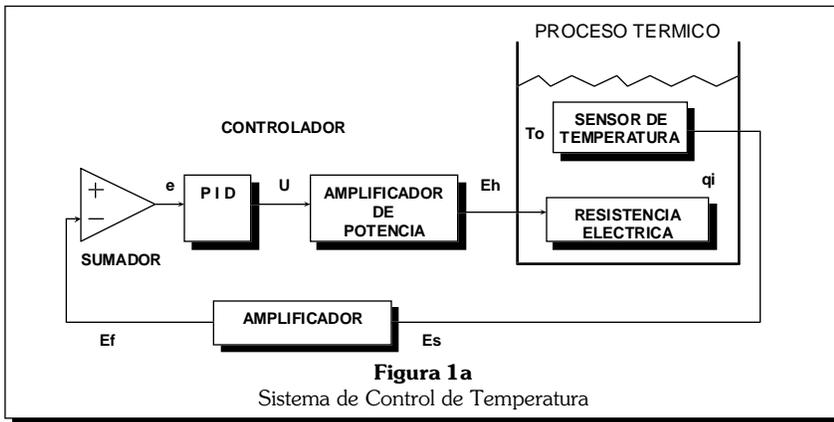
La temperatura del sensor dentro del agua en el recipiente produce

un voltaje  $E_s$ , proporcional a la temperatura **T<sub>o</sub>**. El voltaje obtenido por el sensor se amplifica a un factor **K<sub>1</sub>**, obteniendo un voltaje **E<sub>f</sub>**, el cual es aplicado a un sumador juntamente con el valor deseado de voltaje **E<sub>vd</sub>**. La diferencia de la señal de voltaje desde el sumador es una señal de error **e**.

La señal de error **e** se introduce a la entrada del controlador **PID** (proporcional más integral más derivativo) generando una señal de salida de control **U**, que alimenta después a un amplificador de potencia cuya salida es **E<sub>h</sub>**, para excitar una resistencia eléctrica sumergida en el agua; este a su vez entrega una entrada de calor **q<sub>i</sub>** causando que la temperatura del agua sea **T<sub>o</sub>** y también como el voltaje de retroalimentación **E<sub>r</sub>**, para aumentarlo, y la señal de diferencia en la salida del amplificador sumador **e**, para decrementarlo (a consecuencia de la retroalimentación negativa).

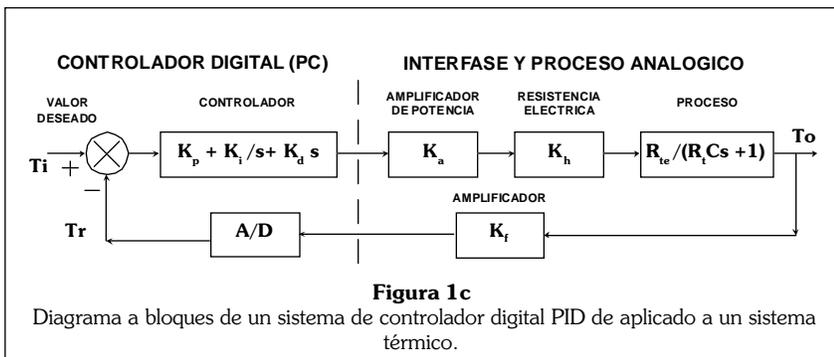
Cuando la señal de diferencia se aproxima al valor de cero, el excitador de corriente corta la señal de entrada de la resistencia eléctrica, por lo tanto, el calor del agua deja de aumentar. Como resultado, la temperatura del agua en el recipiente es la del valor deseado de temperatura **T<sub>o</sub>**.

El sistema de control de temperatura de la **figura 1a** se puede representar por un diagrama a blo-



ques, como se puede observar en la **figura 1b**.

El sistema de la **figura 1b** es un sistema analógico y el controlador se implanta con elementos analógicos. Para el caso de estudio a efectuar el sistema de control será estructurado usando una computadora digital tipo PC. El uso de la computadora digital como controlador lleva consigo un convertidor de señal: el convertidor analógico a digital. Lo demás queda igual (ver **figura 1c**).



alimentado por la resistencia eléctrica al agua,  $q_i$ , y el calor perdido en el recipiente y el ambiente puede ser igual a la elevación de calor en el agua. La acumulación de calor en el agua es proporcional a la razón de cambio de la temperatura del agua, con  $C_t$  de capacitancia térmica del agua, siendo esta la constante de proporcionalidad. De esta manera,

el flujo de calor es:

$$q_i - q_o = C_t (dT_o / dt) \text{ [Btu/s]} \quad (2.1.1)$$

donde  $q_i(t)$  = calor alimentado al agua [Btu/s]

$$q_o = (T_o - T_a) / R_t \quad (2.1.2)$$

$q_o$  = flujo de calor [Btu/s] a través de agua - ambiente.

$R_t$  = Resistencia térmica del agua en el recipiente [°F, sec/Btu].

$C_t$  = Capacitancia térmica del agua en el recipiente [Btu/°F].

Sustituyendo (2.1.2) en (2.1.1) :

$$R_t C_t (dT_o / dt) + T_o = R_t q_i + T_a \quad (2.1.3)$$

$$t To + To = R_t q_i(t) + T_a \quad (2.1.4)$$

Para linealizar el sistema se tiene:

$$T_o(0) = 0 \quad (2.1.5)$$

$$T_a = R_{te} q_i - R_t q_i \quad (2.1.6)$$

donde  $R_{te}$  es la resistencia térmica efectiva de las paredes del recipiente; entonces:

$$\tau To(t) + To(t) = R_{te} q_i(t) \quad (2.1.7)$$

$\tau = R_t C_t$  es la cte. térmica del recipiente con agua.

Este tipo de ajuste matemático puede fácilmente hacerse en situaciones donde el calor  $q_i$  es aplicado a una razón constante y donde  $T_a$  lo

## 2. Diseño

### 2.1 Módulo de proceso

Para iniciar el diseño del sistema de control computarizado antes expuesto es necesario analizar primero el proceso térmico. Para esto, se propondrá un modelo de proceso lineal. El modelo matemático del proceso será derivado al considerar las leyes de la termodinámica, las cuales gobiernan los flujos de calor. Es decir, la diferencia entre el calor

es también. Así mismo,  $R_{te}$  puede ser tratado como otro parámetro del sistema el cual incluye efectos de carga.

Pasando a transformada de Laplace (2.1.7):

$$T_o(s) = \frac{Q_i(s) R_{te}}{\tau s + 1} \quad (2.1.8)$$

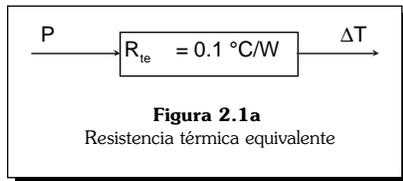
La resistencia térmica equivalente o ganancia del proceso térmico se puede obtener por medios experimentales midiendo los watts de entrada o aplicados a la resistencia eléctrica y el rango de temperatura del agua. Los datos que se tienen de la resistencia eléctrica son :

$$\begin{aligned} R &= 28.3 \text{ W} \\ I &= 4.22 \text{ A} \\ V &= 124 \text{ VAC} \end{aligned}$$

$$P = R I^2 = (28.3)(4.22)(4.22) = 504 \text{ W} \quad (2.1.9)$$

$$\Delta T = T_{75} - T_{25} = (75^\circ\text{C} - 25^\circ\text{C}) = 50^\circ\text{C} \quad (2.1.10)$$

$$\begin{aligned} R_{te} &= \Delta T / P = 50^\circ\text{C} / 504 \text{ W} \\ &= 0.1^\circ\text{C/W} \end{aligned} \quad (2.1.11)$$

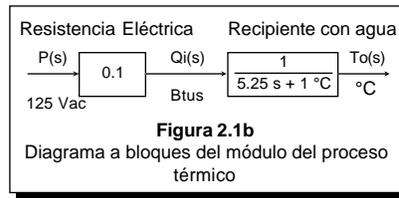


La constante de tiempo térmica  $\tau$ , y la zona muerta  $q$ , también se pueden obtener por método empírico, calculando el tiempo de la temperatura al 63.2 % de  $T_o$ , esto es:

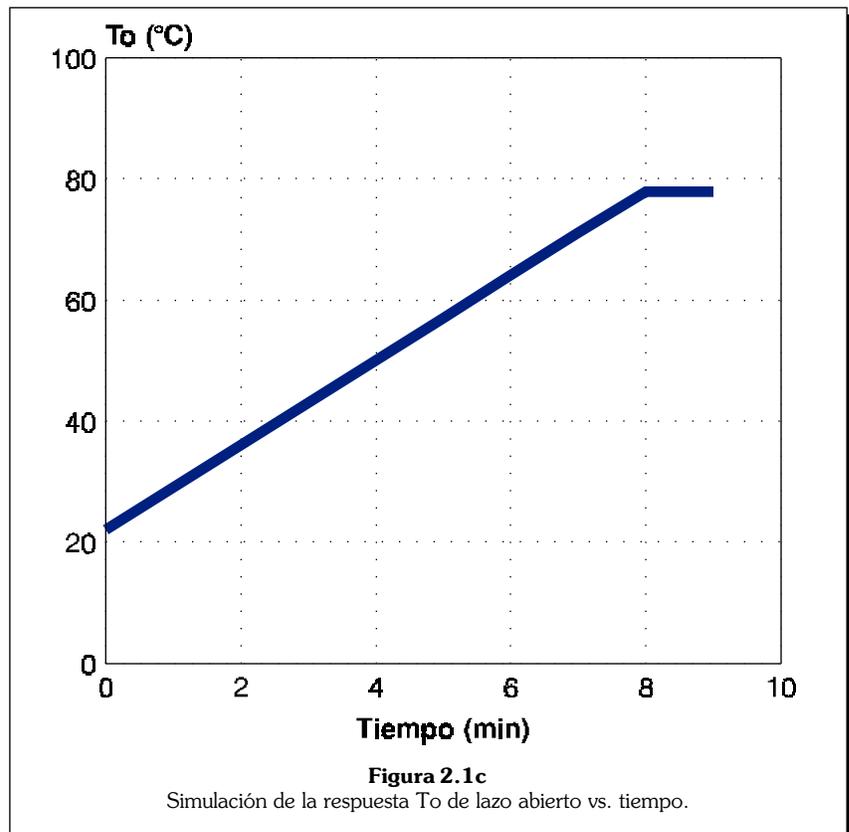
$$\begin{aligned} T_i &= (0.632)(50^\circ\text{C}) + 25^\circ\text{C} \\ &= 31.6^\circ\text{C} + 25^\circ\text{C} \\ &= 56.6^\circ\text{C} \end{aligned}$$

Calentando el agua a este valor se obtiene un tiempo  $\tau = 5.25$  minutos. La zona muerta  $q$  fue de

15 segundos. El modelo térmico correspondiente queda como sigue:

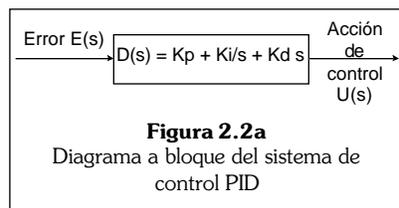


La respuesta en el tiempo se puede apreciar en la **figura 2.1c**.



## 2.2 Módulo de Control

La función de transferencia de un algoritmo **PID** está dada por :



se pueden calcular los parámetros del controlador **PID**:  $K_p$ ,  $K_i$ , y  $K_d$ .

Sustituyendo sus valores :  
 $K = 0.1, \tau = 5.25 \text{ min}, q = 0.25 \text{ min}.$  (2.2.1)

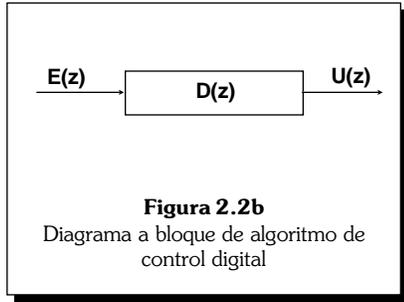
$$\begin{aligned} K_p &= 1.2 t / K q \\ &= (1.2)(5.25 \text{ min}) / (0.1^\circ\text{C/W})(0.25 \text{ min}) \\ &= (6.3) / (0.025) = 252 \end{aligned} \quad (2.2.2)$$

$$K_i = 1/2 q = 1/(2)(0.25) = 1/0.5 = 2 \quad (2.2.3)$$

$$K_d = 0.5 q = (0.5)(0.25) = 0.125 \quad (2.2.4)$$

Se asume que  $T = 0.5$  min.

En el modo digital, el controlador **PID** toma la forma:



El controlador PID se expresa como:

$$D(z) = K_p + K_i T \frac{z+1}{2(z-1)} + K_d \frac{z-1}{Tz} \quad (2.2.5)$$

donde:

$D(z)$  es el algoritmo de control digital PID.

$T$  es el período de tiempo entre muestreos.

$K_i$  ganancia integral.

$K_p$  es la ganancia proporcional.

$K_d$  es la ganancia derivativa.

Entonces, la variable controlada  $U(z)$  es:

$$U(z) = D(z)E(z) = K_p E(z) + [K_i T \frac{z+1}{2(z-1)}] E(z) + [K_d \frac{z-1}{Tz}] E(z) \quad (2.2.6)$$

Tomando la transformada inversa de  $z$ , se tiene:

$$u(k) = u(k-1) + K_p e(k) - K_p e(k-1) + [K_i T e(k)]/2 + [K_i T e(k-1)]/2 + K_d e(k)/T - 2K_d e(k-1)/T + K_d e(k-2)/T \quad (2.2.7)$$

Sustituyendo los valores de  $K_p$ ,  $K_i$ , y  $K_d$  encontrados en la simulación se tiene que:

$$u(k) = u(k-1) + 252.75 e(k) - 252 e(k-1) + 0.25 e(k-2) \quad (2.2.8)$$

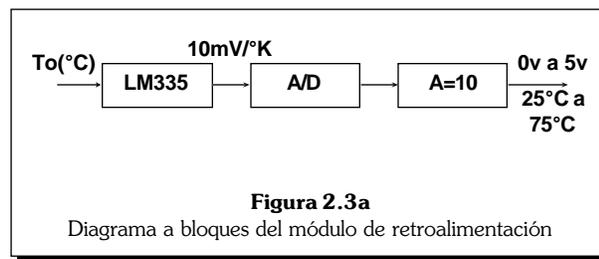
Este será el algoritmo de control **PID** final a implementar, siendo  $u(k-1)$  la señal de control o comando de control retardado un período de tiempo,  $e(k-1)$  la señal de error también retrasado un período de tiempo, y  $e(k-2)$  dos períodos de tiempo.

### 2.3 Módulo de Retroalimentación

Otro módulo de importancia en el sistema de control térmico es su retroalimentación, el cual sustituye al hombre en su intento de controlar un proceso. El módulo de retroalimentación consiste básicamente de: un sensor de temperatura y un sistema de amplificación o acondicionamiento de señal de medida de 0 a 5 voltios.

El sensor de temperatura se diseñó con un circuito integrado LM335 de National Semiconductor, basado en el principio del diodo. El circuito integrado, mediante una polarización externa adecuada, provee a su salida un voltaje del orden de los 10mV/°K, a una corriente de 1mA. (Ver diagrama esquemático final).

El rango de temperatura de operación de este circuito integrado va desde los -55°C hasta los 100°C. En este caso de estudio trabajará desde los 25°C a los 75°C. El circuito integrado para 25°C da un voltaje de salida de 2.98 voltios.



Este voltaje de salida se obtiene a partir de:

0°C equivalen a 273°K, como el dispositivo avanza cada 10mV/°K, entonces para esta temperatura se tienen 2.73 voltios. Entonces para 25°C que va ser el comienzo serán (273 + 25)°K que es igual a 298°K por 10 mV/°K, se llega a 2.98 voltios de salida en el sensor; así, 75°C serán 3.48 voltios. La señal de este sensor se aplica directamente a un convertidor Analógico a Digital para capturar los voltajes entre 2.98 voltios a los 3.48 voltios, siendo una diferencia de 0.5 voltios.

Usando el programa de simulación antes expuesto, se incluye un circuito amplificador diferencial con referencia en 2.98 voltios para compararlo con la lectura de los 25°C y que se obtengan 0 voltios. El amplificador tendrá una ganancia de 10 para obtener un rango de voltaje de 0 a 5 voltios @ 10 mA, que representan el rango de 25°C a 75°C. El diagrama final a bloques de retroalimentación quedará como en la **figura 2.3a y 2.3b**.

### 2.4 Módulo de Excitación

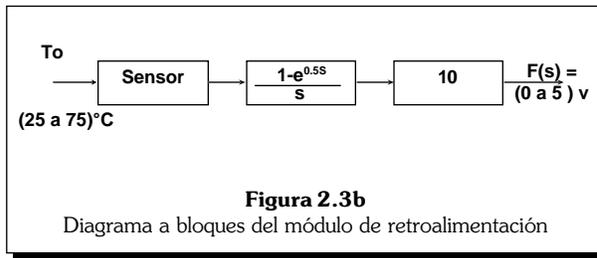
La potencia  $u$  aplicada a la resistencia térmica es una función del error entre el comando de temperatura y la temperatura medida por el sensor, y es calculada por la PC. Esta potencia es aplicada a la resistencia con la ayuda de circuitos de estado sólido, como un dispositivo opto-electrónico y un triac (ver esquema electrónico).

El dispositivo optoelectrónico

aislará la PC del potencial aplicado a la resistencia y además dará el prendido y apagado inteligente al interruptor de potencia o triac. El triac es un interruptor de potencia del estado

sólido, y será el que dé la potencia final a la resistencia térmica. Este interruptor se cerrará cada  $t_p$  se-

gundos o minutos según sea el error; si el error es grande el período de preñado será grande y viceversa, y el apagado será equivalente al tiempo de muestreo  $T_s$ .



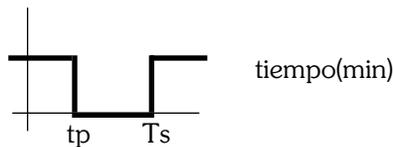
Entonces, la potencia promedio aplicada a la resistencia térmica en el intervalo  $T_s$  es:

$$u = \frac{1}{T_s} \int_0^{T_s} V^2 dt = \frac{V^2 t_p}{R T_s} \quad (2.4.1)$$

$$t_p = \frac{u T_s}{V^2 / R} \quad (2.4.2)$$

Dependiendo de la potencia requerida a ser aplicada a la resistencia eléctrica,  $t_p$  es calculada en la PC. Un pulso de ancho  $t_p$  se dispone cada vez que exista un instante de muestreo.

P(Watts)

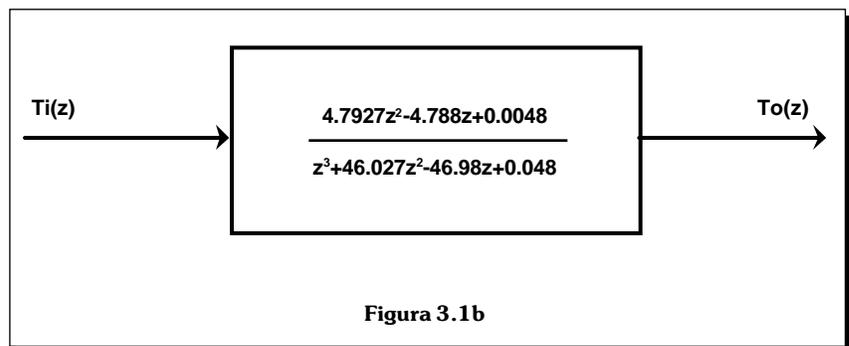
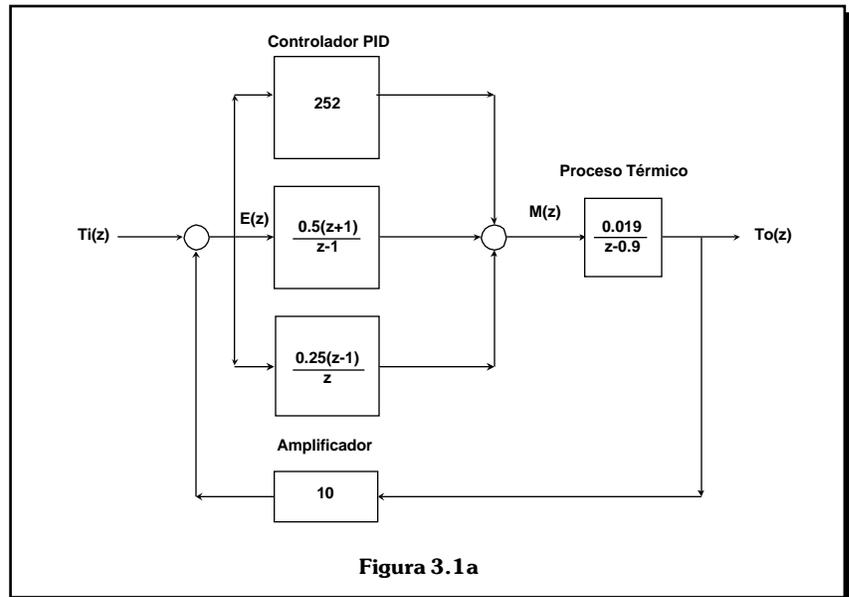


### 3. Simulación

Para la simulación del sistema de control se requiere la función de transferencia total  $T_i(z)T_o(z)$ . Las **figuras 3.1a y 3.1b** muestran esta función tanto en su representación gráfica como en forma matemática.

A partir de esta función de transferencia se obtiene la respuesta del sistema térmico en tiempo discreto  $k$ .

$$T_o(k) = -46.027 * T_o(k-1) + 46.98 * T_o(k-2) - 0.048 * T_o(k-3)$$



$$+ 4.7927 * T_i(k-1) + 4.788 * T_i(k-2) + 0.0048 * T_i(k-3) \quad (3.1)$$

La simulación del sistema de control aplicado al proceso térmico tiene como interés ajustar los parámetros de control  $K_p$ ,  $K_i$ , y  $K_d$  del controlador para una sintonía perfecta. A continuación se presenta el programa que lleva a cabo esto.

```

/*
                INSTITUTO
        POLITECNICO NACIONAL
        CENTRO DE INVESTIGACION
        TECNOLOGICA EN COMPUTACION

Proyecto:
    Software gráfico para un sistema de control PID aplicado a un
    sistema térmico.

Participantes:
    M. en C. Romeo Urbieta Parrazales
    Col. Granjas México, Delg. Iztacalco
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include "dos.h"
#include <string.h>
#include "borracad.c"

/* Definición de variables para el 8255 */
#define PA 0x100
#define PB 0x101
#define PC 0x102
#define PCONTROL 0x103

/* Configuración de los puertos del 8255 */
#define PALABRA1 0x82
#define PALABRA2 0x8B

/* Constantes para el ADC y DAC */
#define BAJO 0
#define ALTO 55000
#define RETAR 0
#define LIM 500

/* Definición de Funciones */
void ini_graficos(void); /* rutina de inicia gráfico */
void cie_graficos(void); /* rutina de cierra gráfico */
void ecua_graf(void); /* rutina ecuaciones a graficar */
void pinta_ejes(void); /* rutina de pintar ejes */
int adc(void);
void dac(void);

int main()
{
    ini_graficos(); /* llama subrutina de iniciar gráficos */
    pinta_ejes(); /* llama subrutina de pintar ejes */
    ecua_graf(); /* llama subrutina de pintar ejes */
    getche(); /* espera tecla */
    cie_graficos(); /* llama subrutina de cerrar gráfico */
    return 0;
}

void ini_graficos(void)
{
    int modo, controlador=DETECT;
    int error_graf; /* controlador = 0; */
    modo = 4;
    initgraph(&controlador, &modo, ""); /* "c:\tgc"; */
    error_graf = graphresult();
    if (error_graf != grOk)
    {
        printf("Error en gráficos -> ", grapherrormsg(error_graf));
        exit(1);
    }
}

void cie_graficos()
{
    closegraph();

```

```

}
void ecua_graf()
{
    char buffer[100];
    double T0, T1, T2, T3, R0, R1, R2, R3
    int xorg = 50, yorg = 440; /* El origen esta en 50,440 */

    k = 0;
    T0 = 0.0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color);

    k = 1;
    R0 = R1 = R2 = cajon[i].M

    T1 = 4.7927*R0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color)

    k = 2;
    R0 = R1 = R2 = R3 = cajon[i].M
    T2 = -46.027*T1 + 4.7927*R1 + 4.788*R0;
    x = xorg + k;
    y = yorg - T0;

    putpixel(x, y, color)

    k = 3;

    T3 = -46.027*T2 + 46.98*T1 + 4.7927*R2 + 4.788*R1 + 0.0048*R0;
    x = xorg + k;
    y = yorg - T0;
    putpixel(x, y, color);

    for (k = 4; k <= 500; k++) {

        T4 = -46.027*T3 + 46.98*T2 - 0.048*T1 + 4.7927*R3 + 4.788*R2 +
        0.0048*R1;

        C0 = C1;
        C1 = C2;
        C2 = C3;
        C3 = C4;

        x = xorg + k;
        y = yorg - T0;
        putpixel(x, y, color);
    }

void pinta_ejes()
{
    char buffer[200];
    int pasov = 50; /*45 */
    int pasoh = 50;
    int xorg = 45, xorg0 = 50, xorg1 = 550;
    int yorg = 189, yorg0 = 435, yorg1 = 440, y=390;

    setbkcolor(BLUE);
    setcolor(WHITE);

    rectangle(70, 1, 550, 50);
    outtextxy(240, 5, "CINTEC");
    outtextxy(150, 20, "Medición de voltaje y temperatura");

    rectangle(70, 100, 550, 147);
    setttextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(225, 103, "Valores a Graficar ");
    outtextxy(160, 120, " Voltaje Temperatura");

    moveto(xorg, yorg); /* POSICION INICIAL */
    lineto(xorg, yorg1); /* MARCO VERTICAL */

```

```

lineto (xorg1, yorg1);      /* MARCO HORIZONTAL */
moveto (xorg, yorg1);
moveto (xorg, yorg);      /* POSICION INICIAL */

while (y > yorg)
{
    /* CONTADOR VERTICAL DE MARCAS VERTICALES */
    moveto (xorg, y);      /* POSICION INICIAL */
    lineto (xorg0, y);     /* PRIMERA MARCA VERTICAL SUPERIOR */
    y = y - pasov;        /* ESPACIO ENTRE MARCA Y MARCA */
}

setttextstyle(SMALL_FONT, VERT_DIR, 5);
sprintf(buffer," Voltaje");
outtextxy(0,175,buffer);

setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"20");
outtextxy(18,390,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"40");
outtextxy(18,340,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"60");
outtextxy(18,290,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"80");

outtextxy(18,240 ,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 4);
sprintf(buffer,"100");
outtextxy(18,190,buffer);
sprintf(buffer," TEMPERATURA ( C)");
outtextxy(10,160,buffer);

moveto (xorg, yorg1); /* POSICION INICIAL */
xorg = xorg + pasoh; /* DATO DE PRIMER MARCA HORIZON- */
/* TAL IZQUIERDA */

while (xorg <= xorg1)
{
    /* CONTADOR HORIZONTAL DE MARCAS */
    moveto (xorg, yorg1); /* POSICION DE PRIMER MARCA */
    /* HORIZONTAL */
    lineto (xorg, yorg0); /* PRIMERA MARCA HORIZONTAL */
    xorg = xorg + pasoh; /* ESPACIO ENTRE MARCA Y MARCA */
    /* HORIZONTAL */
}

setttextstyle(SMALL_FONT, HORIZ_DIR, 5);
sprintf(buffer,"250 300 350 400 450 500 550 600 650 700
75");
outtextxy(49,450,buffer);
setttextstyle(SMALL_FONT, HORIZ_DIR, 5);
sprintf(buffer," TIEMPO (°mS)");
outtextxy(200,460,buffer);
}
    
```

## 4.0 Implantación

La implantación del sistema consiste de: algoritmo de control instalado en la computadora como “software”; una tarjeta que contiene un convertidor A/D como entrada de señal y un circuito integrado opto-electrónico, así como dos interfaces electrónicas, una para acondicionar la señales del sensor y la otra para la resistencia eléctrica.

### 4.1 Comparador y Algoritmo de Control

El comparador del sistema de la figura 1c es como sigue:

$$e(k) = T_i(k) - T_f(k) \quad (4.1.1)$$

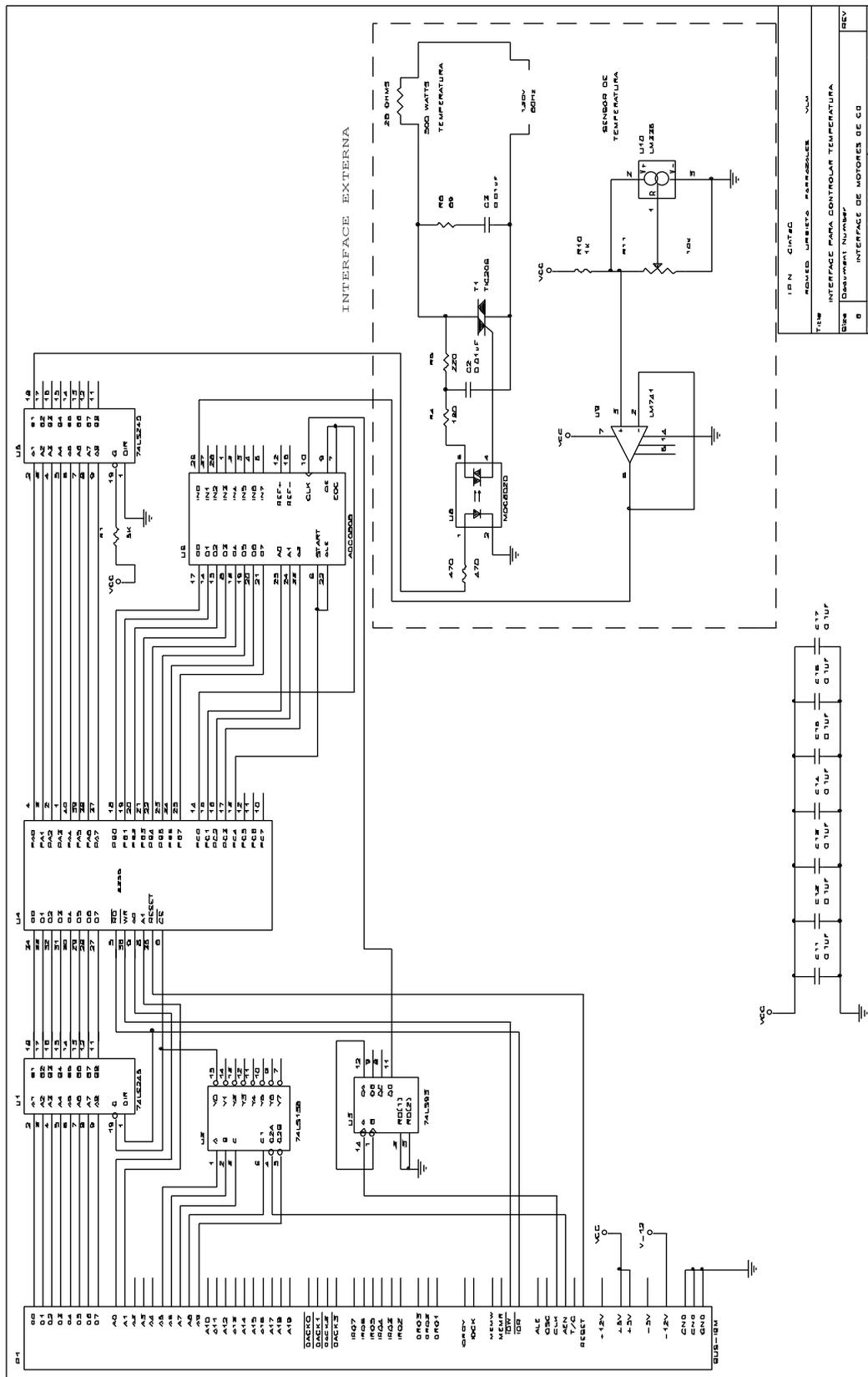
Tomando la transformada inversa de  $z$ , de la ecuación 2.2.7 se tiene:

$$\begin{aligned}
 u(k) = & u(k-1) + K_p * e(k) \\
 & - K_p * e(k-1) + [K_i * T * e(k)] / 2 \\
 & + [K_i * T * e(k-1)] / 2 + K_d * e(k) / T \\
 & - 2K_d * e(k-1) / T \\
 & + K_d * e(k-2) / T
 \end{aligned} \quad (4.1.2)$$

Sustituyendo los valores de  $K_p$ ,  $K_i$ , y  $K_d$  encontrados en la simulación y desarrollando, se tiene que la acción de control es:

$$\begin{aligned}
 u(k) = & u(k-1) + 252.75 * e(k) - 252 * e(k-1) \\
 & + 0.25 * e(k-2) \quad (4.1b)
 \end{aligned} \quad (4.1.3)$$

el cual será el algoritmo de control PID final a implementar en programa en tiempo real, siendo  $u(k-1)$  la señal de control o comando de control retardado un período de tiempo,  $e(k-1)$  la señal de error también retrasado un período de tiempo, y  $e(k-2)$  dos períodos de tiempo. El programa de control está estructurado en lenguaje C++ en tiempo real.



IB N	CINTAC
TIM	IBOLED UBERECS. SAPPARELLE
TIME	INTERFACE PARA CONTROLAR TEMPERATURA
SIZE	DRAWING NUMBER
8	INTERFACE DE MOTORES DE 50
DATE	MARZO 3, 1995
	STAGE
	41

### Conclusiones

El sistema de control de temperatura responde muy bien a los cambios de "set point" de temperatura pedidos, mientras que la inercia térmica es abatida por el sistema de control computarizado. La precisión de este sistema de control es del orden de 0.1%. La implantación del sistema resulta ser barata y versátil.

### Bibliografía

- [1] M. Gopal. Digital Control Engineering. Capítulo 8. Ed. John Wiley & Sons Limited. 1989.
- [2] R. Gayakwad & L. Sokolof. Analog and Digital Control Systems. Pags. 469-482. Ed. Prentice Hall. 1988
- [3] Data Acquisition Linear Devices. Data Book. Pags. 6-21 a 6-29. National Semiconductor . 1989.
- [4] Joseph J. Carr . Sensor and Circuits. Cap.4 Temperature sensors. Ed. Prentice Hall. 1993.
- [5] P.C. Sen. Thyristor DC Drives. Cap.6 Closed Loop Control. Ed. Krieger P.C. 1991.

# Control Remoto de Formato Comercial

C. Aquilino Cervantes Avila  
Alumno de la Maestría del CINTEC-IPN.

**E**l propósito inicial de este trabajo fue el de manejar un sistema mínimo 80188 por medio de un control remoto, como opción adicional al teclado. De esta forma, se parte de las experiencias de los alumnos Marcos Germán Aureliano Guadalupe y Agustín Cruz Contreras, quienes en la materia de prerequisite "Introducción a los Lenguajes de Alto Nivel", impartida por el profesor Miguel Angel Partida Tapia (M. en C.) para la Maestría de este Centro, investigaron el formato que presenta la codificación de un control remoto comercial.

Realmente lo que se diseña en este desarrollo es un teclado común del tipo matriz, pero se aumenta un registro para captura de la tecla, un registro de corrimiento y un codificador que hace el cambio de formato serie al formato comercial. El control remoto se desarrolla sobre un dispositivo MAPL 128; esto da la ventaja de trabajar en un espacio pequeño para la tarjeta del control remoto, y asegura un tiempo de vida para la batería de alimentación mayor a que si fuera desarrollada en base a lógica normal.

## Descripción general

El teclado trabaja de la siguiente manera: existe un contador binario

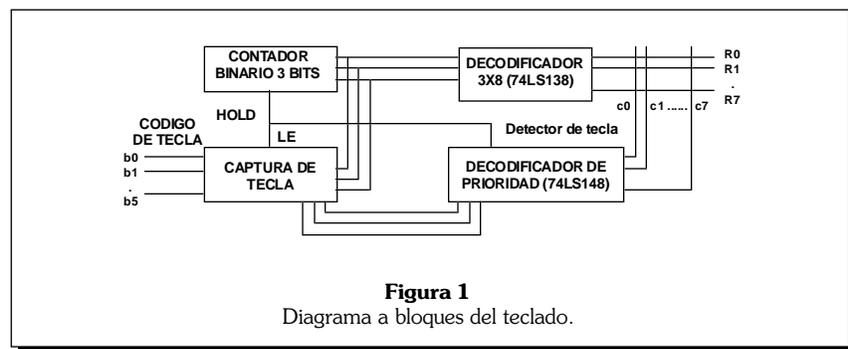
de tres bits con captura (**hold**), que alimenta un decodificador (74LS 138) de 3 X 8, activo bajo. Así, se logra el efecto de "búsqueda de tecla" en las salidas del decodificador; por otra parte, se tiene un decodificador de prioridad (74LS 148) 8 X 3, el cual regresa un número binario correspondiente a la entrada que recibe un nivel bajo. Además, una señal indica si alguna línea de entrada recibió un estado bajo.

Con estos tres dispositivos se forma un teclado colocando las líneas de entrada del codificador de prioridad como columnas y las salidas del decodificador como renglones, agregando un registro de captura de tecla.

su vez al registro de captura, asegurando que se tiene la información tanto de columnas como de renglones.

Para evitar errores debido al ruido eléctrico producido por los rebotes de las teclas, se agrega una red RC en la línea de detección de tecla, eliminando este problema. Con ello, la línea de detección se configura como salida y también como entrada al MAPL128.

La tecla de detección también se usa para indicar un inicio de sincronía, para que el receptor conozca el momento en que está llegando información y no ruido.



NOTA: si existe algún problema en el reconocimiento de tecla, se recomienda que las columnas posean resistencias de levantamiento

Así, se realiza el arreglo para que la línea que detecta algún bajo también detenga al contador y dispare a

La información contenida en el registro de captura, es transportada por un registro de corrimiento para obtener un formato serie. Cada bit es manipulado por el codificador formato comercial y enviado por un emisor óptico.

El codificador comercial transporta la información SOLO EN LA PARTE BAJA de la señal, esto es, en la parte baja de los pulsos tenemos que:

	Representa un inicio o sincronía
	Representa un estado bajo
	Representa un estado alto

Es posible observar que el nivel alto representa un espacio, el estado bajo representa dos espacios y un inicio de transmisión es lo mismo a tres espacios. Aunado a esto es necesario indicar que la codificación es del tipo retorno a uno y, además, el tiempo de permanencia en uno es constante y proporcional a un espacio. El número de pulsos que transmite un control remoto comercial es de 12 pulsos por tecla y en el momento en que no se está enviando información, la salida se encuentra en nivel alto.

Debido a que se tienen 8 columnas y 8 renglones, solo se tienen 6 bits de información y el paquete de datos que recibe el receptor necesita tener 12. La solución a este problema fue recorrer nuevamente los 6 bits de información.

Para hacer más flexible al control remoto, se presenta la opción de una tecla que permita doblar la capacidad del teclado (SHIFT); para este fin se emplea un flip-flop tipo 'D' configurado para que cambie de nivel cada vez que se oprima una tecla externa; este bit se pasa por el registro de corrimiento y se entrega al codificador formato comercial. Entonces, lo que hace el sistema realmente es recorrer primero los 6 bits de información de tecla, recorrer nuevamente los primeros 5 bits y agregar al final, el bit que corresponde a SHIFT.

Una vez que se detecta una tecla, aunque se oprima otra, no se hace caso de ella hasta el término del envío de la información. Si se oprimen dos teclas a la vez solo será reconocida la de mayor prioridad.

Al término de envío de información, se restablece nuevamente la cuenta del contador para seguir con la "búsqueda de teclas".

Si se mantiene una tecla oprimida la información seguirá siendo enviada al receptor; esto permite que sea posible observar la señal en un osciloscopio y crear la función conocida en teclados como "typematic".

### **Descripción de la programación del MAPL**

Para programar el MAPL 128, se utilizó el método de tablas de verdad con paginación manual, usando el paquete OPAL propiedad de NATIONAL SEMICONDUCTORS. Es necesario recalcar que el OPAL no soporta más de un tipo de tabla de verdad, por lo que, si se desea verificar el ejemplo, es necesario utilizar métodos alternativos o compilar en forma parcial las diferentes tablas, y emplear los módulos .EQN.

En la página 000 del MAPL, se localiza un contador binario de 3 bits unido en forma directa a un decodificador 3x8. Se puede observar que cuando la señal "tec" es llevada a cero, se hace una captura de la cuenta que posee el contador, y se hace un cambio a la página 001.

La señal "tec" indica si alguna tecla fue activada; para esto, en el diagrama del control remoto se une la señal de salida "detec" con la señal de entrada "tec". La señal "detec" envía un estado bajo si al menos una de las entradas c0 - c7 es

llevada a cero. Detec es activa en las 8 páginas.

En la página 001 se localiza un decodificador de prioridad el cual permite verificar cual de las señales c0 - c7 fue llevada a un estado bajo; si fueron varias las teclas activadas solo se tomará la de mayor prioridad. El registro de captura que se menciona en el diagrama de bloques se implementa dentro de los propios módulos descritos, gracias a que tanto las salidas como los nodos internos se encuentran registrados.

El registro de corrimiento obliga a tener en una sola línea la información de todos los bits del registro de captura en un instante de tiempo diferente; para lograr este objetivo, se deja que el registro de captura se encuentre activo en la página 010 y se maneja el programa de tal manera que se toma el primer bit del registro de captura y se pasa al codificador comercial; una vez que se terminó de codificar, se regresa nuevamente al registro de captura, para tomar el siguiente bit, se codifica y se regresa nuevamente al registro de captura, y así sucesivamente. Como la información de la tecla activada se encuentra en 6 bits y el formato comercial pide 12, entonces se recorren por segunda vez los mismos 6 bits.

Así, se utilizan las variables de control a4,a3,a2,a1y a0 en forma conjunta con los registros de paginación p0, p1 y p2, para elegir la combinación y página adecuada para seguir el procedimiento descrito.

Para marcar el inicio del dato, la tabla que codifica éste posee dos puntos de entrada: el inicial, en donde marca el inicio de dato la combinación a4,a3,a2,a1 y a0 = 00 000 y para el segundo recorrido, la combinación a4,a3,a2,a1 y a0 = 01 000; para insertar la información

del bit de duplicación de teclas "shift", se intercambia esta información, con el ultimo bit por transmitir del segundo recorrido. Con esto, la información que lee el receptor es: 6 bits de información, 5 bits de redundancia y 1 bit de duplicación de teclas.

Por ultimo se indica que en las tablas de codificación comerciales, se emplean elementos de más para asegurar que los tiempos de intercambio y selección de página no afecten la señal de salida al receptor.

Una vez echo esto, falta escoger el emisor infrarrojo a utilizar, así como el detector óptico para la etapa de recepción; por tanto, del manual de optoelectrónica citado en la bibliografía se eligen los dispositivos, MLED91 para el emisor y MRD911 para el receptor.

El MLED91 es un led infrarrojo, dentro del rango de  $\lambda = 940$  nm. y posee además las características de ángulo de emisión de  $60^\circ$  y potencia de salida de 2.5 mW a 50 mA de corriente. Por su parte, el MRD911, es un fototransistor darlington NPN, que también se encuentra en el rango  $\lambda = 940$  nm y posee una corriente en iluminación, de 25mA para  $V_{CE} = 5V$  y  $H = 0.5$  mW/cm<sup>2</sup>.

El diodo emisor infrarrojo NO se puede conectar en forma directa al MAPL128, esto es debido a que el dispositivo mencionado sólo puede drenar corriente del orden de microamperes; por lo tanto, es necesario colocar un transistor como drenador de corriente.

Para la etapa de recepción se tiene que, por si mismo, el fototransistor no proporciona la señal suficiente para el manejo correcto del circuito. La solución fue emplear un amplificador operacional LM324 para obtener la amplificación nece-

saria para el funcionamiento adecuado.

Para el circuito mostrado se trabajó en forma satisfactoria hasta una distancia de aproximadamente 10 m. Existen amplificadores dedicados para el uso de fotosensores infrarrojos; el problema es que para el funcionamiento adecuado de estos últimos, es necesaria una fuente de 12V. Las ventajas que presentan los amplificadores dedicados son muchas, pero el sistema desarrollado esta atado a un sistema mínimo 80188 que por diseño posee solamente una fuente de 5V; para evitar el utilizar dos fuentes diferentes, o algún otro dispositivo para lograr los 12 V, se optó por el amplificador operacional descrito, LM324, que opera con una sola fuente de 5V.

---

### Conclusiones

---

Para este trabajo se utilizo un formato SONY, solo para ser consistentes con un control remoto comercial, sin embargo, no se tienen completamente las características de codificación de SONY, por lo que el control remoto NO ES compatible con SONY.

Para fines prácticos es más fácil el codificar solo los 6 bits de información que proporciona el teclado y funciona igualmente bien que con el formato aquí seguido. Si se desea mayor seguridad y además no se necesita de una señal de duplicación de teclas, seria conveniente el codificar 2 veces la misma información para averiguar si existió algún error en la transmisión.

Además, sin romper el esquema general, es posible el codificar un numero menor de teclas para aumentar líneas de control parecidas a "SHIFT" y así trabajar en base a las necesidades propias.

La elección del dispositivo MAPL 128 para el desarrollo de esta practica, brinda un espacio pequeño en tarjeta, además de consumir poca corriente y alargar con esto la vida de la batería empleada. El control remoto fue alimentado con una batería standard de 9 voltios y el teclado utilizado es una adaptación de una calculadora.

El ajuste de la señal de reloj se hizo en base a un dispositivo NE555 configurado como multivibrador asintótico con frecuencia variable.

Para terminar, se mencionará que las pruebas efectuadas con los dispositivos indicados en el diagrama brindaron un trabajo satisfactorio hasta para separaciones de 10m, si se apunta en forma directa al emisor del control remoto con el receptor.

## Control Remoto de Formato Comercial

```

PROGRAMA DE APLICACION

BEGIN HEADER

                                IPN

                                CENTRO DE INVESTIGACION TECNOLOGICA EN COMPUTACION

                                EDIFICIO DE GRADUADOS DE UPIICSA

                                CALLE TE NO. 950 COL. GRANJAS MEXICO, MEXICO D.F.
                                CP 08400

                                ALUMNO: AQUILINO CERVANTES AVILA

DESCRIPCION:
Se implementa un control remoto formato SONY, en un MAPL128

END HEADER

BEGIN DEFINITION
Device MAPL128;

Inputs  clk=1,tec,c0,c1,c2,c3,c4,c5,c6,c7,sht;
        {c0 - c7 son las columnas, sht permite duplicar las }
        {teclas, y tec es una retroalimentación externa de detec}

Feedbacks (jk,hold) b1,b2,b3,b4,b5,bs; {bits de captura de información b0 - b5 }
Feedbacks (jk,hold,set) out;           { salida del control remoto }
Outputs (reg) r0,r1,r2,r3,r4,r5,detec; { se declaran 6 renglones }
Feedbacks (buried) p0=37,p1=38,p2=39; { registros de paginación }
Feedback (jk,hold,buried) a0,a1,a2,a3,a4,bn,b0;
        { bits de control bs y bn son bits de corrimiento }

END DEFINITIONS

BEGIN TRUTH_TABLE {Esta tabla tiene el control de la tecla SHIFT }
TTIN sht,bs,p2,p1,p0; {la tecla shift maneja un registro de manera que}
TTOUT bs,p2,p1,p0;    {a cada cambio de "sht" a cero, el bit "bs" cambie}
                    { a su complemento. si "sht" se encuentra en uno }
1 - 000 ? 000      {el valor de "bs" es capturado}
01 000 0 000
00 000 1 000
END TRUTH_TABLE

BEGIN EQUATIONS { Esta ecuación detecta si se oprimió una tecla }
detec = c0 * c1 * c2 * c3 * c4 * c5 * c6 * c7;
        {detec se hace cero si cualquier c0..7 se vuelve cero }

END EQUATIONS

BEGIN TRUTH_TABLE { Esta tabla realiza a un contador binario de 3 bits atado }
                    { a un decodificador 3x8 }
TTIN tec,b2,b1,b0,p2,p1,p0;
TTOUT b2,b1,b0,r0,r1,r2,r3,r4,r5,p2,p1,p0;
0 --- 000 ??? ----- 001 {si "tec" es llevado a cero, entonces se detiene}

```

```

1 111 000 000 011111 000 {la cuenta y se pasa a la página siguiente}
1 000 000 001 101111 000
1 001 000 010 110111 000
1 010 000 011 111011 000 { ( " tec " se conecta a la salida " detec" ) }
1 011 000 100 111101 000
1 100 000 111 111110 000
1 101 000 111 111110 000
1 110 000 111 111110 000
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla declara un decodificador de prioridad}
TTIN tec,c0,c1,c2,c3,c4,c5,c6,c7,p2,p1,p0;
TTOUT b5,b4,b3,p2,p1,p0;

1 ----- 001 --- 000 { si la señal tec es igual a uno, no se pregunta}
0 ----- 0 001 000 010 {por prioridad de la tecla y se regresa a la página }
0 ----- 01 001 001 010 { cero la cuenta y se pasa a la página siguiente}
0 ----- 011 001 010 010 { se obtiene en b5,b4 y b3 la información de la }
                                { columna }
0 --- 0111 001 011 010 { si se reciben 2 o más señales en bajo, se acepta }
0 -- 01111 001 100 010 { la de mayor prioridad }
0 - 011111 001 101 010
0 - 0111111 001 110 010
0 01111111 001 111 010
END TRUTH_TABLE

{ A partir de esta tabla, se deja activo en la página 010 el registro de captura, y se }
{ utilizan las variables de control a2,a1 y a0 para saber que bit es el que se esta }
{ transmitiendo}

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b0 }
TTIN a2,a1,a0,b0,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b0 en bn, se capturan las variables de control,}
000 - 010 ? ??? 011 {y se cambia a la página 011, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b1 }
TTIN a2,a1,a0,b1,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b1 en bn, se capturan las variables de control,}
001 - 010 ? ??? 100 { y se cambia a la página 100, para codificación sony }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b2 }
TTIN a2,a1,a0,b2,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b2 en bn, se capturan las variables de control,}
010 - 010 ? ??? 101 {y se cambia a la página 101, para codificación SONY}
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b3 }
TTIN a2,a1,a0,b3,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0; { Se copia b3 en bn, se capturan las variables de }

```

## Control Remoto de Formato Comercial

```

011 - 010 ? ??? 110 {control, y se cambia a la página 110, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b4 }
TTIN a2,a1,a0,b4,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
        { Se copia b4 en bn, se capturan las variables de control,}
100 - 010 ? ??? 111 {y se cambia a la página 111, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b5 }
TTIN a2,a1,a0,b5,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
        { Se copia b5 en bn, se capturan las variables de control,}
101 - 010 ? ??? 100 {y se cambia a la página 100, para codificación }
END TRUTH_TABLE

{ El siguiente conjunto de tablas, son quienes codifican el bit bn en el formato: }
{ 3 espacios = inicio; 2 espacios = BAJO y 1 espacio = ALTO. }

BEGIN TRUTH_TABLE { Esta tabla manda la señal de inicio, y codifica el bit b0 }
        { en formato comercial }
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT a3,a2,a1,a0,out,p2,p1,p0;

0000 - 1 011 0001 0 011
0001 - 0 011 0010 0 011 { las variables de control a2,a1 y a0 se utilizan para}
0010 - 0 011 0011 0 011 { simular un registro de corrimiento y junto con a3 }
0011 - 0 011 0100 0 011 {garantizar que no haya combinaciones repetidas }
0100 - 0 011 0101 0 011
0101 - 0 011 0100 0 011 {se manejan 6 posiciones en cero, para lograr que el }
0100 - 0 011 0111 1 011 {espacio más pequeño, sea igual al tiempo de retorno}
0111 - 1 000 1000 1 011 { a uno. }
1000 0 1 011 1001 0 011
1001 0 0 011 1010 0 011 {si bn es cero, entonces se mandan 4 ceros y se }
1010 0 0 011 1011 0 011 { regresa a la página 010 para tomar el siguiente bit }
1011 0 0 011 1100 0 011
1100 0 0 011 0001 1 010
1000 1 1 011 1001 0 011 {si bn es uno, entonces se mandan 2 ceros y se }
1001 1 0 011 1010 0 011 { regresa a la página 010 para tomar el siguiente bit }
1010 1 0 011 0001 1 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b1 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0001 0 1 100 0 1010 100
1010 0 0 100 0 1011 100 {si bn es cero, entonces se mandan 4 ceros y se }
1011 0 0 100 0 1100 100 { regresa a la página 010 para tomar el siguiente bit }
1100 0 0 100 0 1101 100
1101 0 0 100 1 0010 010
0001 1 1 100 0 1010 100 {si bn es uno, entonces se mandan 2 ceros y se }
1010 1 0 100 0 1011 100 { regresa a la página 010 para tomar el siguiente bit }

```

```

1011 1 0 100 1 0010 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b2 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0010 0 1 101 0 1011 101
1011 0 0 101 0 1100 101 {si bn es cero, entonces se mandan 4 ceros y se }
1100 0 0 101 0 1101 101 { regresa a la página 010 para tomar el siguiente bit}
1101 0 0 101 0 1110 101
1110 0 0 101 1 0011 010
0010 1 1 101 0 1011 101 {si bn es uno, entonces se mandan 2 ceros y se }
1011 1 0 101 0 1100 101 { regresa a la página 010 para tomar el siguiente bit}
1100 1 0 101 1 0011 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b3 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0011 0 1 110 0 ? 1100 110
1100 0 0 110 0 ? 1101 110 {si bn es cero, entonces se mandan 4 ceros y se }
1101 0 0 110 0 ? 1110 110 { regresa a la página 010 para tomar el siguiente bit}
1110 0 0 110 0 ? 1111 110
1111 0 0 110 1 ? 0100 010
0011 1 1 110 0 ? 1100 110 {si bn es uno , entonces se mandan 2 ceros y se }
1100 1 0 110 0 ? 1101 110 { regresa a la página 010 para tomar el siguiente bit}
1101 1 0 110 1 ? 0100 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b4 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0100 0 1 111 0 1101 111
1101 0 0 111 0 1110 111 {si bn es cero , entonces se mandan 4 ceros y se }
1110 0 0 111 0 1111 111 { regresa a la página 010 para tomar el siguiente bit }
1111 0 0 111 0 1000 111
1000 0 0 111 1 0101 010
0100 1 1 111 0 1101 111 {si bn es uno , entonces se mandan 2 ceros y se }
1101 1 0 111 0 1110 111 { regresa a la página 010 para tomar el siguiente bit }
1110 1 0 111 1 0101 010
END TRUTH_TABLE

{ Debido a que se requieren 12 pulsos de información, se recorren nuevamente los }
{ 5 primeros bits de información, así, se codifica solo una vez el bit b5 }

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b5 en formato SONY}
TTIN a4,a3,a2,a1,a0,bn,out,p2,p1,p0; {se agrega a4 como indicador de recorrido}
TTOUT out,a4,a3,a2,a1,a0,p2,p1,p0;
        { si a4 se encuentra en cero, entonces se codifica el bit b5 y }
1 0101 - 1 100 0 1 0000 101 {se cambia de página para un segundo recorrido,}
0 0101 0 1 100 0 ? 1110 100 { Si a4 esta en uno, entonces se cambia a la }
        { página 101 sin transmitir b5}

```

```
0 1110 0 0 100 0 ? 1111 100 {si bn es cero , entonces se mandan 4 ceros y }
0 1111 0 0 100 0 ? 1000 100 { se regresa a la página 010 para tomar el sig. bit}
0 1000 0 0 100 0 ? 1001 100
0 1001 0 0 100 1 1 1000 010
0 0101 1 1 100 0 ? 1110 100 {si bn es uno , entonces se mandan 2 ceros y se }
0 1110 1 0 100 0 ? 1111 100 { regresa a la página 010 para tomar el siguiente bit}
0 1111 1 0 100 1 1 1000 010
END TRUTH_TABLE

{ Para el pulso número doce, se envía la información contenida en el bit Shift para}
{ duplicar el número de teclas }

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit "Shift" bs en formato SONY}
TTIN a4,a3,a2,a1,a0,bs,out,p2,p1,p0; { Debido a ajuste de tiempos, y conside-}
TTOUT out,a4,a3,a2,a1,a0,p2,p1,p0; { randodo que este es el ultimo bit por tras-}
{ mititr, para este bit (bs) se hace una asignación menos a uno o a cero }
{ en la codificación comercial.}
1 0000 0 0 101 0 1 0001 101
1 0001 0 0 101 0 1 0011 101 {si bs es cero , entonces se mandan 3 ceros y se }
1 0011 0 0 101 0 1 0100 101 { regresa a la página 000 para permitir que otra }
1 0100 0 0 101 1 0 0000 000 { tecla sea aceptada }
1 0000 1 0 101 0 1 0001 101 {si bn es cero , entonces se mandan 4 ceros y se }
1 0001 1 0 101 1 0 0000 000 { regresa a la página 000 para permitir que otra }
END TRUTH_TABLE { tecla sea aceptada }
```

### **Bibliografía**

- [1] Programmable Logic Devices Databook and design guide. National Semiconductor.
- [2] OPAL TM, Manual de usuario.
- [3] Optoelectronics device data. Motorola.
- [4] Operational Amplifiers data book. National Semiconductor.

# Sistema de Control de Lógica Difusa Aplicado a un Motor de C.D.

M. en C. Romeo Urbieto Parrazales  
Profesor e Investigador del CINTEC-IPN.

El presente artículo describe el diseño y simulación de un Sistema de Control de Lógica Difusa aplicado a la variable de posición de un motor de C.D. de 1/15 de H.P. En el diseño se utilizó el método de Sistemas de Estructuras Variables para obtener la estabilidad de la variable de posición. La simulación presenta gráficos de la superficie de Control de Lógica Difusa, y la variable de posición para 25 y 11 reglas.

## Introducción

Recientemente se ha desarrollado un gran interés en el uso de la teoría de los conjuntos difusos para modelar sistemas de control complejos, debido a que los conjuntos difusos son empleados por los seres humanos cotidianamente en el manejo de su cuerpo y de su entorno [1].

Los términos: *pequeño*, *medio*, *grande*, etc. son expresiones humanas usadas cotidianamente para modelar las variables de entrada y salida de un sistema a manejar [2].

Este manejo queda formulado por un modelo de reglas en lenguaje humano. Por ejemplo, si se quiere mover una carga una cierta distancia, la regla sería: Si la distancia es

*pequeña*, la fuerza necesaria debe ser *pequeña*. Dicho en términos de control, si la variable controlada de posición es *pequeña*, la variable manipulada de posición es *pequeña*.

La metodología de trasladar el modelo de las reglas a una estrategia de control con conjuntos y lógica difusa se conoce bajo el nombre de *Control de Lógica Difusa (FLC)* [3].

Un FLC consiste de tres partes principales: las variables de entrada, la base de reglas, y las variables de salida.

La base de reglas origina la estabilidad de la variable de posición y se determina por medio del método de Sistemas de Estructuras Variables.

## Descripción

El FLC básicamente está constituido por dos componentes [4] como se muestra en la **figura 1**.

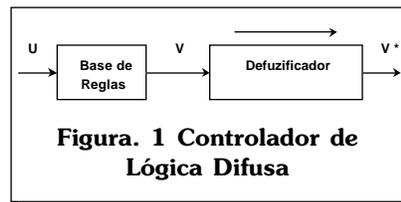


Figura. 1 Controlador de Lógica Difusa

## Base de Reglas

El conocimiento del controlador viene expresado por un conjunto de reglas de tipo granular, de la manera siguiente: Si  $f: X \rightarrow Y, y = f(x)$ , donde  $X$  y  $Y$  son bases de los conjuntos  $U$  y  $V$ , entonces, se puede representar el mapeo como una relación  $F$  en  $X \times Y$ , tal que  $(x,y) \in F$  si  $f(x) = y$ .

$$F = \{ (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \} \quad (1)$$

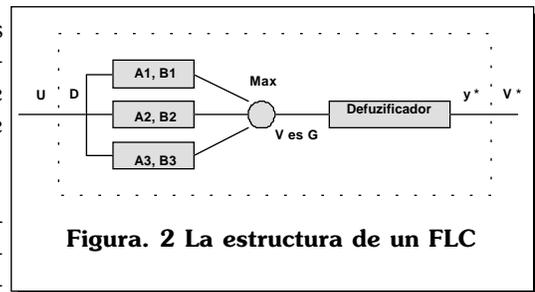


Figura. 2 La estructura de un FLC

donde cada par  $(x_i, y_i)$  es una solución de  $f, f(x_i) = y_i$ , **figura 2**.

Si se denota el par como  $P_i$

$$P_i = \{(x_i, y_i)\} = A_i \times B_i = (A_i \cap B_i) = \text{Min}(A_i(x), B_i(y)) \quad (2)$$

Entonces la relación  $F$  se puede denotar como una unión de pares  $P_i$

$$F = \cup_i P_i, \quad (3)$$

Cada módulo  $P_i$  puede interpretarse semánticamente en la forma

Cuando  $U$  es  $A_i$ , este es el caso de que  $V$  es  $B_i$ .

donde  $A_i$  y  $B_i$  son los conjuntos difusos de entrada y salida del FLC. Si la entrada es una proposición,  $D$  es  $U$ , y el mapeo se puede representar por una variable  $H$ .

$$H = (U, V) = \cup_i P_i \cap D = \text{Max}_i [A_i(x) \wedge B_i(x) \wedge D(x)] \quad (4)$$

A fin de tener el valor de  $V$  se puede tomar la proyección en  $V$ , de esta manera, la salida  $V$  es  $G$ .

$$G(y) = \text{Max}_x [H(x, y)] = \text{Max}_i [\text{poss}[A_i | D] \wedge B_i(y)] \quad (5)$$

**Defuzificación**

La defuzificación es el paso final del FLC, y es el uso de la salida  $G$  para seleccionar un "singleton" o punto de membresía  $y^* \in Y$ , como la salida del controlador.

Usando el procedimiento de defuzificación de Area Central, se llega finalmente a la ley de control  $u(k)$ , donde  $m$  es el no. de reglas empleadas.

$$y^* = \frac{\sum_{i=1}^m G(y_i) y_i}{\sum_{i=1}^m G(y_i)} = u(k) \quad (6)$$

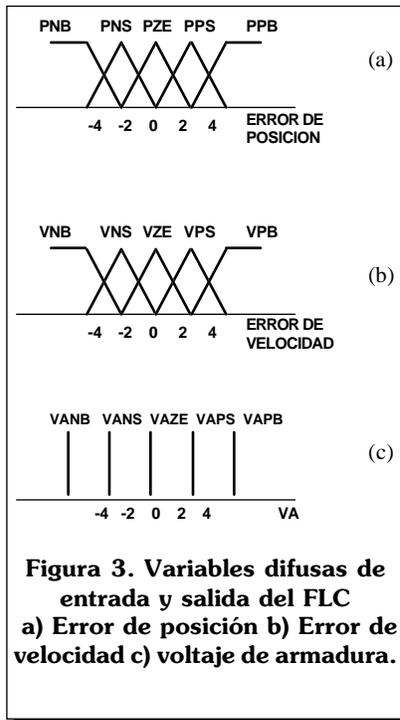
**Diseño**

El diseño del FLC se plantea de la siguiente forma:

Las funciones de membresía del FLC tienen un universo de -4 a 4 (error de posición {-377 a 377 rad}, error de velocidad {-675 a 675 rad/seg}, voltaje de armadura -24 a 24 voltios), **figura 3**.

**Base de Reglas**

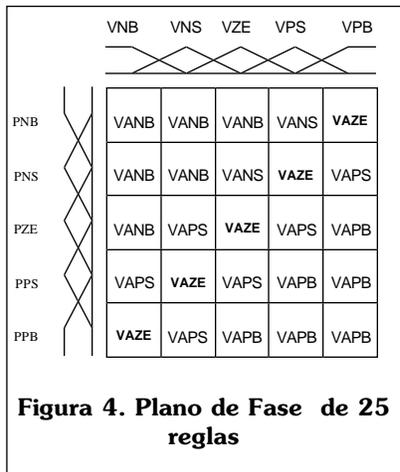
La base de reglas se presenta como las relaciones difusas,  $R_i$ : Si



**Figura 3. Variables difusas de entrada y salida del FLC a) Error de posición b) Error de velocidad c) voltaje de armadura.**

$(A_i, B_i)$  entonces  $C_i$ , donde  $A_i$  y  $B_i$ , son variables difusas de entrada (error de posición difuso y error de velocidad difuso) que determinan la ley de control difuso  $C_i$  (voltaje de armadura difuso) [5]. La base de reglas  $R_i$  constituye una "estructura" del FLC que se cambia de acuerdo a los estados del proceso.

El plano de fase queda determinado por las dos entradas y la salida. La matriz formada es de 5x5 o sea

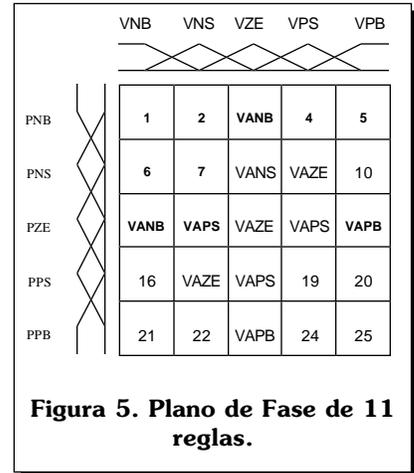


**Figura 4. Plano de Fase de 25 reglas**

25 reglas **If-Then**. En el plano de fase anterior se repiten las reglas: 1, 2, 4, 5, 6, 7, 10, 16, 19, 20, 21, 22, 24 y 25. Observe la **figura 4**.

**Defuzificación**

La defuzificación de la variable



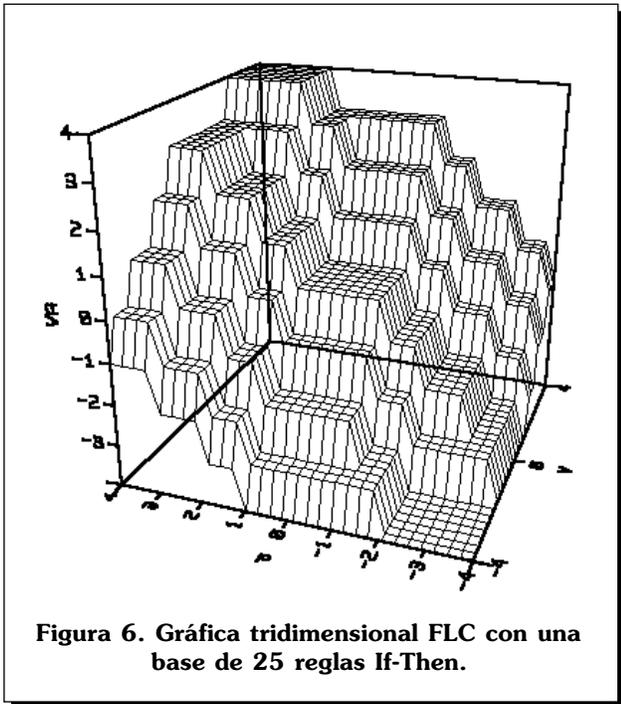
**Figura 5. Plano de Fase de 11 reglas.**

$VA$  se obtiene aplicando la formula (6), en la que los pesos de  $VA$  son "puntos" (singletons) unitarios.

**Simulación**

La simulación del FLC se muestra en las **figuras 6** y **7**. Estas gráficas se obtuvieron por medio de un programa de simulación de controladores FLC, el FLD (Fuzzy Logic Designer). En las gráficas se pueden observar las dos variables entrada difusas y la salida difusa del controlador de Lógica Difusa. En la **figura 6** se observa una superficie de cambio mas fina, no así en la **figura 7**, donde el cambio es mas abrupto.

El FLD genera un programa y un archivo de encabezado de la operación del controlador, codificado en lenguaje C++, los cuales son Motor.C y Motor.h Las subrutinas importantes en motor.C son:



## Conclusiones

El diseño de los parámetros del sistema de control difuso de posición resulta sumamente simple para su cálculo e implantación en una computadora personal.

Los resultados de la simulación determinan que las respuestas de la posición de la flecha del motor de c.d. se estabilizan muy bien en el rango de operaciones de posición deseados.

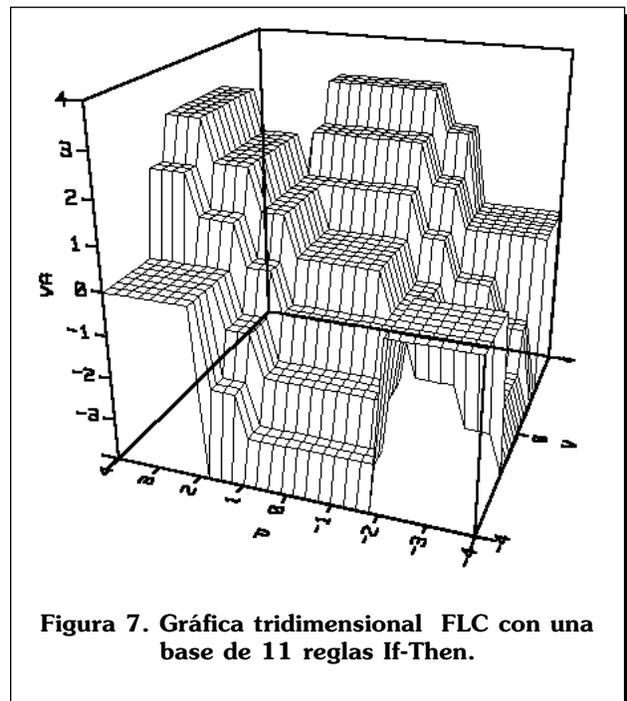
`fld_fuzzy_inputs`(float pos, float vel)  
`fld_fuzzy_outputs` (float pos, float vel)

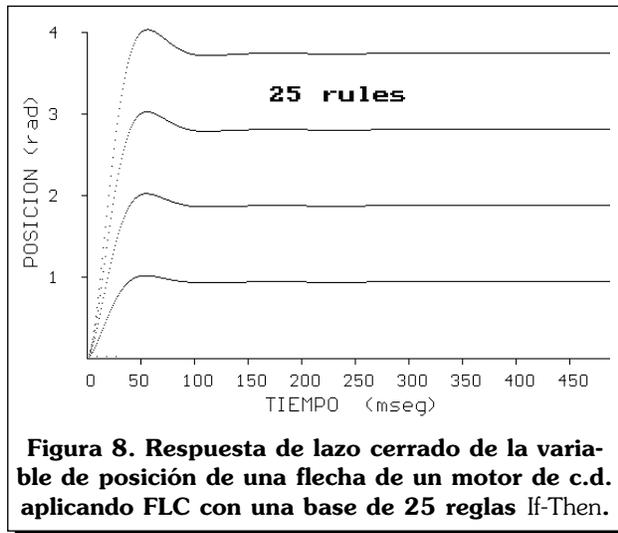
La primera evalúa los parámetros de la variables difusas de entrada. La segunda calcula los parámetros de salida del controlador.

## Resultados de la Simulación

Las subrutinas antes mencionadas se insertan en un programa de simulación que incluye también el modelo matemático del motor de c.d. [6,7] y su respectiva retroalimentación, conformando de esta manera un sistema de control FLC de lazo cerrado.

La variable simulada es la posición de la flecha en un tiempo de 500 mseg. El transitorio se observa hasta los 100 mseg. Las gráficas presentes son para un FLC con una base de 25 y 11 reglas, **figuras 8 y 9**.





**Referencias**

[1] Witold Pedrycz. Fuzzy Control and Fuzzy Systems. Research Studies Press Ltd., 1993, ch.4, pp. 103-106.

[2] Li-Xi Wang. Adaptive Fuzzy Systems and Control. Prentice Hall. Cap.1,2 pp. 1-28.

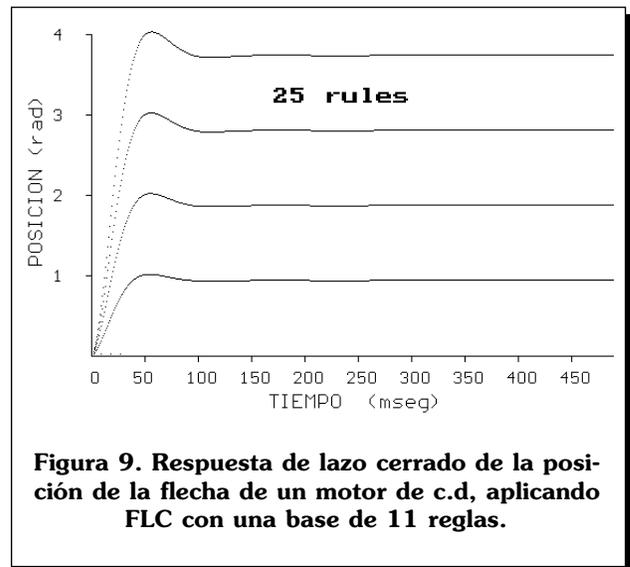
[3] Shigeyasu Kawaji and Nobutomo Matsunaga. "Fuzzy Control off VSS and its Robustness". Kandel and Langholz. Fuzzy Control Systems. CRC Press, 1994, ch. 10, pp.223-242.

[4] Ronald R. Yager. "Alternative Structures for Knowledge Representation in Fuzzy Logic Controllers". Fuzzy Control Systems. CRC press. pp 100-136.1993.

[5] Arthur Ramer. "Information Complex and Fuzzy Control". Fuzzy Control System. CRC press. pp.76-95.1993.

[6] Romeo U. Parrazales. "Simulación de la Posición de un Motor de d.c. de 1/15 de H.P Controlador por un Algoritmo PI usando una Microcomputadora IPNe16-m." Revista Polibits. Vol1. Num.9 .pp 49-62, CINTEC IPN. 1992.

[7] Timothy A. Adcock. Digital Signal Processing. Semiconductor Group. Texas Instrument Incorporated. pp 1-37.1993.



# Generador de Patrones de Televisión

*C. José Anibal Arias Aguilar  
C. Osvaldo Espinosa Sosa  
Alumnos de la Maestría del CINTEC.*

*M. en C. Juan Carlos González Robles  
Jefe del Departamento de Producción.*

**A** continuación se describe el diseño de un circuito generador de patrones de televisión, el cual genera una señal de video compuesto que permite hacer pruebas de ajuste en un tubo de rayos catódicos ("Catode Ray Tube", CRT).

## Introducción

Un CRT produce una imagen de video mediante la proyección y barrido de un haz de electrones sobre una pantalla cubierta con fósforo. Para la exploración de una imagen por el haz de electrones en un CRT se desvía el haz de su trayectoria a un lado y otro de la capa fotoconductora y, al mismo tiempo, se le hace descender, de forma que la recorra por completo.

Existen dos formas de desviar un haz de electrones de su trayectoria: una mediante un campo eléctrico, y otra mediante un campo magnético. La desviación por campo eléctrico está basada en el principio de que cargas eléctricas del mismo signo se repelen y cargas eléctricas de signo opuesto se atraen.

La desviación por campo magnético se basa en el principio de que toda corriente eléctrica viene acompañada de un campo magnético, y que en éste polos del mismo signo se repelen y polos de signo opuesto se atraen.

Así, si se somete un haz de electrones procedente del cátodo de un tubo de rayos catódicos a la influencia de un campo eléctrico o magnético, éste se desplazará de su trayectoria en mayor o menor grado y en un sentido o en otro, según sea el sentido e intensidad de dichos campos.

## Barrido Horizontal y Vertical

Como un punto arbitrario de la pantalla a un momento determinado en el tiempo, se asume que el haz de electrones está localizado en la esquina superior de la pantalla (**Figura 1**). Bajo la influencia electromagnética del yugo de deflexión horizontal, el haz es movido de izquierda a derecha de la pantalla. Al mismo tiempo, bajo la influencia del yugo vertical es impulsado hacia abajo, produciendo una línea de barrido visible moviéndose hacia abajo al extremo derecho. En este punto, el haz es movido rápidamente al extremo izquierdo de la pantalla, produciéndose retraso horizon-

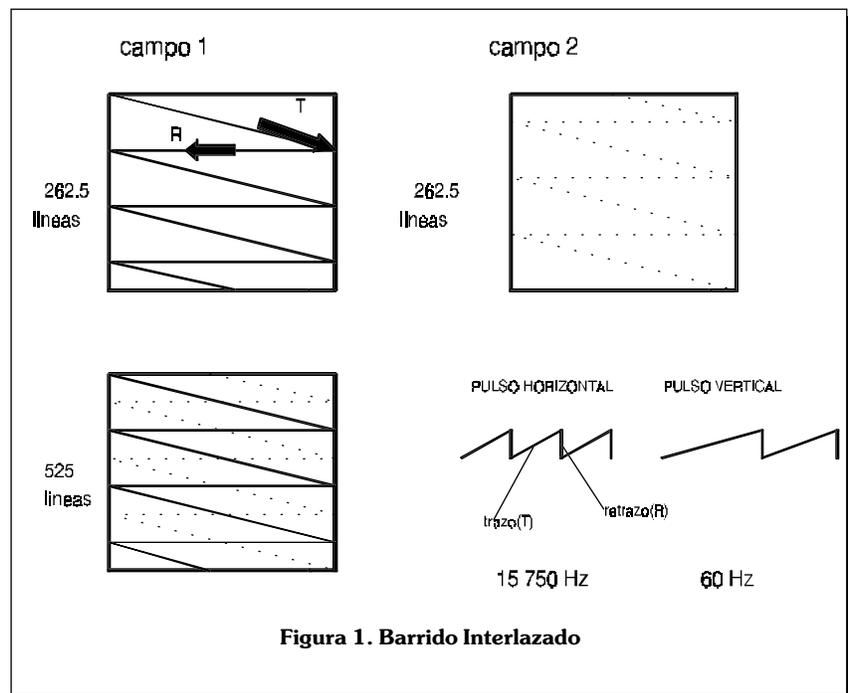


Figura 1. Barrido Interlazado

tal. Las líneas de retraso no son vistas en la pantalla.

Después de 262.5 líneas barridas, la influencia magnética del yugo vertical se invierte y hace que el haz retorne a la parte superior y *continúe* su movimiento hacia la derecha y abajo. Al siguiente retraso vertical tenemos lo siguiente: antes del primer retraso se originaron líneas de barrido «pares», y durante el segundo líneas «impares». Los dos tipos de líneas son entrelazadas, lo cual resulta en una trama de 525 líneas que barren cada elemento de imagen en la pantalla.

La frecuencia de barrido vertical monocromo es de 15,760 Hz (para color es de 15,736 Hz). La frecuencia vertical es de 60 Hz. Como existen dos campos por cada trama, la velocidad de repetición de la imagen es de 30 por segundo -lo suficientemente rápido para prevenir el parpadeo.

### Señal de imagen. Sincronía.

Para generar la señal de video, la escena o imagen a transmitir es reproducida sobre la placa fotoconductora de un tubo explorador de imagen. En este tubo dicha imagen se transforma en impulsos eléctricos; es decir, durante el proceso de exploración se convierten las diferencias de iluminación de la imagen en diferencias de voltaje que aparecen a la salida del tubo de exploración.

La correspondencia entre los valores de tensión obtenidos y los valores de iluminación dependen de la clase de tubo utilizado en la cámara y de su circuito externo. Así, en algunos tubos el voltaje de salida puede ser elevado cuando se explora un punto de imagen oscuro y bajará cuanto mayor sea la iluminación del punto de imagen. En otros

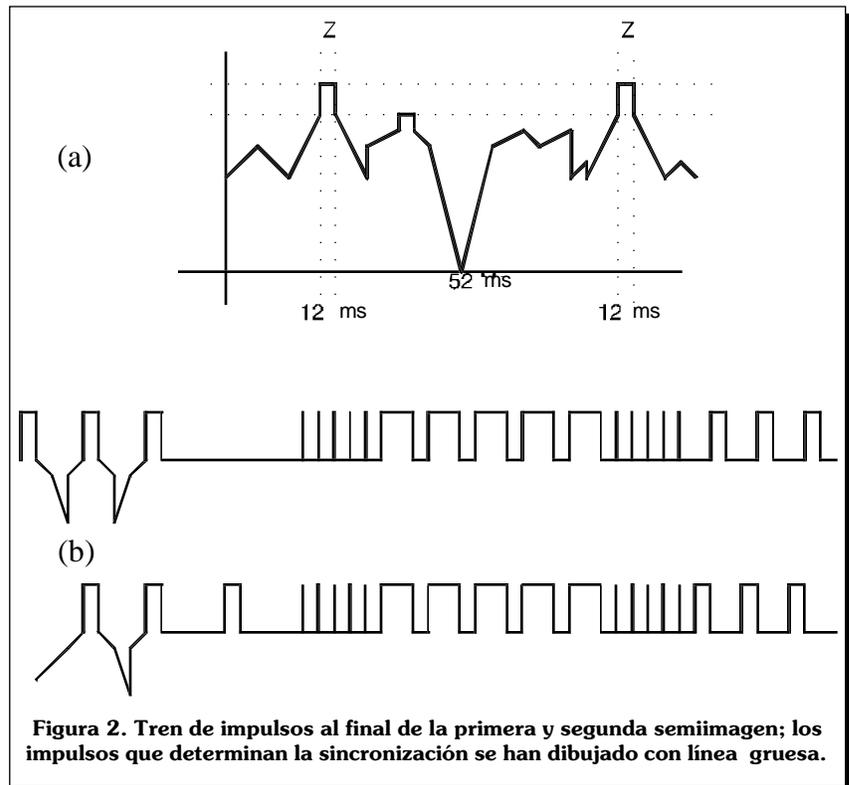
tubos sucede exactamente lo contrario.

En general, las imágenes que se transmiten son complejas, por lo que se generan formas de onda de voltaje complejas según el nivel de iluminación de cada punto. En la **figura 2 (a)** se muestra una señal de video, donde se observa que el tiempo de exploración horizontal es de 52 ms y el tiempo de retraso es de 12 ms. En el tiempo de retraso aparece, como es lógico, un voltaje de salida que resulta indeseado y por ello se elimina mediante un impulso de bloqueo. Durante este tiempo el tubo explorador suministra un voltaje cuyo valor corresponde al límite del negro.

Durante el transcurso del periodo de retraso de línea se mezcla, además, un impulso de sincronización de líneas Z, el cual anuncia al receptor la transmisión de la próxima línea. De esta forma la explora-

ción de líneas en el receptor se inicia de forma sincrónica con la exploración de líneas en el tubo de la cámara del emisor.

Este impulso de sincronización de líneas se halla superpuesto al límite de negro que se presenta durante el tiempo de retroceso. Así, si al voltaje máximo del impulso de sincronización de líneas Z se le asigna el valor de 100%, entonces resulta que el límite del negro está al 75% y el límite del blanco al 10% de dicho valor. Entonces es válido decir que el impulso de sincronización es más negro que el negro. Cuando el haz electrónico explorador del tubo de imagen en el centro emisor llega al borde inferior de dicha imagen, entonces debe añadirse una señal de sincronización para el retroceso de imagen, es decir una señal de sincronización que haga que el punto explorador retorne rápidamente al borde superior de la pantalla.



**Figura 2. Tren de impulsos al final de la primera y segunda semiimagen; los impulsos que determinan la sincronización se han dibujado con línea gruesa.**

Dado que el retroceso de imagen dura por lo menos 20 veces el tiempo invertido en el retroceso de una línea, resulta que para esta señal de sincronía se dispondrá de mucho más tiempo que para el retroceso de línea. Por estos motivos el cambio de imagen se anuncia con unos largos impulsos marcados con Y en la **figura 2 (b)**, los cuales tienen una duración total de 144 ms. Como se observa, esta duración es bastante más larga que la del impulso de líneas Z, que tan sólo dura 6 ms.

Para que durante los impulsos de imagen no se pierda la sincronización de las líneas, se añaden impulsos de línea sobre los impulsos de imagen Y. Así, en la parte superior de la figura 2 (b) se puede ver la sucesión de impulsos originados por el retroceso de imagen al final de la primera semiimagen. Antes y después de los impulsos de sincronización de imagen Y se dispone una serie de impulsos muy cortos de compensación designados como impulsos precursores YP e impulsos sucesores YS.

Estos impulsos precursores y sucesores evitan que los impulsos de imagen en el receptor no queden influenciados por los impulsos de línea. Volviendo a la figura 2 (b), en ella se puede observar que los impulsos de sincronización de imagen son los mismos en ambas semiimágenes, con la única diferencia de que el principio de la segunda semiimagen está desplazado media línea con respecto a la primera semiimagen con el fin de obtener la exploración entrelazada.

### Descripción del Circuito

La importancia de este circuito radica en que se usó un dispositivo MAPL/128 (Arreglo múltiple de ló-

gica programable), que ofrece la ventaja de sustituir a múltiples dispositivos de lógica tradicional o lógica programable; debido a esto disminuye el área física de la tarjeta del circuito impreso se reduce el consumo de corriente de la fuente de poder empleada.

### Características del MAPL.

El MAPL/128 pertenece a la tecnología EECMOS y posee una arquitectura de arreglos lógicos programables integrando múltiples FPLA'S (Arreglo lógico programable en campo), lo que permite facilitar el diseño de máquinas de estado, controladores, secuenciadores de microinstrucciones, interfaces y en general cualquier diseño lógico secuencial. Las características que posee son las siguientes:

- Alta densidad, con arquitectura PLA.
- Velocidades de operación de 33, 40 y 45 MHz.
- Bajo consumo de potencia.
- Borrable eléctricamente.
- 100% de funcionalidad de prueba.
- Lógica reconfigurable instantáneamente.
- Capacidad mínima de 100 ciclos de programación-borrado.
- 27 macroceldas con registros tipo DE, JK, RS o T.
- Precarga asíncrona, y capacidad de reinicio.
- Soportado por OPAL y por software popular dedicado al tema.

La arquitectura MAPL es funcionalmente equivalente a un FPLA grande, teniendo un total de 128 productos términos. En realidad, el MAPL/128 tiene ocho planos o páginas FPLAS, consistiendo cada página de un arreglo AND y un arreglo OR, ambos programables.

Cada arreglo AND posee 58 entradas y 16 términos producto, los cuales pueden conectarse a cualquiera de las 54 sumas de términos, por lo que cada página tiene una configuración de 58 X 16 X 54.

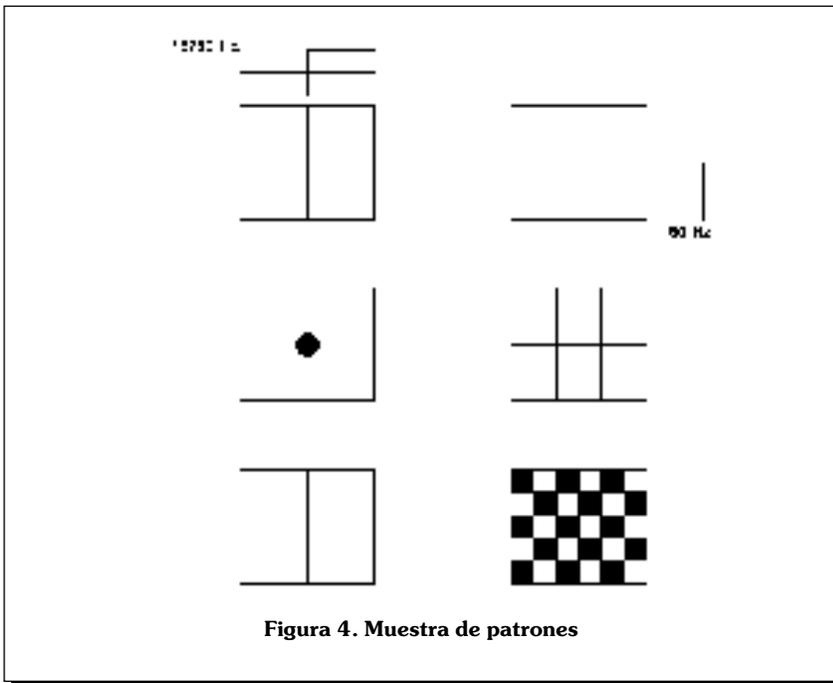
Tres registros internos manejan la paginación, teniendo una página activa en un momento dado. Al igual que los demás registros internos, estos pueden usarse también para almacenar bits.

Las páginas se seleccionan en el momento de la aplicación; las salidas de las macroceldas son controladas por reloj y también lo son los registros de página. Lo anterior tiene el efecto de preseleccionar la página donde se encuentra el siguiente estado lógico. Debido a que el dispositivo posee arquitectura PIPELINE, la siguiente página no acumula retardo, con lo que la paginación aparece transparente al usuario.

En general, los dispositivos MAPL incluyen una estructura de macrocelda flexible, configurable por el usuario. La misma macrocelda registrada se emplea para las funciones de salida, retroalimentación o entrada/salida. Cada macrocelda puede ser configurada como registro JK o D; ambos tipos se implementan en hardware y no requieren transformaciones por software o por lógica tradicional. Todas las macroceldas se sincronizan por el flanco de subida de la señal de reloj.

Las macroceldas de salida presentan una señal de habilitación (OE) que permite llevar al dispositivo a un estado de alta impedancia. Es posible utilizar esta terminal en la función indicada o como entrada al arreglo AND; además se tiene que las macroceldas de salida pueden ser configuradas como activas en alto o en bajo.





```

begin header
  IPN-CINTEC
  autor: OSVALDO ESPINOSA SOSA
  Circuito generador de patrones.
end header

begin definition

  device MAPL128 ;
  input CLK ;
  inputs sel1=5, sel2=4, sel3=3, sel4=2,
  qd1=6, qd2=7, qd3=8, qd4=9, qd5=10,
  qd6=11, qd7=23, qd8=22, qd9=21, qd10=20,
  qd11=19, qd12=18 ;
  feedbacks (jk, hold, buried) qc1=29, qc2=30,
  qc3=31, qc4=32, qc5=33, qc6=34, qc7=35 ;

  outputs sal1=12, sal2=13, sal3=15, sal4=16 ;
  set selec = [ sel1,sel2,sel3,sel4 ] ;

end definition

begin equations

  inter1 = qc1 * qc2 * qc3 * qc4 ;
  E0 = inter1 * qc5 * qc6 * qc7 ;
  inter2 = qd5 * qd6 * qd7 * qd8 ;
  E1 = inter2 * qd9 * qd10 * qd11 ;
  E2 = E0 + E1 ;
  E3 = E0 * E1 ;
  
```

```

  E4 = inter1 ;
  inter3 = qd5 * qd6 * qd7 ;
  E5 = inter3 * qd9 * qd8 ;
  E6 = inter1 + E5 ;
  E7 = inter1 * E5 ;
  {
    inter4 = ( qd3 * /qd4 ) + ( /qd3 * qd4 ) ;
    inter5 = ( qd11 * /qd12 ) + ( /qd11 * qd12 ) ;
    E8 = inter4 + inter5 ;
    E9 = inter4 * inter5 ;
    inter6 = ( qd1 * /qd10 ) + ( /qd1 * qd10 ) ;
    E10 = E9 * inter6 ;
    E11 = inter6 ;}
  E12 = qd5 ;
  E13 = qd9 ;
  E14 = qd5 + qd9 ;
  {
    E15 = qd5 * qd9 ;
  }
  sal1 := E0 * ( selec == 0 )
  + E1 * ( selec == 1 )
  + E2 * ( selec == 2 )
  + E3 * ( selec == 3 )
  + E4 * ( selec == 4 )
  + E5 * ( selec == 5 )
  + E6 * ( selec == 6 )
  + E7 * ( selec == 7 ) ;
  sal3 := E13 * ( selec == 13 )
  + E12 * ( selec == 12 )
  + E14 * ( selec == 14 ) ;
  sal4 := qc4 ;
  inter7 = qd1 * qd2 * qd3 * qd4 ;
  
```

```

inter8 = qd7 * qd8 * qd9 * qd10 ;
sal2 := inter7 + ( inter8 * qd11 * qd12 ) ;

end equations

begin truth_table

ttin qc1,qc2,qc3,qc4,qc5,qc6,qc7 ;
ttout qc1,qc2,qc3,qc4,qc5,qc6,qc7 ;

——— ———!
———1 ——!-
——11 ——!—
——111 ——!—
—1111 ——!——
—11111 ——!——
-111111 !———

end truth_table

begin vectors

CLK, sel1, sel2, sel3, sel4, qd1, qd2, qd3, qd4, qd5,

```

```

qd6, qd7, qd8, qd9, qd10, qd11, qd12 ;

c 0100 000000000000
c 0100 000000000001
c 0100 000000000010
c 0100 000000000011
c 0100 000000000100
c 0100 000000000101
c 0100 000000000110
c 0100 000000000111
c 0100 000000001000
c 0100 000000001001
c 0100 000000001010
c 0100 000000001100
c 0100 000000001101
c 0100 000000001110
c 0100 000000001111
c 0100 000000010000
c 0100 000000010001
c 0100 000000010010
c 0100 100000010011

end vectors

```

---

### ***Bibliografía***

---

Enciclopedia de la radio, Televisión y Hi-Fi. Francisco Ruiz Vasallo. Ediciones CEAC.

MAPL Design Guide. National Semiconductor. 1993.

OPAL, Software User's Manual. National Semiconductors. 1993.

## Reporte de Actividades realizadas en el CINTEC durante el período del 31 de Agosto de 1994 al 1o. de Septiembre de 1995

El presente informe abarca el período comprendido entre el 31 de Agosto de 1994 al 1o. de Septiembre de 1995, cubriendo el espacio de los Semestres de Otoño de 1994, Primavera 1995 y el inicio del semestre Otoño de 1995. Este informe está estructurado con base en las actividades sustantivas del Centro, las cuales son Docencia, Investigación y Desarrollo Tecnológico y las actividades desarrolladas alrededor del equipo instalado en el Programa de Autoequipamiento en Materia de Computo, como son Mantenimiento, actualización de equipo, etc.. En todos los casos se indica la Subdirección responsable de las acciones y en lo posible el departamento correspondiente.

### DOCENCIA

La Maestría en Ingeniería de Cómputo tiene como especialidad la de Sistemas Digitales. Fue iniciada en el semestre de Otoño de 1988, como consecuencia de los objetivos de creación del Centro. El desarrollo del programa es responsabilidad de la Subdirección Académica y los departamentos asociados a esta Subdirección.

El programa actualmente está estructurado por siete materias obligatorias, dos asignaturas electivas de un total de ocho que se ofrecen actualmente y tres seminarios departamentales. La carga de trabajo se complementa con un trabajo de Tesis. De acuerdo con el programa académico el alumno deberá complementar su formación participando en sesiones de laboratorio y apoyando proyectos de investigación. Anteriormente se tenían siete materias optativas y durante el periodo reportado se registro ante DEPI la materia de **Arquitecturas Paralelas**.

Durante este período se han sucedido 3 diferentes semestres lectivos. La carga académica de estos semestres fue cubierta tanto con la impartición de materias de la Maestría del CINTEC, como apoyando cursos de nivel licenciatura y Maestría en la UPIICSA. El tiempo aquí manifestado es el tiempo de clases frente a grupo, teniéndose que considerar un tiempo semejante para preparación de clases, atención de alumnos, etc. La distribución de la carga académica se dio como sigue:

Otoño 1994	⇒	70 Hrs./Sem. en CINTEC	/	24.5 Hrs./Sem. en <b>apoyo</b> a UPIICSA
Primavera 1995	⇒	59 Hrs./Sem. en CINTEC	/	17 Hrs./Sem. en <b>apoyo</b> a UPIICSA
Otoño 1995	⇒	90 Hrs./Sem. en CINTEC	/	31 Hrs./Sem. en <b>apoyo</b> a UPIICSA

El programa cuenta con un promedio de 67 alumnos, con base a un proceso de admisión anual. Durante este periodo se han titulado 3 alumnos y se han registrado 9 nuevos temas de Tesis haciendo un total de 23 temas de Tesis registrados. Durante este periodo 16 alumnos terminaron el 100% de créditos de la Maestría por materias, siendo susceptibles de graduarse en el corto plazo, teniéndose además un promedio de 8 alumnos como becarios PIFI durante el período reportado.

Como parte de las medidas de apoyo Docente que la Subdirección Académica de este Centro considera importante esta la inscripción al Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) de la totalidad de los alumnos de tiempo completo del Centro, y de los Profesores titulares y asociados del mismo. Esto se realizo a finales del Semestre Primavera 1995, pretendiendo continuar con este apoyo en forma constante.

En materia de recursos humanos, 6 profesores del Centro se encuentran desarrollando programas Doctorales, de los cuales 1 es Candidato a Doctor y los demás se encuentran en la fase de Investigación Documental. Otros 2 profesores se encuentran cursando estudios de Maestría y uno de ellos se graduara en el corto plazo.

## **INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO**

Esta actividad es coordinada y desarrollada principalmente por la Subdirección Académica de este Centro, participando también la Dirección y la Subdirección de Vinculación y Desarrollo Tecnológico. En la actualidad se tiene 6 líneas de trabajo, eliminando la existencia de proyectos individuales. Las 6 líneas de trabajo mencionadas son las siguientes :

- Procesamiento de Imágenes.
- Robótica, generación de Autómatas en el Campo de la Electrónica.
- Procesamiento Paralelo Como Alternativa Potencial a Máquinas Grandes (MainFrames) y Supercomputadoras.
- Aplicaciones Dedicadas de Microprocesadores en Medición y Control Industrial.
- Computación, electrónica de Automatización y Control de Procesos.
- Aplicaciones de Microprocesadores en Control Proporcional, Integral y Derivativo.

En total, estas 6 líneas de trabajo se pueden traducir en el desarrollo de 30 proyectos de Investigación, tomado en consideración proyectos nuevos registrados en 1994 y 1995 y aquellos proyectos de tipo recurrente en operación durante el periodo reportado. En resumen, en el periodo se tuvieron 22 proyectos nuevos, 7 registros recurrentes y la conclusión de 10 proyectos.

La publicación y difusión de los resultados de la Investigación y Desarrollo Tecnológico constituye un parte muy importante del quehacer del CINTEC. En el período que abarca el informe, el Centro publicó 3 números de su revista (POLIBITS). En estos números aparecen 16 artículos, 3 de ellos con participación de estudiantes del CINTEC, 1 como colaboración externa de estudiantes del ITESM (como parte de su trabajo de Obtención de Grado) y 1 con colaboración conjunta entre profesores del CINTEC y la UAM-IZTAPALAPA.

El Centro asistió en este período a un total de 13 eventos Académicos, presentando un total de 20 trabajos. Del total de eventos, 4 de ellos fueron de carácter Internacional, celebrándose en las ciudades de México, D.F., en la Universidad de las Américas en Cholula, Puebla, en la Ciudad de Dallas, Texas y en Río de Janeiro, Brasil. Cabe hacer notar que en el evento celebrado en Brasil se presentaron 3 trabajos y en el de la Universidad de las Américas 2 .

En las hojas anexas a este reporte se encuentran detalladas todas las actividades aquí reportadas, indicando mas en extenso los datos indicados.

## **EL PROGRAMA DE AUTOEQUIPAMIENTO**

Las actividades desarrolladas dentro de este programa constituyen parte de las responsabilidades de la Subdirección de Vinculación y Desarrollo Tecnológico.

Los equipos de cómputo entregados por parte de este Centro a diversas U.R. del Instituto, en cada caso, se amparan mediante un acta administrativa de entrega-recepción, y suman un total de 27 equipos, cuyas características se indican a continuación:

## Reporte de Actividades

DESCRIPCION	CANTIDAD
IPN e16M (1 unidad de disco removible, 1 disco duro 20MB.)	1(*)
IPN e16M (2 unidad de disco removible, 1 disco duro 20MB.)	2
IPN e32 (2 unidad de disco removible, 1 disco duro 40MB o mayor capacidad)	24
Total de equipos entregados:	27

(\*) Nota: Equipo reasignado.

El valor total del equipo asignado en el período asciende aproximadamente a N\$ 120,000.00, de los cuales 19 equipos se instalaron en este Centro y 8 en diferentes unidades del Instituto.

Actualmente la actividad sustantiva del programa de autoequipamiento se concentra en el mantenimiento del equipo de cómputo asignado desde los inicios del programa hasta la fecha. En el anexo RESUMEN DE MANTENIMIENTO PREVENTIVO, se presenta el resumen de las acciones de mantenimiento preventivo, en el cual se revisaron un total de 406 equipos, con una inversión total de 155 horas/hombre. El anexo MANTENIMIENTO CORRECTIVO, presenta el desglose de las actividades de mantenimiento correctivo que se realizaron, se observa un total de 634 acciones de mantenimiento con un total de 6849 horas/hombre invertidas, de los cuales solo 97 de los servicios corresponden a fallas de los equipos diseñados en el CINTEC que representa un 15.30% del total de servicios, porcentaje que ha disminuido de manera constante conforme aumenta el número de servicios y tiempo de operación; encontrándose el mayor número de fallas en los subsistemas no fabricados por este Centro.

El anexo COSTOS ASOCIADOS presenta el resumen de los costos asociados al mantenimiento de los equipos de cómputo, se hace notar que se consideran todos los factores de operación (salarios, refacciones, combustibles, asesorías, etc.); además, es importante destacar que el servicio se presta sin costo alguno para las U.R's del Instituto.

**Reporte de Actividades**

<b>POBLACIÓN ESCOLAR CINTEC PRIMAVERA '94 - OTOÑO '95</b>						
<b>SEMESTRE</b>	<b>ALUMNOS DE NUEVO INGRESO</b>	<b>ALUMNOS REINSCRITOS</b>	<b>TOTAL DE ALUMNOS INSCRITOS</b>	<b>TEMAS DE TESIS REGISTRADOS EN EL SEMESTRE</b>	<b>EGRESADOS 100% de créditos</b>	<b>TITULADOS</b>
<b>PRIMAVERA '94</b>	13	39	52	3	5	
<b>OTOÑO '94</b>	(Propedéutico) 44	33	77	1		2
<b>PRIMAVERA '95</b>	29	22	51		11	1
<b>OTOÑO 95</b>	(Propedéutico) 42	47	89	5		

**PROFESORES DEL CINTEC REALIZANDO ESTUDIOS DE ACTUALIZACION O POSGRADO**

**NIVEL MAESTRIA EN CIENCIAS**

NOMBRE	INSTITUCION	SITUACION ACTUAL
VALENTE LOPEZ MUÑOZ	CINTEC	ULTIMO SEMESTRE
RUBEN PEREDO VALDERRAMA	CINTEC	20. SEMESTRE

**NIVEL DOCTORADO**

NOMBRE	INSTITUCION	DOCTORADO	SITUACION ACTUAL
ANTULIO MORGADO VALLE	UAM-IZTAPALAPA	DOCTORADO EN CIENCIAS CON ESPECIALIDAD EN COMPUTO PARALELO	INVESTIGACION PRELIMINAR
TEODORO ALVAREZ S.	UAM-IZTAPALAPA	DOCTORADO EN CIENCIAS CON ESPECIALIDAD EN COMPUTO PARALELO	INVESTIGACION PRELIMINAR
MIGUEL A. PARTIDA TAPIA	UAM-IZTAPALAPA	DOCTORADO EN CIENCIAS CON ESPECIALIDAD EN COMPUTO PARALELO	INVESTIGACION PRELIMINAR
ROMEO URBIETA P.	UAM-IZTAPALAPA	DOCTORADO EN CIENCIAS CON ESPECIALIDAD EN COMPUTO PARALELO	INVESTIGACION PRELIMINAR
MIGUEL LINDIG BOS	UNAM	DOCTORADO EN ING. ELECTRICA CON ESPECIALIDAD EN ELECTRONICA	CANDIDATO A DOCTOR
L. ALFONSO VILLA VARGAS	UNIVERSIDAD DE CATALUÑA (ESPAÑA)	DOCTORADO EN ARQUITECTURA Y TECNOLOGIA DE COMPUTADORAS	INVESTIGACION PRELIMINAR

**RELACION DE ARTICULOS PUBLICADOS  
AGOSTO 1994 - SEPTIEMBRE 1995  
(REVISTA POLIBITS)**

**VOL.1 NUM. 13**

1	DESARROLLO DE PROGRAMAS RESIDENTES EN BASE A UN TSR GENERICO	EDUARDO VEGA ALVARADO
2	APLICACIONES DEL PROCESAMIENTO GRAFICO Y DE IMAGENES: LA REALIDAD VIRTUAL (INTRODUCCION)	HECTOR S. GARCIA SALAS
3	TEMPORIZADOR PROGRAMABLE EN BASE A UN DISPOSITIVO MAPL 128	AQUILINO CERVANTES AVILA
4	ANALISIS DISEÑO Y CONSTRUCCION DE UN CONTROL DIGITAL DIRECTO USANDO LA COMPUTADORA IPN E32	ROMEO URBIETA PARRAZALES TEODORO ALVAREZ SANCHEZ MARTIN S. DOMINGUEZ G. (ALUMNO) CLAUDIA LARA VIVAS (ALUMNA) MANUEL SAENZ SANCHEZ (ALUMNO) RAYMUNDO TELLEZ CORTEZ (ALUMNO)
5	EL IMPACTO DE LA NUEVA GENERACION DE MICRO-PROCESADORES EN LA INGENIERIA DE SOFTWARE	MIGUEL A. PARTIDA TAPIA

**VOL. 1 NO. 14**

1	DISEÑO DE CIRCUITOS LOGICOS EN BASE A LA TECNOLOGIA FPGA: UN EJEMPLO DE APLICACION (COMPENDIO)	JUAN CARLOS GONZALEZ ROBLES EDUARDO VEGA ALVARADO
2	SIMULACION DE REDES NEURONALES. MEMORIA BIDIRECCIONAL BIVALENTE ADAPTIVA: BAM	MIGUEL A. PARTIDA TAPIA RUBEN PEREDO VALDERRAMA (ALUMNO) FRANCISCO F. CORDOVA Q. (ALUMNO)
3	PERCEPCION COMPUTACIONAL	PABLO MANRIQUE RAMIREZ RICARDO BARRON FERNANDEZ
4	MULTIPLICADOR DIGITAL DE FRECUENCIA PROGRAMABLE ADAPTIVO DE ALTA PRECISION.	MIGUEL A. PARTIDA TAPIA ADRIANO DE LUCA JOHN GODDARD

## Reporte de Actividades

5	COMPUTACION DE USUARIO FINAL: ¡CAMBIAR O MORIR!	BENITO PIÑEIRO OMAR HUERTA JUAN CARLOS CORRAL (ALUMNOS DEL ITESM)
<b>VOL. 1 NO. 15</b>		
1	COMO DIRECCIONAR MAS ALLA DEL PRIMER MEGABYTE EN EL MODO REAL DE LAS PC	GUSTAVO A. MAS LEVARIO
2	DISEÑO, SIMULACION Y CONSTRUCCION DE UN CONTROL PID APLICADO A UN SISTEMA TERMICO	ROMEO URBIETA PARRAZALES
3	CONTROL REMOTO DE FORMATO COMERCIAL	AQUILINO CERVANTES AVILA
4	SISTEMA DE CONTROL DE LOGICA DIFUSA APLICADO A UN MOTOR DE C.D	ROMEO URBIETA PARRAZALES
5	GENERADOR DE PATRONES DE TELEVISION	JUAN CARLOS GONZALES ROBLES JOSE A. ARIAS GUILAR (ALUMNO) OSVALDO ESPINOSA SOSA (ALUMNO)
6	REPORTE DE ACTIVIDADES REALIZADAS EN EL CINTEC DURANTE EL PERIODO DEL 31 DE AGOSTO DE 1994 AL 1o. DE SEPTIEMBRE DE 1995	

**Reporte de Actividades**

<b>ASISTENCIAS A CONGRESOS, EVENTOS ACADEMICOS, CONFERENCIAS, ETC.</b>		
<b>AGOSTO 1994 - SEPTIEMBRE 1995</b>		
<b>TEMA Y NOMBRE DEL PARTICIPANTE</b>	<b>NOMBRE DEL EVENTO, LUGAR Y FECHA</b>	<b>TIPO</b>
A PARALLEL ARCHITECTURE BASED ON THE SHARED MEMORY PRAM  M. EN I. MIGUEL LINDIG BOS	<b>THE INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING APPLICATIONS AND TECHNOLOGY</b> , DALLAS, TEXAS, 18 AL 21 DE OCTUBRE DE 1994	INTERNACIONAL
SISTEMA DE CONTROL REMOTO DE PARAMETROS  M. EN C. MIGUEL A. PARTIDA TAPIA	<b>XVI CONGRESO NACIONAL DE INGENIERIA ELECTRONICA</b> , CHIHUAHUA, CHIH. 24 AL 28 DE OCTUBRE DE 1994	NACIONAL
UNA ARQUITECTURA PARA PROCESAMIENTO EN PARALELO  M. EN I. MIGUEL LINDIG BOS	<b>XXVII CONGRESO NACIONAL DE LA SOCIEDAD MATEMATICA MEXICANA</b> QUERETARO, QRO. 2 AL 8 DE OCTUBRE DE 1994	NACIONAL
SISTEMA DE CONTROL REMOTO DE PARAMETROS  M. EN C. MIGUEL A. PARTIDA TAPIA	<b>LA COMPUTACION EN EL PROCESO DE GLOBALIZACION ECONOMICA</b> MEXICO, D.F., 9 AL 11 DE NOVIEMBRE DE 1994	NACIONAL
UNA UNIDAD DE CONTROL DE RUTAS PARA COMPUTADORAS CON ARQUITECTURA EN PARALELO  M. EN C. TEODORO ALVAREZ SANCHEZ M. EN C. ANTULIO MORGADO VALLE	<b>IV SEMANA DE LA COMPUTACION</b> PUEBLA, PUE. 14 AL 17 DE NOVIEMBRE DE 1994	NACIONAL
VINCULACION ENTRE EL SECTOR PRODUCTIVO E INSTITUCIONES EDUCATIVAS  ING. HECTOR GARCIA ROJAS	<b>II ENCUENTRO NACIONAL DE VINCULACION</b> CHIHUAHUA, CHIH. 17 Y 18 DE NOVIEMBRE DE 1994	NACIONAL
REALIDAD VIRTUAL Y EDICION DE IMAGENES POR COMPUTADORA  DEA. SAMUEL GARCIA SALAS	<b>ESIME CULHUACAN</b> . MEXICO, D.F., MARZO DE 1995	NACIONAL
ORGANIZADOR Y MODERADOR DE LA SESION: "COMPARACION DEL PENTIUM CON POWER PC"  M. EN I. MIGUEL LINDIG BOS	<b>COMDEX MEXICO '95</b> MEXICO D.F. MARZO DE 1995	INTERNACIONAL
PROCESAMIENTO PARALELO  M. EN C. TEODORO ALVAREZ SANCHEZ	<b>VI SEMANA DEL CONOCIMIENTO</b> MEXICO, DF. MARZO DE 1995	NACIONAL
SOFTWARE PARA EL DESARROLLO DE HARDWARE  M. EN C. EDUARDO RODRIGUEZ ESCOBAR	<b>VI SEMANA DEL CONOCIMIENTO</b> MEXICO, DF. MARZO DE 1995	NACIONAL

## Reporte de Actividades

TEMA Y NOMBRE DEL PARTICIPANTE	NOMBRE DEL EVENTO, LUGAR Y FECHA	TIPO
TECNOLOGIAS EN LOGICA PROGRAMABLE  M. EN C. ANTULIO MORGADO VALLE	<b>VI SEMANA DEL CONOCIMIENTO</b> MEXICO, DF. MARZO DE 1995	NACIONAL
MULTIPLICADOR DIGITAL DE FRECUENCIA PROGRAMABLE ADAPTIVO DE ALTA PRECISION  M. EN C. MIGUEL A. PARTIDA TAPIA	<b>V CONGRESO INTERNACIONAL DE ELECTRONICA, COMUNICACIONES Y COMPUTADORAS</b> CHOLULA, PUE. 24 AL 26 DE ABRIL DE 1995	INTERNACIONAL
DISEÑO Y CONSTRUCCION DE UNA TARJETA PARA DISCOS FLEXIBLES Y DISCOS DUROS INTELIGENTES, PARA UNA PC/AT  M. EN C. ANTULIO MORGADO VALLE	<b>V CONGRESO INTERNACIONAL DE ELECTRONICA, COMUNICACIONES Y COMPUTADORAS</b> CHOLULA, PUE. 24 AL 26 DE ABRIL DE 1995	INTERNACIONAL
LAS POSIBILIDADES DE LA EDUCACION UTILIZANDO METODOS DE MULTIMEDIA Y DE REALIDAD VIRTUAL  DEA. SAMUEL GARCIA SALAS	<b>SEMINARIO NACIONAL DE ACREDITACION Y CERTIFICACION PARA LA COMPETITIVIDAD EN INFORMATICA Y COMPUTACION INCOCI'95</b> GUADALAJARA, JAL, 3 AL 5 DE MAYO DE 1995	NACIONAL
PROGRAMACION ORIENTADA A OBJETOS: SIMULACION DE UNA RED NEURONAL  M. EN C. MIGUEL A. PARTIDA TAPIA ING. FRANCISCO CORDOVA QUIROZ ING. RUBEN PEREDO VALDERRAMA	<b>SEMINARIO NACIONAL DE ACREDITACION Y CERTIFICACION PARA LA COMPETITIVIDAD EN INFORMATICA Y COMPUTACION INCOCI'95</b> GUADALAJARA, JAL, 3 AL 5 DE MAYO DE 1995	NACIONAL
EL IMPACTO DE LA ENSEÑANZA DE LA ING. ELECTRONICA Y LA CIENCIAS COMPUTACIONALES CON EL USO DE LOS FPGA'S  M. EN C. MIGUEL A. PARTIDA TAPIA	<b>SEMINARIO NACIONAL DE ACREDITACION Y CERTIFICACION PARA LA COMPETITIVIDAD EN INFORMATICA Y COMPUTACION INCOCI'95</b> GUADALAJARA, JAL, 3 AL 5 DE MAYO DE 1995	NACIONAL
ASISTENTE  M. EN C. EDUARDO RODRIGUEZ E.	<b>4A. REUNION NACIONAL DE DIRECTORES DE INSTITUCIONES DE EDUCACION EN INFORMATICA</b> QUERETARO, QRO. 15 A 16 DE JUNIO DE 1995	NACIONAL
DESIGN OF A CHIP SET FOR A PARALLEL COMPUTER BASED ON THE CROSSBAR INTERCONNECTION PRINCIPLE  M. EN I. MIGUEL LINDIG BOS	<b>38TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS</b> RIO DE JANEIRO, BRASIL. AGOSTO 13 AL 16 DE 1995	INTERNACIONAL
A FUZZY LOGIC CONTROLER FOR A C.D. MOTOR  M. EN C. ROMEO URBIETA PARRAZALES	<b>38TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS</b> RIO DE JANEIRO, BRASIL. AGOSTO 13 AL 16 DE 1995	INTERNACIONAL
DISEÑO Y CONSTRUCCION DE UN MULTIPLICADOR DIGITAL DE FRCUENCIAS  M. EN C. MIGUEL A. PARTIDA TAPIA	<b>38TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS</b> RIO DE JANEIRO, BRASIL. AGOSTO 13 AL 16 DE 1995	INTERNACIONAL

<b>RELACIÓN DE PROYECTOS EN PROCESO</b>				
<b>PERIODO DE AGOSTO DE 1994 A SEPTIEMBRE DE 1995</b>				
<b>PROYECTOS DE INVESTIGACIÓN 1994</b>				
<b>Nombre del proyecto</b>	<b>No. de registro</b>	<b>Descripción</b>	<b>Responsables y participantes</b>	<b>Estado Actual</b>
<b>PROYECTOS NUEVOS</b>				
Diseño y construcción de una máquina paralela basada en el microprocesador Pentium de Intel	942804	Conclusión del diseño físico de la máquina paralela diseñada durante la primera parte del proyecto, en base al microprocesador Pentium	Miguel Lindig Bos Pablo Manrique Oscar Camacho N. Eduardo Vega A.	En proceso
Diseño y construcción de una tarjeta . reconocedora y procesadora de voz utilizando un microprocesador DSP, geometría algebraica y redes neuronales	942812	El trabajo será enfocado al reconocimiento y análisis de aspectos y frecuencias de la voz, así como comparaciones de amplitudes fonéticas.	Pablo Manrique R Ricardo Barrón F. Teodoro Alvarez S.	Terminado
Diseño y construcción de un controlador lógico programable.	942847	El proyecto plantea el diseño y producción de un sistema de control lógico programable (PLC) que sea capaz de controlar o supervisar un proceso o procesos en forma rápida y precisa.	Eduardo Rodríguez Ignacio Minjarez T. Aquilino Cervantes A	Terminado
Desarrollo de software para una máquina de pruebas mecánicas-dinámicas controlada por computadora.	942855	Pretende el desarrollo del software orientado al usuario de tal forma de servir de interface entre la computadora anfitriona, el sistema de control y la máquina de pruebas mecánicas- dinámicas.	Eduardo Rodríguez Ignacio Minjarez T. Aquilino Cervantes A	Terminado
Sistema de control compensado e implementación del TMS320C25	942863	Consiste en la creación de una tarjeta de circuito impreso integrada por 12 dispositivos adaptados para implementar un control PID de aplicación general.	Teodoro Alvarez S. Pablo Manrique R. Romeo Urbieto P.	Terminado
Diseño y construcción de una tarjeta para evaluación de PLD's	942871	Diseño de una tarjeta para pruebas dinámicas de diseños basados en PLD's ("Testability").	Antulio Morgado V. Ignacio Minjarez T. Juan C. González R.	Terminado

Sistema de Control Remoto de Parámetros.	942995	Diseñar y Construir un sistema digital que controle la excitación de los actuadores existentes en el laboratorio del STC, de tal manera que las señales de vibración que se registran en pruebas de campo sobre diferentes tipos de vehículos se puedan reproducir en el laboratorio.	Miguel A. Partida Tapia Oscar Camacho N. Ma. Elena Aguilar J. Enrique Jarquin L.	Terminado
Diseño, simulación y construcción de un sistema de control adaptable aplicado a un motor de CD.	943002	El trabajo consiste en controlar la variable de velocidad de un motor C.D. usando las tecnologías de microcontroladores recientes aplicado a estas necesidades y técnicas de control modernas.	Romeo Urbietta P. Valente López M.	Terminado
Diseño, simulación y construcción de un sistema de control PID aplicado a un sistema térmico.	942821	El trabajo consiste en controlar la variable temperatura de un recipiente conteniendo agua, usando las tecnologías de microcontroladores recientes aplicado a éstas necesidades y la técnica de control clásico PID.	Romeo Urbietta P. Valente López M. Teodoro Alvarez S.	Terminado
Diseño y construcción de un medidor supervisorio de consumos de energía eléctrica en base al microcontrolador 80C196KC con interface serie para PC y 64 K de RAM.	942839	Muestrear los parámetros de corrientes y voltajes a través de un convertidor analógico digital el cual alimentará estos datos digitalizados a un microcontrolador donde son depurados y modificados para después depositarlos en una memoria de solo lectura (RAM), con el finde efectuar acciones de control y optimar en lo posible los consumos de energía eléctrica.	Oscar Camacho N. Gabriela Tenorio S.	Terminado
<b>PROYECTOS RECURRENTE REGISTRADOS EN 1994</b>				
Diseño y desarrollo de un conjunto de tarjetas de adquisición y control múltiple para prácticas de electricidad y electrónica aplicada para los alumnos del CECyT Miguel Bernard	932094		Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbietta P. Teodoro Alvarez S. Macario García A.	En proceso

Desarrollo y aplicación de software y tarjetas de adquisición de datos para procesos de fundición y laboratorio de arenas.	932078	Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbieto P. Teodoro Alvarez S. Macario García A.	En proceso
Aplicación de los sistemas CAD-CAM	932086	Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbieto P. Teodoro Alvarez S. Macario García A.	En proceso
Elaboración de un manual didáctico de dibujo técnico asistido por computadora para los alumnos de técnico en dibujo industrial por computadora del CECyT Miguel Bernard.	932108	Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbieto P. Teodoro Alvarez S. Macario García A.	En proceso
Diseño y construcción de los mecanismos necesarios para la manipulación de dispositivos de montaje de superficie.	900672	Héctor García Rojas Juan C. González R.	En proceso
Visión mecánica por medio de comparación de patrones.	932132	Miguel Lindig Bos Héctor García Rojas	En proceso
Implementación de un paquete de análisis y procesamiento de imágenes para el microprocesador DSP 56001 de Motorola.	922552	Ricardo Barrón F Pablo Manrique R. Juan Ramírez	En proceso
Implementación de PLD's y ASIC's	932116	Antulio Morgado V. Juan C. González R. Oscar Camacho N.	Terminado

PROYECTOS DE INVESTIGACIÓN 1995				
Nombre del proyecto	No. de	Descripción registro	Responsables y participantes	Estado Actual
<b>PROYECTOS NUEVOS</b>				
Ruteador Virtual	950998	Analizar el problema de la transferencia de datos entre nodos utilizando canales virtuales y algoritmos de enrutamiento que permitan una operación libre de tiempos muertos.	Teodoro Alvarez S. Antulio Morgado V. alumna: Gabriela Tenorio S.	En proceso
Un algoritmo paralelo para enrutamiento en redes permutadas.	951005	Utilizar técnicas actuales de routing permutadas, comparar los algoritmos en tiempo de ejecución para un procesador y n-procesadores.	Teodoro Alvarez S. Antulio Morgado V. Pablo Manrique R.	En proceso
Diseño y construcción de una computadora personal portátil basada en el microprocesador 80386SL de Intel.	951013	Diseño y construcción de una tarjeta madre tipo PC/AT en base al microprocesador 80386SL con funciones periféricas, integradas para ser utilizadas como máquina portátil con base en alimentación por baterías recargables.	Marco A. Ramírez S. Miguel A. Partida T. Juan Julio Cruz Carrasco Tomás H. Ramírez Cueto Enrique Jarquín León alumnos: Ignacio Morales Q. Ma. Aurora Segura.	En proceso
Implementación de control fuzzy aplicado.	950921	Diseño y simulación de algoritmos de control, construcción de los sistemas de control PI y difuso, programación de algoritmos de control en tiempo real.	Romeo Urbieto P. Ma. Elena Aguilar J.	En proceso
Diseño de un algoritmo adaptivo para un multiplicador digital de frecuencia de 16 bits de precisión.	950904	Diseñar un circuito digital que permita servir como plataforma para ser integrado en un solo dispositivo Field Programmable Gate Array (FPGA) cumpliendo la función de un multiplicador digital de frecuencia.	Miguel A. Partida T. Alberto Flores R. Oscar Camacho N. Rubén Peredo V. alumno: Francisco F. Córdoba	En proceso

Diseño y construcción de un simulador para redes neuronales del tipo Bam y Hopfield.	950912	Realizar rutinas en lenguaje orientado a objetos, que definan la estructura básica de los componentes para simular las redes neuro-nales tipo Bam y Hopfield	Miguel A. Partida T. Alberto Flores R. Oscar Camacho N. Rubén Peredo V. Héctor S. García S. alumnos: Francisco F. Córdoba Rafael Noriega O. Jorge E. Piña León F. Romero A.	En proceso
Desarrollo de una interface de control apagado-encendido para un controlador lógico programable diseñado en el CINTEC.	950963	Diseño e implementación de un prototipo de tarjetas para control del tipo apagado-encendido, se espera con esto fortalecer y hacer más versátil el Controlador Lógico Programable diseñado en el CINTEC.	Oscar Camacho N. Ignacio Minjarez T. Marco A. Ramírez S. Fernando Vázquez T. Juan C. González R. alumnos: José Ortega B. Juan Hernández F. Agustín Cruz C.	En proceso
Desarrollo de un sistema monitor y administrador del PLC desarrollado en el CINTEC.	950947	Se espera obtener un software base para el sistema PLC, que servirá de plataforma para la implantación de un sistema operativo propietario del PLC.	Eduardo Rodríguez E. Juan C. González R. Eduardo Vega A. Elizabeth Acosta G.	En proceso
Desarrollo de una interface de control de motores a pasos para un controlador lógico programable.	950955	Obtener un prototipo de apoyo didáctico para ser utilizado en los laboratorios tanto para el desarrollo de otros proyectos como para la impartición de las cátedras de la Maestría en Ingeniería de Cómputo impartida en el Centro.	Ignacio Minjarez T. Eduardo Rodríguez E. Juan C. González R. alumnos: Juan A. Rodriguez Dulce L. Escobar Aureliano Marcos Agustín Cruz	En proceso
Desarrollo de una tarjeta de video de cristal líquido de 640 X 400.	950939	Este proyecto plantea en el diseño e implementación de un prototipo de tarjetas de video para controlar pantallas de cristal líquido del tipo de pantallas de las computadoras portátiles así como de equipos de control industriales.	Ignacio Minjarez T. Eduardo Rodríguez E. Juan C. Cruz C. alumnos: Emilio Chávez A. Rosa Y. Bello Arguimiro Millán	En proceso

Desarrollo de un sistema de comprensión de imágenes usando algoritmos rápidos.	950971	Desarrollar e implementar programas de compactación de imágenes de propósito general que se puedan usar tanto en aplicaciones específicas como para fines didácticos y que tengan como característica principal velocidad y eficiencia.	Ricardo Barrón F. Silvia Toledo M. Adauto Y. Ortiz R. Ma. Elena Aguilar Alumno: Osvaldo Espinosa	En proceso
Implementación y simulación de un sistema de graficación para señales procesadas por un microprocesador DSP TM320C30 y un controlador de sistemas de vídeo TMS340161 de Texas Instruments.	950980	Reconocimiento y análisis de espectros y frecuencias de las señales, así como al procesamiento, control y comparaciones de las amplitudes y anchos de banda de las señales en tiempo real.	Pablo Manrique R Silvia Toledo M. Teresa Lozano H. Enrique Gómez C. alumnos: José A. Arias A. Ignacio R. Rosas	En proceso
<b>PROYECTOS RECURRENTES REGISTRADOS EN 1995</b>				
Diseño y construcción de una máquina paralela basada en el microprocesador Pentium de Intel.	942804		Miguel Lindig Bos Oscar Camacho N. Gustavo A. Mas L. Eduardo Vega A.	Terminado
Aplicación de los sistemas CAD-CAM	932086		Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbieto P. Teodoro Alvarez S. Macario García A.	En proceso
Elaboración de un manual didáctico de dibujo técnico asistido por computadora para los alumnos de técnico en dibujo industrial por computadora del CECyT Miguel Bernard.	932108		Miguel Lindig Bos Miguel A. Partida T. Eduardo Rodríguez E. Romeo Urbieto P. Teodoro Alvarez S. Macario García A.	En proceso

<b>RESUMEN DE ACTIVIDADES DE MANTENIMIENTO PREVENTIVO</b>					
AÑO	MES	TOTAL DE CENTROS	TOTAL DE SERVICIOS	EQUIPOS REVISADOS	HRS/HOMBRE SERVICIO
1994	*AGOSTO			17	6
	*SEPTIEMBRE			55	18
	*OCTUBRE	1	2	39	22
	*NOVIEMBRE	1	2	50	26
	*DICIEMBRE			2	1
1995	*ENERO			42	14
	*FEBRERO			27	10
	*MARZO			32	11
	*ABRIL			12	4
	*MAYO			20	7
	*JUNIO			34	11
	*JULIO			31	10
	*AGOSTO			15	5
	*SEPTIEMBRE			30	10
TOTALES		2	4	406	155
(*)NOTA:EQUIPOS REVISADOS EN LAS INSTALACIONES DEL CINTEC NOTA: ESTA TABLA SOLO ES DE RESPALDO					

## Reporte de Actividades

<b>ANALISIS DEL SERVICIO DE MANTENIMIENTO CORRECTIVO POR TIPO DE FALLA</b>												
<b>1994</b>												
MES	TOTAL SERVICIOS	EQUIPO SERIE IPN				DRIVES	S U B S I S T E M A					
		T	A	B	C		FUENTES	MINITORES	TECLADOS	IMPRESORA	IDE/VGA	OTROS
SEP.	73	26	1	6	19	28	0	5	1	2	5	6
OCT.	62	13	4	5	4	13	0	12	15	2	5	2
NOV.	62	19	0	9	10	21	0	8	8	2	3	1
DIC.	3	1	0	0	1	2	0	0	0	0	0	0
<b>1995</b>												
ENE.	72	25	0	13	12	17	0	8	20	1	0	1
FEB.	56	15	0	6	9	27	1	4	3	3	1	2
MAR.	77	16	2	1	13	21	1	9	20	3	2	5
ABR.	30	12	1	6	5	6	0	4	3	3	2	0
MAY.	42	18	3	8	7	11	0	4	2	3	3	1
JUN.	67	18	1	13	4	28	0	3	4	5	6	3
JUL.	65	29	14	11	4	14	0	7	3	7	2	3
AGO.	25	15	0	6	9	4	0	1	3	0	2	0
<b>TOTALES</b>	<b>634</b>	<b>207</b>	<b>26</b>	<b>84</b>	<b>97</b>	<b>192</b>	<b>2</b>	<b>65</b>	<b>82</b>	<b>31</b>	<b>31</b>	<b>24</b>
<b>PORCENTAJE</b>												
	100%	32.65%	15.30%			30.28%	0.32%	10.25%	12.93%	4.89%	4.89%	3.79%
<p>NOTAS : T= TOTAL DE EQUIPOS DE LA SERIE IPN  A= SERVICIOS DE ADAPTACION O MODIFICACION DE CONFIGURACION  B= SERVICIOS A EQUIPOS IPN QUE NO PRESENTARON FALLAS  C= SERVICIOS A EQUIPOS IPN CON FALLA REAL</p> <p>«LOS SUBSISTEMAS DRIVE, FUENTE, MONITOR TECLADO, TARJETAS IDE, VGA»  NO SON FABRICADOS POR EL CINTEC</p>												

<b>RESUMEN COSTO DE MANTENIMIENTO PERIODO SEP94 - AGOS95</b>				
<b>MANTENIMIENTO PREVENTIVO</b>				
	<b>HORAS</b>	<b>COSTO</b>	<b>SUB TOTAL 1 N\$</b>	<b>SUB TOTAL N\$</b>
SERVICIO	139	1560.4		
TRANSPORTE	10	298.33		
<b>SUBTOTAL</b>	<b>149</b>	<b>1858.73</b>	<b>1858.73</b>	
<b>MANTENIMIENTO CORRECTIVO</b>				
SERVICIO	6849.1	101052		
TRANSPORTE	375.5	17042.35		
<b>SUBTOTAL</b>	<b>7272.6</b>	<b>118094.35</b>	<b>118094.35</b>	<b>N\$ 119953.08</b>
<b>OTROS SERVICIOS</b>				
<b>INSTALACION DE EQUIPO</b>				
SERVICIO	19	384.9		
TRANSPORTE	4	119.33		
<b>SUBTOTAL</b>	<b>23</b>	<b>504.23</b>		<b>N\$504.23</b>
OTROS MATERIALES				N\$14,441.18
MANTO A SUBSISTEMAS				N\$5,901.50
VIATICOS				N\$ 1,330.45
PASAJES				N\$ 1,237.80
COMBUSTIBLES				N\$2,605.85
<b>COSTO TOTAL</b>				<b>N\$ 145974.09</b>

