

39  
2ej

# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



## FACULTAD DE INGENIERIA

**DISEÑO Y CONSTRUCCION DE UN SISTEMA  
DE CAPTURA DE IMAGENES Y CONTROL DE  
ASIGNACION DE EQUIPO**

**T E S I S**  
**PARA OBTENER EL TITULO DE**  
**INGENIERO EN COMPUTACION**

**P R E S E N T A N :**

**DIANA ANABEL HERNANDEZ ALONSO**  
**GABRIELA DE JESUS MAGALLANES GONZALEZ**

**DIRECTOR DE TESIS:**  
**M. EN I. JUAN CARLOS ROA BEIZA**

**México, D.F.**

**1993**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

UNAM



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

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

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

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

# CONTENIDO

## Justificación

### Capítulo I Sistema Digitalizador de imágenes

- I.1 Qué es y para qué sirve un Sistema Digitalizador de Imágenes? 3**
- Terminología
  - El ojo humano
  - Aplicaciones del procesamiento de imágenes
- I.2 Principales componentes de un Sistema Digitalizador de imágenes 19**
- Elementos de un digitalizador
  - Características de un digitalizador de imágenes
  - Componentes de un digitalizador de imágenes
  - Mecanismos para scanear
  - Cámara para captura de imágenes digitales
- I.3 Formatos típicos para la transmisión y recepción de información 37**
- Transmisión digital de imágenes
  - Métodos de modulación
  - Formatos típicos
  - Códigos Binarios
- I.4 Características principales de las señales de video generadas 55**
- Colorimetría

Elementos de imagen  
Exploración Horizontal y vertical  
Parámetros de imagen  
Señal de video compuesta  
Señales de video rojo, verde y azul  
La imagen en el procesamiento digital  
Características del desplegado  
Tecnologías del desplegado

## **Capítulo II** Sistema de captura y procesamiento

### **II.1** Tarjetas digitalizadoras de video 75

Diseño general  
Tarjetas digitalizadoras  
Imagen digitalizada

### **II.2** Software del procesamiento de imágenes 93

Organización del software  
Arquitectura de software  
Productos para el manejo de imágenes

### **II.3** Características y selección del código de barra óptico 101

Introducción  
Patrones del código de barras

### **II.4** Características y selección del sistema lector de código de barras óptico 121

La luz  
El espectro electromagnético  
Clasificación de los scanners

<b>Capítulo III</b>	<b>Diseño y Construcción</b>	
<b>III.1</b>	<b>Características y selección de la tarjeta manejadora de video</b>	<b>137</b>
	Entradas	
	Salidas	
<b>III.2</b>	<b>Diseño y construcción del controlador de equipo</b>	<b>145</b>
	Microcontrolador	
	Bloque analógico de selección	
	Sección de potencia	
	Fuente de poder	
	Programas	
<b>III.3</b>	<b>Desarrollo de utilerías de Software para el manejo, integración e impresión de credenciales con su respectivo código de barras</b>	<b>165</b>
<b>Conclusiones</b>		<b>193</b>
<b>Apéndice</b>		
<b>Apéndice A</b>	<b>Programas Paradox</b>	<b>197</b>
<b>Apéndice B</b>	<b>Programas Controlador</b>	<b>227</b>
<b>Apéndice C</b>	<b>Programas Tarjeta</b>	<b>233</b>
<b>Apéndice D</b>	<b>Programas Temporizadores</b>	<b>277</b>
<b>Apéndice E</b>	<b>Hojas técnicas</b>	<b>281</b>
<b>Bibliografía</b>		<b>289</b>

# **Justificación**

## **JUSTIFICACION DEL TEMA**

Hoy en día la automatización del trabajo se ha convertido en una herramienta cotidiana, que permite agilizar las tareas monótonas, es por eso que surge la idea de armar un equipo que permita generar credenciales universales.

El concepto de credencial universal se puede entender como un sistema que a través de la computadora, se pueda captar la imagen de un alumno, así como sus datos para obtener una credencial con fotografía y código de barras, que pueda servir de identificación oficial dentro de la Facultad y permita que cualquier dependencia que preste un servicio a los alumnos pueda automatizar su tarea de prestador de servicios.

Justamente para ejemplificar la automatización en el préstamo de un servicio se desarrollará un sistema controlador de asignación de equipo en un centro de cómputo. Este sistema implícitamente tiene otras ventajas además de la asignación de equipo como:

1. Estadísticas de uso de equipo, que permita la justificación de adquisiciones futuras, como: hardware, software y material didáctico.
2. Registro constante de las personas que usan el equipo perfectamente identificados para evitar agresiones físicas al mismo.

### *Justificación*

3. Ahorro de energía eléctrica, al apagarse el equipo automáticamente que no este asignado en un momento determinado.

El equipo que se utiliza para desarrollar este proyecto fue adquirido con fines de docencia e investigación de la Facultad, específicamente en la división de Ingeniería Eléctrica, Electrónica y Computación (DIEEC), se aprovechará esta oportunidad para darlo a conocer a través de una aplicación útil con proyecciones de uso en toda la Facultad.

Es importante mencionar que en otros lugares se han desarrollado sistemas que comprenden digitalización de imágenes y expedición de documentos utilizando técnicas más avanzadas de software y hardware, siguiendo el avance de la tecnología que es a pasos agigantados, sin embargo este tipo de proyectos son cada vez más necesarios, pues surgen como la solución de problemas de tipo administrativo como los que comúnmente existen en la Facultad y en muchas otras partes.

# Capítulo

# I

**Sistema Digitalizador de Imágenes**

# **I.1 ¿ QUE ES Y PARA QUE SIRVE UN SISTEMA DIGITALIZADOR DE IMAGENES ?**

El procesamiento digital de imágenes y la manipulación de imágenes por computadora, es relativamente reciente y como otros campos, sufre mitos, desavenencias e información falsa. En este campo convergen diversos conceptos de óptica, electrónica, matemáticas, fotografía y computación; que finalmente resultan confusos.

## **TERMINOLOGIA**

Antes de definir el procesamiento digital de imágenes, se aclararán varios términos que se involucran en este tema. Algunas definiciones fueron tomadas del "diccionario Webster":

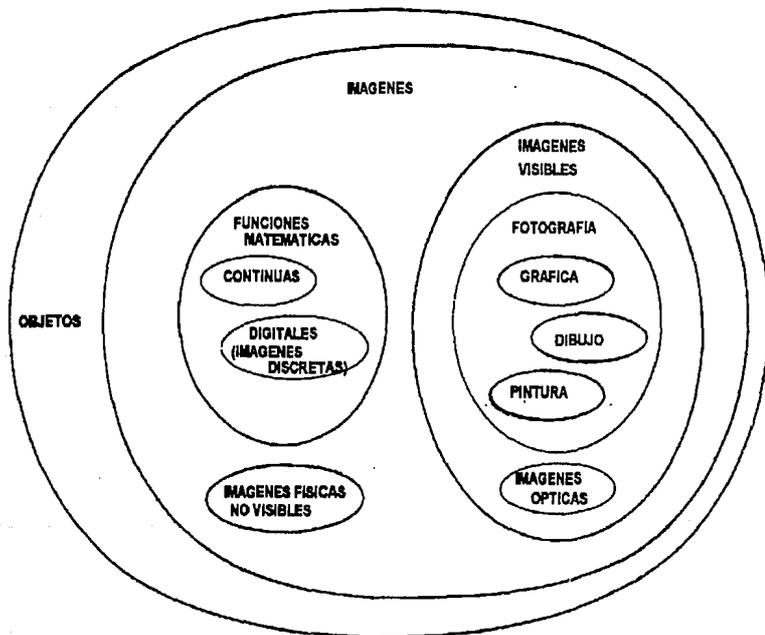
### ***Imagen***

es una "representación o imitación de un objeto, una descripción gráfica" o bien, "una imagen es una representación de algo".

Considerando el conjunto de todos los objetos, las imágenes forman un subconjunto. Existe una correspondencia entre cada imagen en el subconjunto y el objeto que se esta usando para representarlo.

Figura I.1.1.

En el conjunto de imágenes, existe un subconjunto que contiene las imágenes visibles, esto es, que han sido percibidas por el ojo. Dentro de este conjunto existen varios subconjuntos representando los métodos de representación. Este incluye fotografías, dibujos y pinturas. Las imágenes son distribuciones de propiedades físicas. Por ejemplo:



I.1.1 Esquema de tipos de imágenes

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

● **Las imágenes ópticas** son distribuciones espaciales de luz intensa, estas son:

● imágenes físicas visibles

son aquellas que pueden ser vistas por el ojo humano.

● imágenes físicas no visibles

son la temperatura, la presión y los mapas de distribución de densidad de población.

● imágenes físicas multiespectrales

tienen más de una propiedad definida en cada punto.

● **Las imágenes abstractas** son las imágenes referentes a las matemáticas, a las funciones continuas y a las funciones discretas de imágenes digitales.

### ***Fotografía***

es una imagen restringida, es "una representación por una pintura, un dibujo o una gráfica, que proporciona una descripción exacta de un objeto o cosa que permiten formar una imagen mental que dará una idea exacta de las cosas". En el ambiente del procesamiento de imágenes, en todo caso, la palabra fotografía es algunas veces usada como el equivalente de la palabra imagen.

### ***Digital***

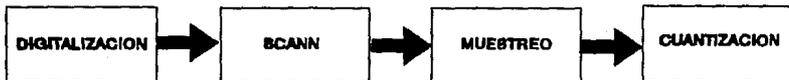
se refiere a los "cálculos por métodos numéricos o por unidades discretas".

### **Imagen digital**

es la "representación numérica de un objeto (la cual puede ser una fotografía)", o bien "es una muestra, una función cuantizada en dos dimensiones la cual es generada por medios ópticos, es una muestra en un espacio rectangular uniformemente cuadrículado, y una cuantización en intervalos iguales de nivel de gris". Una imagen digital es un arreglo rectangular bidimensional de valores cuantizados.

### **Procesamiento**

es el "método de hacer cosas subjetivas para un proceso". Un *proceso* es una "serie de acciones metódicas y ordenadas para obtener resultados". Esta serie de acciones u operaciones son aplicadas sobre un objeto para alterar su forma y representación. Figura 1.1.2.



**1.1.2 Pasos de la digitalización.**

### **Digitalización**

es el "proceso que convierte una imagen analógica en digital". La operación inversa es desplegar, esto es, la generación de una imagen visible a partir de una imagen digital.

### **Scann**

es el "direccionamiento de localidades específicas dentro del dominio de la imagen". Cada una de las direcciones de las subregiones pequeñas en el proceso de scann es llamado "un elemento de fotografía", el cual es abreviado con la palabra **pixel**.

Cuando se digitalizan fotografías, el método scann es el proceso de direccionamiento secuencial de cada punto en la película. El término scann se toma como un equivalente a la palabra "digitalizar". El patrón de cuadrulado rectangular es llamado scanner.

### **Muestreo**

está definido como el "cálculo de los niveles de una imagen en la localización de cada pixel". Cuando digitalizamos imágenes, frecuentemente es deseable emplear dispositivos que conviertan energía luminosa en energía eléctrica. Los dispositivos de este tipo son llamados transductores y el proceso se llama transducción.

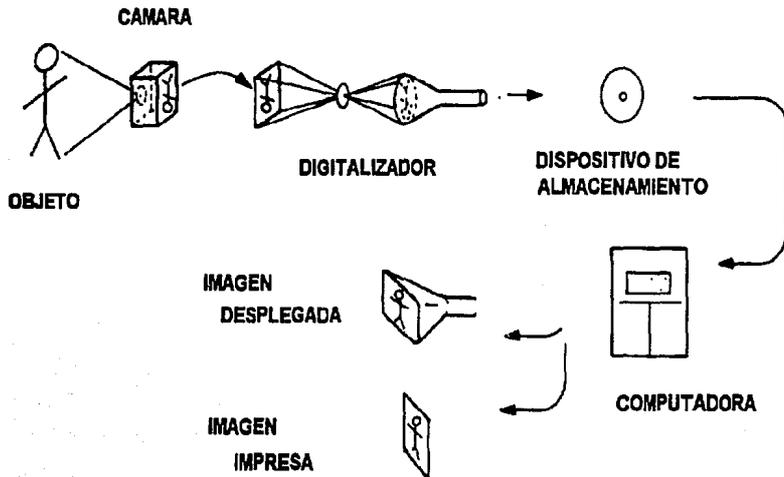
### **Cuantización**

es la representación de un valor por un entero. En los procesos numéricos con computadoras, es necesario reducir los valores de las mediciones continuas de unidades discretas a su representación por números.

Finalmente el **procesamiento digital de imágenes** es una "representación numérica subjetiva de los objetos por una serie de operaciones". En el caso de las fotografías, el procesamiento cambia su forma para tomar una más atractiva.

Los pasos de la digitalización: scanning, muestreo, transducción (si

es necesario), y cuantización son suficientes para obtener la representación numérica de una imagen. El proceso inverso nos permitirá desplegar la imagen. Figura 1.1.3.



### 1.1.3 Procesamiento digital de imágenes

Las operaciones que se pueden realizar con imágenes digitales se clasifican en:

- Una operación global,

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

es aquella que es aplicada igualmente en toda la imagen digital.

- Una operación puntual

el valor del pixel de salida depende sólo del valor de su correspondiente pixel de entrada. Las operaciones puntuales son algunas veces llamadas manipulación de contraste.

Otros Conceptos utilizados son:

- Una operación local

es cuando el valor del pixel de salida depende de los valores de los pixeles adyacentes del correspondiente punto de entrada.

- La notación de contraste

se refiere a la variación en las escalas de grises en una imagen.

- El ruido

es como un contaminador aditivo (o multiplicativo) de una imagen.

- La densidad de la muestra

de una imagen digital es el número de niveles de grises por unidad de medida en el dominio.

- La resolución en la escala de grises

es el número de niveles de grises por unidad de medida de

amplitud de la imagen.

- La amplificación

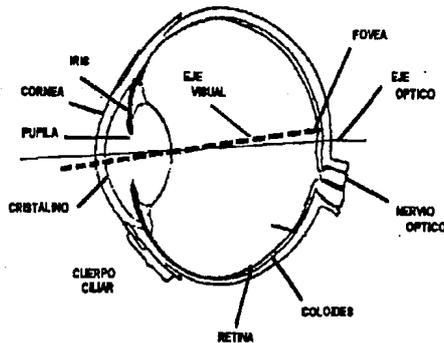
se refiere a la relación del tamaño entre una imagen y el objeto que la imagen representa. La amplificación es una relación completa entre la entrada y la salida digital de las imágenes.

## EL OJO HUMANO

Anatómicamente el ojo tiene la forma de una canica grande, Figura. I.1.4, mide 24 mm y se encuentra dividido en dos espacios virtuales en el segmento anterior se encuentran las estructuras ópticas con valor dióptrico (una *dioptría* es la capacidad que tiene un lente en hacer converger los rayos de luz a un metro de distancia).

La primera estructura es la *córnea*, es una estructura transparente, situada en la parte anterior del ojo, envía los rayos de luz convergente hacia el cristalino; sigue un espacio virtual denominado *cámara anterior* por situarse adelante del *iris* (esta estructura hace las veces de diafragma de una cámara fotográfica, el iris nos da el color del ojo).

En la cámara anterior se encuentra un líquido llamado *humor acuoso* que da la presión intraocular.



I.1.4 Esquema del ojo humano

Por la parte posterior del iris se encuentra la **cámara posterior** que mide 1 mm aproximadamente ahí también se encuentran los **procesos ciliares** que son en donde se produce el humor acuoso.

La siguiente estructura es una lente biconvexa en el centro y se llama **crystalino** (cuando se opaca se forma una catarata) su fuerza es aproximadamente de 20 dioptrías y hace converger los rayos de luz en la **retina**, a través de una estructura transparente, viscosa y semejante a una gelatina que se denomina **humor vítreo**.

La luz del objeto que se observa incide en la retina y ahí se forma la imagen. La visión de patrones se lleva a cabo mediante la distribución discreta de receptores de luz sobre la superficie de la retina.

La retina es la capa más interna del ojo, está constituida por tejido nervioso; tiene 9 capas de células dentro de las cuales las más importantes por su función son 3 tipos de neuronas: los **fotoreceptores** que son células especializadas y toman 2 tipos de aspectos: los conos y los bastones.

Los **conos** están localizados principalmente en la porción central de la retina denominada **fóvea** y son muy sensitivos al color. El ojo es capaz de observar detalles muy finos puesto que cada cono está conectado a una terminal nerviosa. Los músculos del ojo controlan su movimiento para que el objeto de interés caiga en la **fóvea**. La visión mediante los conos se denomina fotópica.

El número de **bastones** es mucho mayor que el número de conos. La amplia distribución y el hecho de que muchos bastones están conectados

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

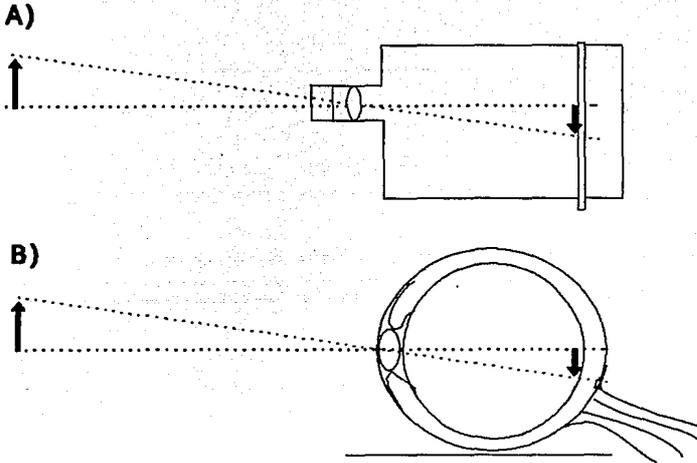
a una sola terminal nerviosa reduce la resolución de los detalles discernibles por estos receptores. Los bastones dan un campo general de visión, no se relacionan con la visión de los colores y son muy sensitivos a los niveles de iluminación. A este tipo de visión se le denomina escotópica.

La retina tiene una superficie aproximada de 5 cm, existen unos 130 millones de bastones y 7 millones de conos aproximadamente, hay una estructura funcionalmente muy importante en la retina que se denomina **mácula**, corresponde al punto de visión más nítido, está situada hacia el lado temporal del ojo,

En el centro de la mácula existe una zona de depresión todavía más fina y de mayor concentración de células fotorreceptoras que se denomina **foveola** y mide 400 micras, en la **fóvea** la concentración de conos es de aproximadamente 150,000 conos por  $\text{mm}^3$  y la agudeza visual es de 20/20. En la fóvea no existen bastones.

Estas células fotorreceptoras se conectan con 2 tipos más de células nerviosas, estas se denominan **células bipolares** y otras las **ganglionares** estas células al unirse forman el nervio óptico que lleva la información al cerebro en la corteza cerebral y en la región occipital donde se hace receptible la imagen que captamos.

Para entender el funcionamiento del ojo adecuadamente compararemos el ojo a una cámara fotográfica, Figura. I.1.5, así la córnea se asemeja al lente anterior de la cámara fotográfica, el diafragma de la cámara fotográfica corresponde al iris que al cerrarse y abrirse permite la entrada de luz al ojo y la película corresponde a la retina.



### 1.1.5 Formación de la imagen en la cámara fotográfica y en el ojo.

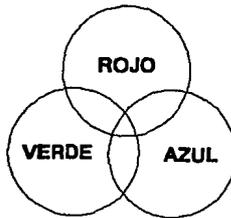
Existe algo importante que los ojos presentan y es el fenómeno de acomodación similar al enfoque de una cámara fotográfica, esto está determinado por el cristalino y el cuerpo ciliar, cuando se tensa esta estructura se puede enfocar y ver de cerca, y cuando se ve a lo lejos desaparece esta tensión.

Lo que si es importante recalcar es que los fotorreceptores al color son únicamente 3 en la retina, en otras palabras existen 3 tipos de conos los rojos, los azules y los verdes.

La visión se produce en el cerebro al interpretar la información que recoge el ojo a través de la retina. Las células de la retina, conos y

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

bastones se estimulan por la luz que penetra al ojo. Los conos se encargan de ver el color de acuerdo a la teoría de mayor aceptación, tienen tres pigmentos: **ROJO**, **VERDE** y **AZUL**, Figura I.1.6. Con la mezcla de estos tres pigmentos se generan todos los colores que el ojo puede percibir.



**I.1.6 Colores primarios para el ojo.**

La mayoría de las computadoras utilizan un monitor tipo televisión, estos monitores tienen un **RTC** (tubo de rayos catódicos), para generar la imagen emiten fotones intermitentes dentro del espectro electromagnético, estos fotones llegan a la retina estimulando los conos y bastones.

Es interesante considerar el número de diferentes fotografías que existen en la vida real. Consideraremos como "fotografías diferentes" las que son percibidas por el ojo humano y que son diferentes para otro.

El ojo humano puede distinguir aproximadamente **40** distintas

tonalidades de grises. El análisis precedente asume que el ojo humano es el último consumidor del "procesamiento digital de imágenes". Si consideramos una fotografía digital de 8 x 10 pulgadas para 1000 líneas de 1000 muestras cada una, esto es, un millón de píxeles, los píxeles son por ellos mismos indistinguibles.

Esto significa que el muestreo es bastante bueno para representar la imagen exactamente como el ojo humano la tiene concebida. Una fotografía de 8 x 10 pulgadas por lo tanto puede ser completamente representada en forma única por una imagen de millón de puntos teniendo 40 niveles de grises.

Así, la importancia del ojo humano, se hace presente, cuando se trata el tema de "procesamiento digital de imágenes", ya que su funcionamiento es la base para el entendimiento de este tópico.

## **APLICACIONES DEL PROCESAMIENTO DE IMAGENES**

Son muchas las aplicaciones que surgen a partir de la facilidad de almacenar y manipular las imágenes digitalizadas, es por ello que actualmente se invierten muchos recursos humanos y tecnológicos para la investigación de nuevas técnicas. Algunas aplicaciones se mencionan a continuación:

- Las **cámaras digitalizadoras** producen y transmiten fotografías rápidamente. El servicio de noticiarios usa cámaras digitalizadoras para tomar fotos y transmitir las vía modem.

Con una cámara digital, la velocidad de distribución de fotos es cercana a la velocidad de la televisión en vivo, pero la calidad es mucho mejor que la de las fotos transmitidas por cable.

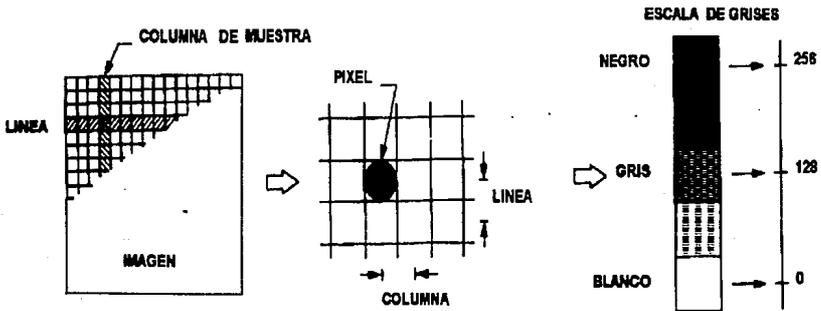
- Probablemente la ventaja más importante del uso de las imágenes digitales es en **documentos publicitarios**. Se insertan fotografías de ejecutivos en las cartas, diagramas en manuales técnicos, productos fotográficos en manuales de vendedores, y documentación de fotografías en presentaciones.

- Las imágenes digitales son utilizadas en el desarrollo de **librerías de imágenes**. Se usan discos duros, discos ópticos y otros medios de almacenamientos para guardar imágenes de alta calidad. Las imágenes digitales son usadas en el ambiente de la fotografía convencional en la que se elimina la necesidad de procesos químicos.

**El factor que determina el futuro del procesamiento digital de imágenes, es la reducción de costos del equipo. Un segundo factor es el incremento de disponibilidad de equipo para digitalización de imágenes y despliegue, a través de microprocesadores a bajo costo, y el uso de sistemas que contengan cámaras que permitan digitalizar, procesar y almacenar para posteriormente desplegar y editar la información capturada de una forma rápida y eficiente.**

## I.2 PRINCIPALES COMPONENTES DE UN SISTEMA DIGITALIZADOR DE IMAGENES

El procesamiento digital de imágenes básicamente requiere de una computadora. En suma, el sistema debe tener dos elementos de entrada/salida, un digitalizador de imágenes y un dispositivo de despliegue. Este proceso de convertir la imagen en datos numéricos es llamado digitalización. La imagen se divide en pequeñas regiones llamadas elementos de imagen o *pixeles*.



### I.2.1 Elementos de imagen (pixeles).

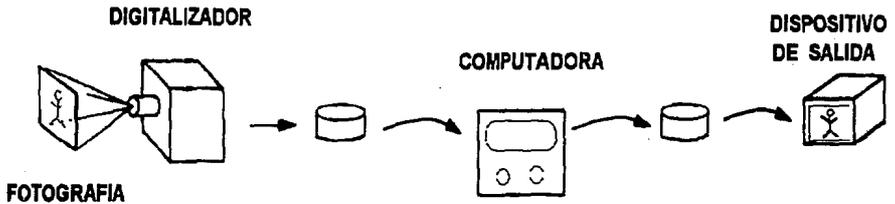
El esquema de subdivisión común es la muestra rectangular en rejilla. La imagen es dividida en líneas horizontales formadas de pixeles

adyacentes. Figura 1.2.1

En cada pixel, la intensidad de la imagen es muestreada y cuantizada. Este paso genera un entero en cada pixel que representa la brillantez de la imagen en un punto. Dados los valores de todos los pixeles, la imagen es representada por una matriz de enteros.

Cada pixel tiene una localización o dirección (número de línea o renglón y número de columna) y un valor entero llamado el *nivel de gris* o escala de grises.

La Figura 1.2.2, muestra un sistema para el procesamiento de imágenes. La imagen digital producida por el digitalizador es la imagen de entrada a la computadora, la cual es leída línea por línea.



1.2.2 Sistema de Procesamiento Digital de Imágenes.

La computadora genera la imagen de salida pixel por pixel, que es simultáneamente escrita en un dispositivo de almacenamiento de datos línea por línea. El producto final es desplegado por un proceso que es contrario a la digitalización.

La escala de grises de cada pixel es usada para determinar la brillantez del punto correspondiente en el despliegue de la pantalla.

## **ELEMENTOS DE UN DIGITALIZADOR**

Un digitalizador de imágenes debe ser capaz de: dividir una imagen en elementos-imagen y en direcciones individuales, medir el nivel de grises de la imagen de cada pixel, cuantizar las mediciones para producir un entero, y escribir este conjunto de enteros en un dispositivo de almacenamiento.

Por lo anterior un digitalizador se compone de 5 elementos:

- **Ventana de muestreo.**

Permite el acceso a uno o un conjunto específico de pixeles de la imagen.

- **Mecanismo de rastreo para la imagen (scan).**

Permite el movimiento de la ventana de muestreo sobre la imagen siguiendo un patrón determinado, para asignar direcciones a cada uno de los pixeles.

- **Sensor.**

Mide el grado de luminosidad y brillantez de la imagen en cada uno de los pixeles a través de la ventana de muestreo. El sensor es comúnmente un transductor que convierte la intensidad de la luz en voltaje o corriente eléctrica.

- **Cuantizador.**

Convierte la salida continua de sensor en un valor discreto (entero). El cuantizador es un convertidor Analógico-Digital, que produce un número que es proporcional a la entrada de voltaje o corriente.

- **Medio de salida.**

Los valores del nivel de grises producidos por el cuantizador deben ser almacenados en un formato adecuado para subsecuentes procesamientos. El medio de salida puede ser cinta magnética, discos magnéticos, etc.

## **CARACTERITICAS DEL DIGITALIZADOR DE IMAGENES.**

Los digitalizadores difieren en los elementos que los componen para ejecutar sus funciones, pero deben ser equivalentes en cuanto a sus funciones básicas. Estas son:

- El tamaño de la ventana de muestreo
- El espacio entre los pixeles adyacentes
- El tamaño de la imagen que puede manejar el digitalizador
- El parámetro físico que se mide y cuantifica. (la brillantes)
- La linealidad del digitalizador
- El número de niveles de grises
- El nivel de ruido

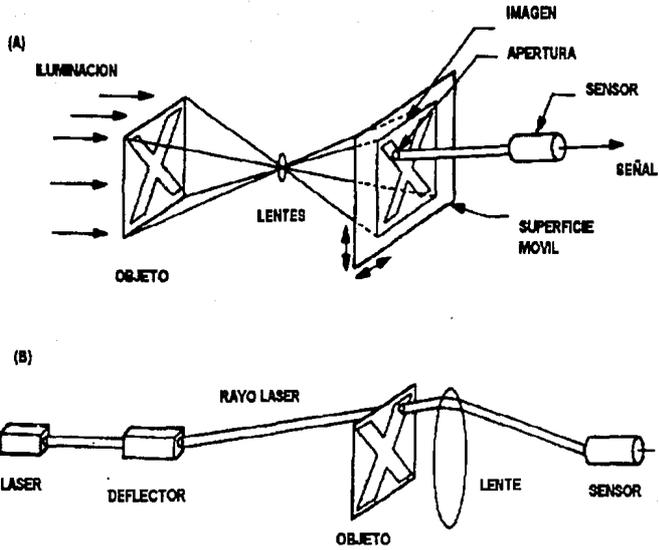
Existen dos técnicas importantes de digitalización:

### ***Scan-out.***

En el sistema "scan-out", el objeto o imagen de video son iluminados continuamente, y la apertura de muestreo permite al sensor "ver" un pixel a la vez. Figura 1.2.3.

### ***Scan-in.***

En el sistema "scan-in", sólo un pequeño punto del objeto es iluminado y la luz transmitida es captada por el sensor. En este caso, el objeto es rastreado con un rayo de luz. Figura 1.2.3.



**1.2.3 Técnicas de Digitalización:**  
**a) Digitalizador Scan-Out, b) Digitalizador Scan-In**

## **COMPONENTES DE UN DIGITALIZADOR DE IMAGENES.**

Como se explicó anteriormente, un digitalizador debe tener una fuente de luz, un sensor de luz y un sistema de rastreo (scan). A continuación se explicarán más detalladamente estos componentes:

- **Fuente de luz.**

Tiene como función iluminar al objeto que será digitalizado con el fin de proveer un medio de transmisión de la imagen hacia el sensor. Para sistemas "scan-out" se utiliza la **luz incandecente**.

Para sistemas "scan-in" se utiliza un foco pequeño, que es manejado con lentes para obtener un punto de luz. El **rayo láser** podría ser utilizado para la iluminación general en un sistema "scan-out", pero su principal ventaja radica en producir un punto pequeño de alta intensidad para el sistema "scan-in". Figura 1.2.3. Recientemente los **LEDs** (light-emitting diodes) forman fuentes de luz compactas y son utilizados en sistemas "scan-in".

- **Sensores de luz.**

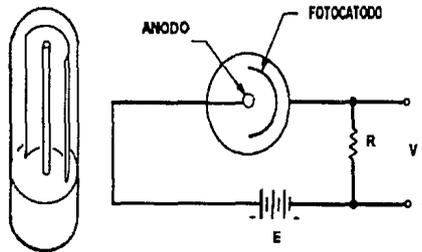
Los sensores de luz producen una señal eléctrica proporcional a la intensidad de la luz. Existen varios tipos:

- Dispositivos Fotoemisores
- Dispositivos Fotoconductores
- CCD (Dispositivos acoplados por carga)

a) Dispositivos fotoemisores. (Fototubos)

Las sustancias fotoemisoras emiten electrones cuando la luz incide sobre éstas. Los **fototubos** tienen un ánodo cargado positivamente y un cátodo cargado negativamente.

Cuando los fotones están suficientemente energizados, golpean el fotocátodo, y los electrones son liberados de la superficie, estos emigran al ánodo, produciendo un flujo de corriente. Esta corriente es proporcional al flujo de fotones en el fotocátodo. La corriente es enviada por un circuito externo, así es muestreada y cuantizada. Figura 1.2.4.

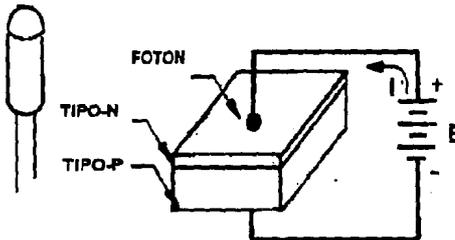


1.2.4 Esquema de un fototubo

b) Dispositivos fotoconductores. (fotodiodos y fototransistores)

Los **fotoconductores**, como sulfato de cadmio y sulfato de selenio, disminuyen la resistencia eléctrica cuando se exponen a la luz. Los fotodiodos y fototransistores cambian sus características bajo la influencia de la luz.

El **fotodiodo** es un dispositivo de estado sólido de unión tipo P-N que puede ser expuesto a la luz. En operación, la unión muestra una corriente inversa y una alta impedancia. Una capa del dispositivo (por ejemplo la capa-N) es delgada así que la luz puede penetrar la unión. Figura 1.2.5.



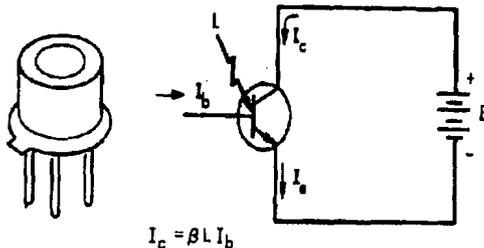
1.2.5 Esquema de un fotodiodo

El número de electrones libres es proporcional a la intensidad de luz incidente. La aplicación de la luz en la unión da como resultado una transferencia de energía en forma de fotones, lo que origina un número elevado de portadores minoritarios y un mayor nivel de corriente inversa. La corriente es proporcional al flujo de fotones.

El **fototransistor** está montado en un plástico transparente o tiene un lente que permite el acceso de la luz a la unión. El movimiento de estos portadores constituye la corriente de la base en el transistor. La corriente del colector es proporcional a la corriente de la base pero es amplificada por el factor de amplificación del transistor. Figura 1.2.6.

Externamente, el comportamiento del fototransistor es como el del

fotodiodo excepto por su alta sensibilidad. Ambos el fotodiodo y el fototransistor tienen una respuesta rápida y estable a la variación de la intensidad de la luz, por lo anterior son excelentes sensores para la digitalización de imágenes.



1.2.6 Esquema del fototransistor

**c) Dispositivos acoplados por cargas (CCD)**

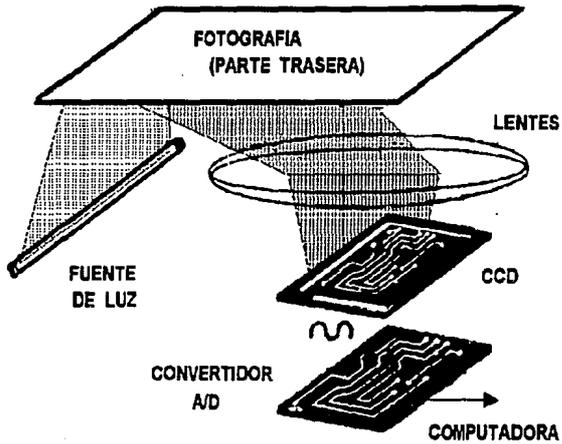
El CCD es un circuito integrado que contiene una hilera de sensores luminosos fuertemente empaquetados; el voltaje de salida de cada sensor luminoso es proporcional a la cantidad de luz que incide sobre él.

El documento que va a ser leído queda iluminado por una fuente de luz fluorescente o incandescente. La luz reflejada se ve a través de unos lentes y ésta es reflejada hacia el CCD. Figura 1.2.7.

Una sola línea leída del documento se enfoca hacia el CCD y el voltaje de salida de cada sensor luminoso se convierte a un valor digital

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

(un patrón binario de bits) mediante un convertidor Analógico-Digital.



### 1.2.7 Dispositivo acoplado por carga

## MECANISMOS PARA SCANNEAR

A continuación se discutirá las técnicas que serán utilizadas para mover el scanner o el punto de iluminación sobre de la imagen. Se consideran la fuente de luz, el sensor, y el mecanismo de scan como un sistema completo de digitalización.

- **Scanner Moving-Mirror**

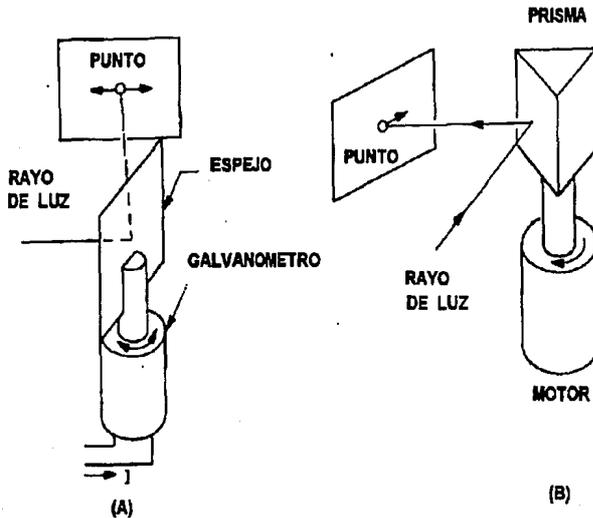
En los sistemas "scan-in", el rayo de iluminación puede ser reflejado por un espejo montado en un motor galvanómetro. La corriente a través del motor rota el espejo reflejando el rayo de iluminación.

La mayor desventaja de la oscilación de un espejo es la baja velocidad de operación. El volumen del espejo y la aceleración son determinantes para la velocidad a la cual pueden operar. Figura 1.2.8.

Los scanners Moving-Mirror están expuestos a distorsiones geométricas de la imagen scaneada. El desplazamiento lineal del punto a scannear es proporcional no al ángulo de deflexión del espejo pero sí a la tangente de ese ángulo, Figura 1.2.8.

Esto produce un espaciamiento de pixeles no uniforme, resultando una especie de alfilerero distorsionado cuando la imagen es desplegada.

Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.

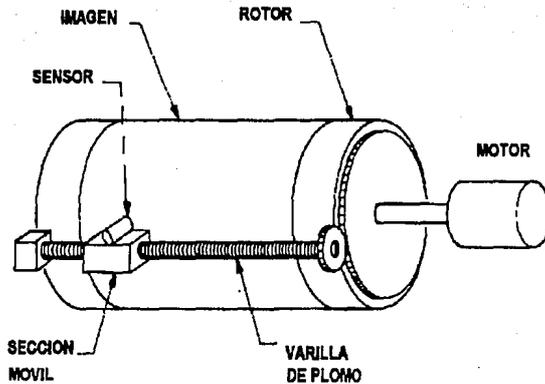


### 1.2.8 Scanner Moving-mirror

- **Scanner mecánico**

La Figura 1.2.9, muestra dos métodos mecánicos para rastrear (scan) imágenes, la rotación de un tambor y una varilla de plomo.

Una imagen fotografiada es expuesta parcial o totalmente, alrededor del tambor cilíndrico, el tambor se rota para obtener la imagen pasada a través de una apertura estática. Este efecto permite rastrear (scan) en una sola dirección. La apertura puede ser montada en una varilla de plomo que rota para mover la apertura a través de la imagen.

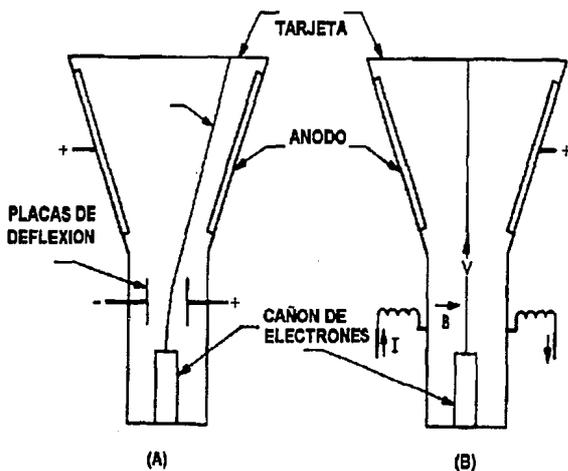


**1.2.9 Scanner Mecánico**

En la Figura 1.2.9, la rotación del tambor y la varilla de plomo han sido combinados para producir un scanner de imágenes de dos/dimensiones. Los mecanismos de Scanner como estos están limitados en su velocidad de operación pero provee una buena estabilidad geométrica y un costo relativamente bajo.

- **Scanner de rayo de electrones**

Existen algunos dispositivos electrónicos útiles en la digitalización y despliegue de imágenes con un rayo de electrones. La siguiente figura (Figura 1.2.10) ilustra dos métodos para desviar un rayo de electrones para rastrear (scan) una tarjeta.



### 1.2.10 Scanner de Rayo de Electrones

#### a) Deflexión electrostática, b) Deflexión electromagnética

- Un **rayo de electrones**,

generado por un cañón de electrones en la base del tubo, es atraído hacia la tarjeta por el ánodo cargado positivamente. Como el rayo de electrones pasa entre las placas electrostáticas para desviarlo, el campo eléctrico ejerce una fuerza en los electrones, cambiando la dirección de su trayectoria. Figura 1.2.10.a.

- Un **campo magnético transversal**,

puede ser usado para desviar un rayo de electrones. La fuerza

sobre las partículas cargadas en el campo magnético es el vector producto de la velocidad de las partículas y del campo magnético. Figura 1.2.10.b.

## **CAMARA PARA CAPTURAR IMAGENES DIGITALES**

Existe dos tipos de cámaras que usan CCD's (charge-coupled devices, dispositivos de carga acoplados) y son:

- still-video (video continuo)
- true digital (digitalización verdadera).

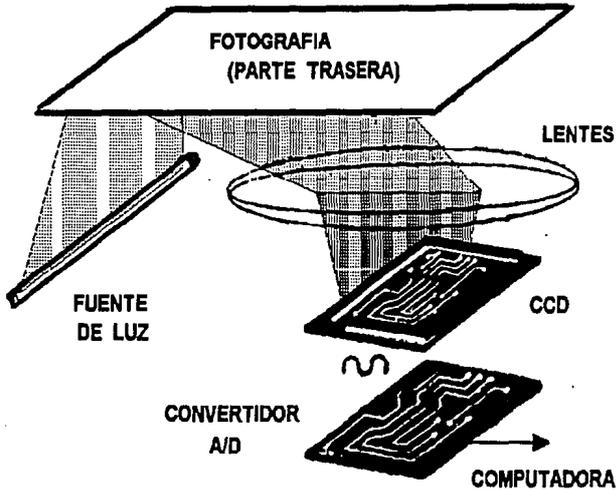
Los lentes de una cámara digital enfocan la imagen en un CCD instalado. Los CDDs captan las imágenes sobre un conjunto de semiconductores colocados como una rejilla, los cuales detectan la luz. Figura 1.2.11.

El CCD es la entrada para una computadora que consiste de un microprocesador, memoria, circuitos de multiplexaje, y fuente de poder.

Las imágenes son almacenadas en la memoria de la cámara o dentro de un dispositivo de almacenamiento permanente como un disco flexible o disco duro.

Las imágenes se almacenan en la memoria de la cámara, y se usan diferentes métodos para transferirlas a la computadora, usando el puerto serial o paralelo de la computadora.

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*



**1.2.11 Esquema de una cámaras digital**

- **Cámara Still-video**

Las primeras cámaras comerciales que usaron CCDs fueron las cámaras "still-video". Permite observar imágenes de la cámara de video en monitores o bien imprimirlas.

Las cámara "still-video" están limitadas a la resolución de la televisión la cual es menos de 500 líneas de rastreo. Una imagen "still-video" es similar en calidad a una fotografía de periódico.

- **Cámara True-Digital**

Las cámaras "true-digital" son combinaciones híbridas de una cámara óptica con un microprocesador de una computadora. La resolución de las cámaras digitales está limitada sólo por la sensibilidad del CCD a la luz.

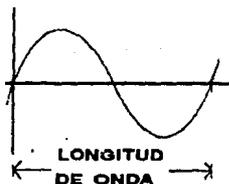
Usando una cámara digital, se tienen las ventajas de portabilidad y resultados inmediatos, esto es, se observa exactamente como la imagen será a la salida y para las aplicaciones de cámaras digitales se tienen las técnicas de post-procesamiento digital.

## I.3 FORMATOS TIPICOS PARA LA RECEPCION Y TRANSMISION DE INFORMACION.

### TRANSMISION DIGITAL DE DATOS

En este punto se describirán los principales métodos de transmisión y recepción de información. El propósito de un sistema de comunicación es la transmisión de información desde un punto en el espacio a otro.

Desde el punto de vista de los sistemas, una señal de entrada y su respuesta se emplean para caracterizar el comportamiento de un sistema dado. Una *señal* se define como una función univaluada en el tiempo. Figura I.3.1.



I.3.1 Representación de una señal

Las señales pueden clasificarse en categorías básicas por su energía

o potencia finita. Otras categorías útiles comprenden las señales periódicas o aperiódicas y las señales aleatorias o determinísticas. Para la transmisión digital de datos, dos voltajes diferentes son empleados para la representación binaria del 0 y 1.

El intervalo en esta transmisión,  $T$ , es conocido como el intervalo de la señal o el tiempo del bit y esta es su velocidad de transmisión,  $1/T$ , se refiere a cuantos bits por segundo *bps*.

La rapidez de la transmisión, es la frecuencia ó ancho de banda necesaria en los medios de transmisión y equipos de comunicaciones. Por esta razón, la velocidad de transmisión es el valor mas significativo en los componentes físicos de comunicaciones. Los cuales son diseñados para ofrecer una velocidad de transmisión mas o menos de 1000bps o 1 kbps.

Para los sistemas de comunicación digitales y específicamente binarios, se envía información caracterizada por una muestra de entre  $n$  posibles entradas. Para enviar esta muestra, se genera una palabra digital compuesta por  $m$  símbolos binarios, cada palabra binaria consta de  $m = \log_2 n$  dígitos binarios para presentar una muestra de entre  $n = 2^m$  posibilidades. Ejemplo:

n	1	2	3	....	254	255	256
b0	0	1	0	....	1	0	1
b1	0	0	1	....	0	1	1
b2	0	0	0	....	1	1	1
b3	0	0	0	....	1	1	1
b4	0	0	0	....	1	1	1
b5	0	0	0	....	1	1	1
b6	0	0	0	....	1	1	1
b7	0	0	0	....	1	1	1

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

Cuando se usa de esta manera,  $m$  se llama número de **bits** (binary digits, dígitos binarios) necesarios para representar a uno entre  $n$  posibles estados de entrada. Así, se necesita una palabra de ocho bits para describir uno de los 256 posibles estados de entrada.

La rapidez de la señal transmitida es:

$$R = mr \text{ bits por segundo (bits)}$$

$m$  número de bits de la palabra binaria

$r$  símbolos por segundo

En el receptor la señal transmitida se adultera por la adición de ruido  $y$ , como resultado, el receptor cometerá algunos errores. Para lo anterior, se debe estudiar la **capacidad del canal** ( $C$ ) que está dada por la ley de Hartley-Shannon:

$$C = B \log_2(1 + S/N) \text{ bps}$$

donde:

$B$  - es el ancho de banda del canal (en Hz)

$N$  - potencia media del ruido

$S$  - potencia media de la señal transmitida

$N/S$  - relación señal a ruido (signal-to-noise ratio)

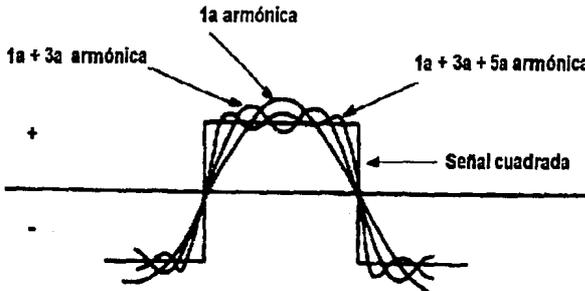
Para tener un resultado aceptable se considera que  $R \leq C$ , esto reducirá la presencia de posibles errores.

El término **baud** es una medida de la velocidad de transmisión de símbolos y dependerá de la codificación digital empleada, por lo tanto es

redundante hablar de **baud-rate** ya que por sí sólo, baud, es la medida de la velocidad de transmisión (  $1/T$  símbolos por segundo, donde T es el intervalo de T segundos).

Para una señal multi-nivel teniendo mas de dos niveles, los bauds y bps serán diferentes. Por ejemplo, en un sistema de 300 bauds, los símbolos se enviarán a razón de 300 por segundo.

Para determinar la frecuencia actual ó ancho de banda ocupada por un tren de pulsos rectangulares (secuencia de niveles binarios 0 y 1), necesitamos el **Análisis de Fourier** Esto será presentado por una serie de armónicas (teóricamente infinito) que son necesarias para representar la forma de onda en el dominio de la frecuencia.



### I.3.2 Síntesis de armónicas

Un camino muy usual de referirse a los requerimientos del ancho de banda de la frecuencia con los bps, para nuestros propósitos es llevar a

cabo un proceso de síntesis de armónicas limitando el número de armónicas. En la Figura I.3.2, se ilustra gráficamente.

Comenzando con un término,  $\cos(2\pi t/T)$ , como la componente fundamental, una tercera armónica,  $-1/3\cos(2\pi 3t/T)$  es sumada. Esto tiene el efecto de aplanar la curva fundamental como se muestra, introduciendo unas ondas que tienden a sintetizar el pulso. La adición de una cuarta armónica de los N-términos de la serie:

$$x(t) = \cos(2\pi t/T) - 1/3\cos(2\pi 3t/T) + 1/5\cos(2\pi 5t/T) - 1/7\cos(2\pi 7t/T) + \dots + 1/N\cos(2\pi Nt/T)$$

produce una sumatoria la cual es utilizada para formar un pulso rectangular. Una síntesis es razonable al considerar arriba de la quinta componente, así un bps será aproximadamente equivalente a 5Hz.

La información digital es más usualmente transmitida como un grupos de dígitos que como una secuencia de 0's y 1's. Algunos términos usados son los siguientes:

**Byte:** grupo de 8 bits binarios frecuentemente representan un carácter alfanumérico particular.

**Frame:** un grupo de bits (frecuentemente un grupo grande) es un ciclo de eventos como una secuencia de grupos en el tiempo, cada uno representa una señal digital separada o un valor de control.

**Campo:** un grupo en el tiempo de frame, formado por un número de bits (quizás sólo uno) teniendo un significado asignado a la posición particular dentro del frame.

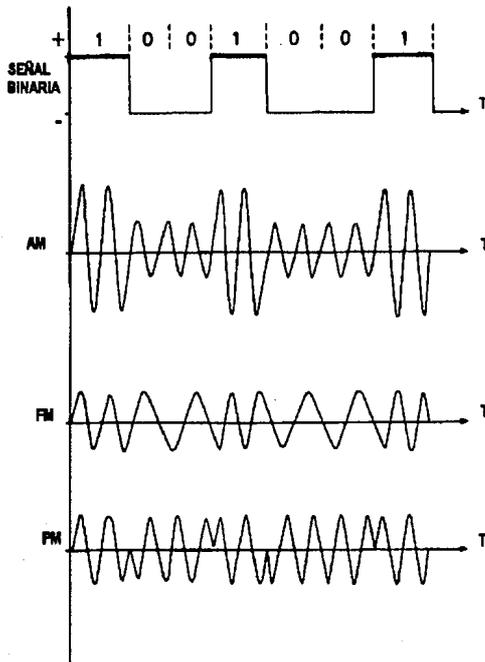
**Packet:** un pequeño grupo de dígitos teniendo una buena definición que puede formar parte de un mensaje digital completo.

El ancho de banda es uno de los requerimientos para la comunicación efectiva. El otro elemento es la potencia, la fuerza de la señal definida en términos del producto de corriente suministrada en el medio de transmisión a un nivel de voltaje dado.

Frecuentemente este es medido en términos del producto del cuadrado de la corriente y la impedancia. En canales de comunicación específicos un recurso puede ser mas valiosos que otro.

## MÉTODOS DE MODULACIÓN

En general existen tres técnicas que son aplicables tanto a señales continuas (analógicas) y señales discretas (digitales); estas nos dan como resultado la modulación de amplitud, frecuencia y fase. Figura I.3.3.



**I.3.3 Métodos de modulación de señales digitales**

- **Modulación de la Amplitud**

La multiplicación de una onda por una senoidal traslada de frecuencia su densidad espectral. Este efecto es conocido como propiedad de modulación, puede utilizarse para generar ondas de amplitud modulada y para demodularlas. La adición de una frecuencia portadora se usa para ayudar en la demodulación. Este principio se usa en la radiodifusión comercial de **(AM)**.

- **Modulación de la frecuencia.**

La variación de la frecuencia de una senoidal continua en proporción a la amplitud de una señal modulada, se llama modulación en frecuencia **(FM)**.

- **Modulación de la fase.**

La variación de la fase de una senoidal continua proporcionalmente a la amplitud de una señal modulada, se llama modulación en fase **(PM)**. Ambos son ejemplos de la modulación de ángulo.

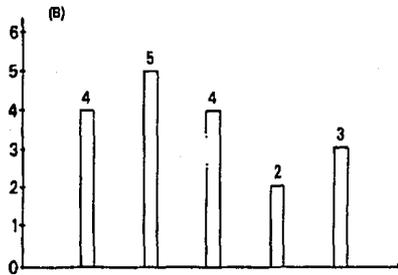
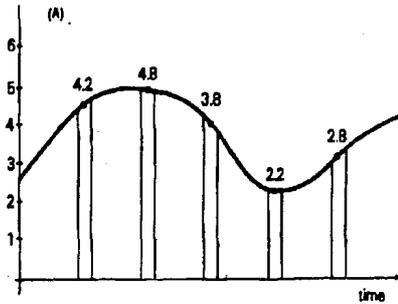
- **Modulación de Pulsos**

En la modulación de pulsos, se envían valores de muestra variando un parámetro en un tren de pulsos. En la modulación análoga de pulsos, la altura, el ancho o la posición de los pulsos varía en proporción directa a la amplitud de la señal en los instantes de muestra.

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

En la modulación de código de pulso, se cambia un código en etapas discretas, en proporción a la amplitud de la señal cuantizada en los instantes de muestra. El dispositivo que convierte una señal analógica, en señal digital para transmisión de datos es el codec (codificador/decodificador), similar al modem.

## FORMATOS TÍPICOS



**1.3.4 Modulación de Pulsos (a) Muestreo de la señal analógica, (b) Muestreo y cuantización de la señal, (c) señal digital.**

La principal técnica utilizada en la codificación es la modulación de

**código de pulsos (PCM), en esta los datos binarios son transmitidos codificando cada uno de ellos como un elemento de la señal.**

**El esquema de codificación es conocido como código fuente y sirve para mapear los bits en un elemento de la señal. Figura 1.3.4.**

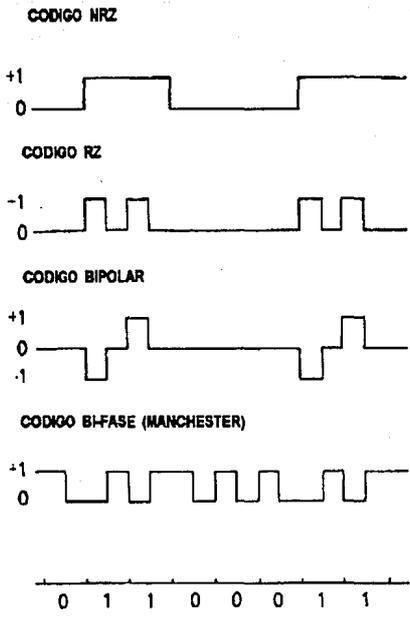
**La representación de una señal analógica para ser transportada por un canal digital requiere un proceso de conversión el cual involucra tres elementos:**

- **muestreo en el dominio del tiempo.**
- **cuantización en el dominio de la amplitud.**
- **codificación en una forma digital.**

**La señal digital que ha sido codificada para la transmisión, consiste de una secuencia de pulsos discretos discontinuos o bits de datos. Cuando estos son codificados son referenciados como un elemento de la señal.**

**En el caso más simple, existe una correspondencia entre los bits y los elementos de la señal, un 0 binario es representado por un nivel de voltaje bajo y un 1 binario es representado por un nivel de voltaje alto. Los esquemas mas usuales son: Figura 1.3.5**

- **No Return to Zero(NRZ)**
- **Return to Zero (RZ)**
- **Bi-polar**
- **Bi-phase (código Manchester)**



### 1.3.5 Esquemas de codificación

## **CODIGOS BINARIOS**

Los códigos binarios para representar eléctricamente los dígitos binarios 1 y 0 utilizan dos niveles de voltaje (ó elementos de señal), uno alto y otro bajo. Estos elementos de señal dan origen a los pulsos de la señal digital.

Una característica importante dentro de los códigos es lo concerniente al signo algebraico de los niveles de voltaje o los elementos de señal, se dice que un código es:

### ***Unipolar***

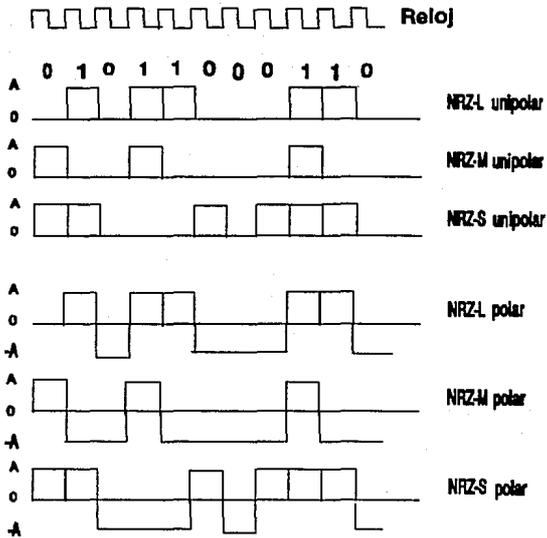
si los niveles de voltaje que se utilizan tienen el mismo signo algebraico y sólo difieren en magnitud. En la práctica por lo general estos niveles son un voltaje positivo  $A$  (nivel alto) y un voltaje 0 (nivel bajo)

### ***Polar***

si los niveles de voltaje que se utilizan tienen la misma magnitud pero signos algebraicos opuestos. En la práctica estos niveles son un voltaje  $A$  (nivel alto) y un nivel negativo  $-A$  (nivel bajo).

### **● NRZ (No Return to Zero) Sin retorno a Cero.**

Los términos marca y espacio por razones históricas se refieren a los dígitos binarios 1 y 0 respectivamente. Los códigos NRZ se pueden presentar en la forma unipolar o polar.



### I.3.6 NRZ (Not Return to Zero) Unipolar y Polar

#### NRZ-L

- 1 Nivel alto durante todo el intervalo del bit.
- 0 Nivel bajo durante todo el intervalo del bit.

#### NRZ-M

- 1 Transición de un nivel a otro al principio del intervalo del bit.
- 0 Sin Transición.

### NRZ-S

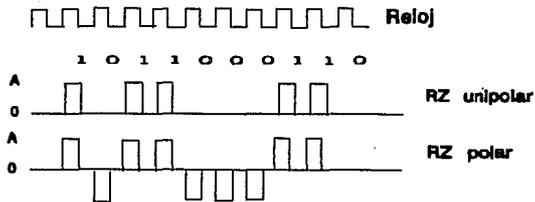
- 0 Transición de un nivel a otro al principio del intervalo del bit.
- 1 Sin Transición.

La mínima duración de los niveles de voltaje es igual al intervalo del bit,  $T_b$ . Esto origina que los pulsos que constituyen la señal digital tenga una duración mínima igual al intervalo del bit,  $T_b$ .

### ● RZ (Return to Zero) De retorno a Cero.

Los códigos RZ son de la forma unipolar o polar.

- 1 nivel alto durante la mitad del intervalo del bit. Se pasa a un nivel cero durante la otra mitad del intervalo del bit.
- 0 nivel bajo durante la mitad del intervalo del bit. Se pasa a un nivel cero durante la otra mitad del intervalo del bit.



### 1.3.7 RZ (Return to Zero)

La mínima duración de los niveles de voltaje ó elementos de señal es igual a la mitad del intervalo del bit,  $T_b/2$ . Esto da origen a que los pulsos que constituyen la señal digital tenga una duración mínima igual a la mitad del intervalo del bit,  $T_b/2$ .

- **PE (Phase Encoded) Codificación de Fase.**

Los códigos  $Bi-\phi-L$ ,  $Bi-\phi-S$ ,  $Bi-\phi-M$ , Manchester diferencial y DM pueden aparecer en la forma polar o unipolar, aunque en la práctica la forma polar es la mas utilizada.

**$Bi-\phi-L$  (Biphase Level) Bifásico de Nivel ó Manchester II**

- 1 Nivel alto en la primera mitad del bit con transición a un nivel bajo durante la segunda mitad del intervalo del bit.
- 0 Nivel bajo en la primera mitad del bit con transición a un nivel alto durante la segunda mitad del intervalo del bit.

**$Bi-\phi-M$  (Biphase Mark) Bifásico de Marca ó Manchester I**

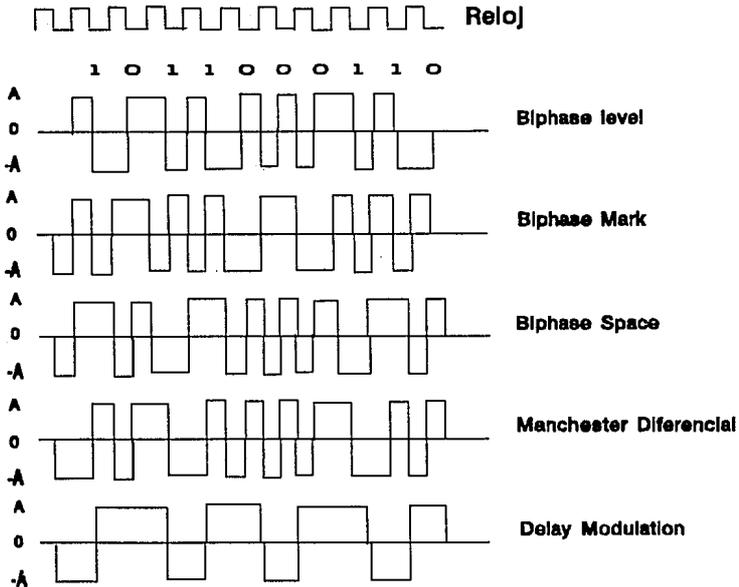
Siempre existe una transición de nivel de inicio de cada intervalo.

- 1 transición de nivel a la mitad del intervalo del bit.
- 0 sin transición de nivel a la mitad del intervalo del bit.

**$Bi-\phi-S$  (Biphase Space) Bifásico de Espacio manchester Diferencial.**

Siempre existe una transición de nivel al inicio de cada intervalo de bit.

- 0 transición de nivel a la mitad del intervalo del bit.
- 1 sin transición de nivel a la mitad del intervalo del bit.



### 1.3.8. PE (Phase Encoded)

#### Manchester Diferencial

Siempre existe una transición de nivel a la mitad de cada intervalo de bit.

0 transición de nivel al inicio del intervalo del bit.

1 sin transición de nivel al inicio del intervalo del bit.

**DM (Delay Modulation) Modulación por retardo o Código Miller**

- 1 transición de nivel a la mitad del intervalo del bit.
- 0 sin transición de nivel a la durante todo el intervalo del bit si el siguiente bit es un 1 Transición de nivel al final del intervalo del bit si el siguiente bit es un 0.

En los códigos Bi- $\phi$ -L, Bi- $\phi$ -M, Bi- $\phi$ -S, manchester Diferencial y DM, la mínima duración de los niveles de voltaje o elementos de señal es igual a la mitad del intervalo del bit,  $T_b/2$ .

Esto da origen a que los pulsos que constituye la señal digital tengan una duración mínima igual a la mitad del intervalo del bit. En el código DM la mínima duración de los niveles de voltaje es igual al intervalo del bit,  $T_b$ .

Los Códigos de Transmisión o Códigos de Línea en un sistema de comunicación digital son las diferentes formas de representar eléctricamente durante la transmisión a los dígitos binarios 1 y 0 que forman la información. El código NRZ-L es el código en que se generan e interpretan las señales digitales binarias en los circuitos electrónicos.

## **I.4 CARACTERISTICAS PRINCIPALES DE LAS SEÑALES DE VIDEO GENERADAS**

### **COLORIMETRIA**

La luz es la radiación de energía electromagnética que el ojo humano percibe y la sensación de color está directamente relacionada con la frecuencia de la luz observada. La longitud de onda de luz a las que el ojo humano es sensible, es de  $4 \times 10^{-8}$  cm a  $7 \times 10^{-5}$  cm.

Si un objeto es iluminado por luz blanca, refleja sólo las longitudes de onda que corresponden a su color y absorberá el resto de ellas; pero si luz con la que es iluminado no contiene las longitudes de onda correspondientes a ese color, absorberá toda la luz y no reflejará nada; a este proceso se le llama proceso sustractivo.

Los colores primarios: rojo, verde y azul son utilizados en televisión debido a la amplia gama de colores que se producen con su combinación; estos colores son llamados primarios aditivos ya que no pueden ser reproducidos por la combinación de los otros colores. Se escogieron los colores en función a los filtros factibles de fabricar para las cámaras.

El ojo humano tiene sensores que perciben estos colores, estos son los conos , células fotorreceptoras que se encuentran en la retina, y que

son capaces de percibir millones de colores a partir de los colores primarios. Para mayores detalles, recurrir al punto 1.1.

La NTSC (National Television System Committee) seleccionó los colores:

**Rojo** con una longitud de onda de 615 micras.  
**Verde** con una longitud de onda de 532 micras.  
**Azul** con una longitud de onda de 470 micras.

Los colores que se obtienen de la combinación de los colores primarios son los siguientes:

Color	Rojo	Verde	Azul
Blanco	X	X	X
Amarillo	X	X	
Magenta	X		X
Cian		X	X

Ahora bien, observando la tabla tenemos los complementos de los colores primarios: rojo - cian, verde - magenta y azul - amarillo.

El color blanco se obtiene de la combinación de los colores primarios con su respectivo complemento, o bien con la siguiente combinación: magenta y amarillo, amarillo y cian, azul y amarillo.

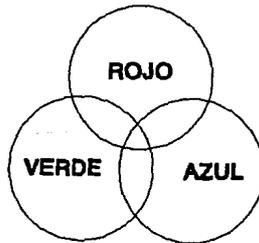
Los objetos que tienen color o matiz tienen tres características:

- Brillantez o luminancia
- Saturación
- Hue

**Brillantez** es la sensibilidad que tiene el ojo humano en función de la respuesta característica de un color. El ojo es más sensible al verde, es menos sensible al rojo y todavía menos sensible al azul. La brillantez es proporcional a la energía producida por el reflejo de un color.

**Saturación** indica la pureza de un color o la ausencia del blanco.

**Hue** es una indicación de la longitud de onda predominante en el reflejo al que es sensible el ojo.



#### **1.4.1 Colores primarios**

En el sistema de televisión a color, se transforma la luz en voltaje para transmitirlo, y al recibirlo se realiza el proceso inverso, se convierte

el voltaje en luz. Lo anterior se logra con un tubo cubierto de fósforo sensible a la excitación de un haz de electrones.

De acuerdo con la sensibilidad del ojo, la componente de brillantez de una cámara de tres tubos (RGB) se define:  $Y(\text{brillantez}) = 0.30R + 0.59G + 0.11B$ .

En el interior de una cámara la sección de óptica se compone de un juego de lentes que sirven para distinguir correctamente el objeto, otra sección se integra de un prisma que separa la luz en tres componentes de color rojo, verde y azul, los cuales inciden sobre los tubos que corresponden a los colores antes mencionados.

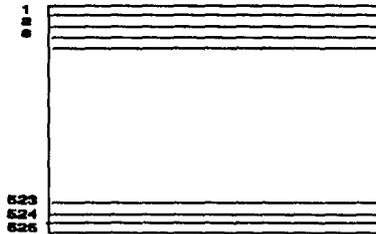
## ELEMENTOS DE IMAGEN

La televisión es básicamente un sistema para reproducir una imagen sin movimiento. Sin embargo las imágenes se muestran una sobre otra rápidamente para proporcionar la ilusión de movimiento.

Un cuadro de imagen es un grupo de pequeñas áreas de luz y sombra. Cada área pequeña de luz o sombra es conocido como "elemento de imagen".

Un cuadro en la televisión se forma de **525 líneas**. Las 525 líneas son exploradas en 1/30 segundos, por lo tanto los cuadros se repiten a una frecuencia de **30 Hz**. Figura 1.4.2.

Los cambios alternantes de brillante a opaco producen el parpadeo (flicker), este efecto se debe evitar. Por lo anterior, cada cuadro se divide en dos partes, así el cuadro será presentado 60 veces en un segundo, para evitar el parpadeo; este efecto se logra entrelazando las líneas horizontales, formando dos grupos de líneas, uno con las líneas pares y el otro con las líneas impares. Cada grupo de líneas pares e impares se llama "campo".



#### 1.4.2 Elementos de imagen

El período de barrido de una línea es de  $63.5 \mu\text{s}$ , por lo tanto la frecuencia es de  $15,750 \text{ Hz}$ , ahora bien si restamos el tiempo de borrado (tiempo de retorno de derecha a izquierda):  $65.5 \mu\text{s} - 10.2 \mu\text{s}$ , el tiempo que nos queda es  $53.3 \mu\text{s}$ , que es el tiempo para barrer una línea completa.

El límite de alta frecuencia es de  $4 \text{ MHz}$ , esto es  $1/(4 \times 10^6) = 0.25 \mu\text{s}$ , este es el tiempo que se necesita para explorar dos elementos adyacentes de una imagen, por lo tanto 8 elementos se barrerían en  $1 \mu\text{s}$ . Así  $8 \times 53.3 = 456$  elementos se pueden explorar durante un período de

línea activo de  $53.3\mu$ .

Ahora bien, el número de elementos verticales, se obtiene así: el número de líneas visibles es de 525 menos las líneas que dura el borrado vertical, con un tiempo de borrado vertical del 8% del total de líneas, esto nos resulta  $(525 * 0.8) / 100 = 42$ , y nos restan  $(525 - 42) 483$ , y con una razón de utilización de 0.7; finalmente tenemos el número de líneas útiles en el muestreo vertical  $(483 * 0.7) = 338$ .

Finalmente el número máximo de elementos de imagen visible es de  $(426 * 338)$  nos da como resultado un número cercano a 144 000, siendo esta cantidad independiente del tamaño de la imagen.

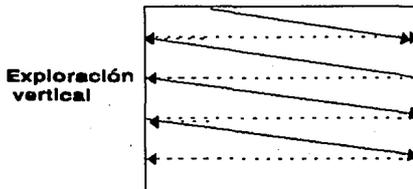
## EXPLORACION HORIZONTAL Y VERTICAL

Todo el desarrollo del método de exploración de un tubo de imágenes, parte de que éste maneja un haz de electrones sobre una pantalla cubierta de fósforo.

A este recorrido del haz se le llama exploración o barrido. Siempre se lleva la exploración en forma rectilínea y la velocidad en que se hace es constante. Esto es vertical y horizontal. Figura 1.4.3.

El barrido horizontal no es totalmente horizontal, es un poco inclinado, e inicia en la esquina superior izquierda y cuando llega a su límite a la derecha este haz regresa hacia la izquierda con mayor velocidad pero siendo invisible para el ojo humano.

### Exploración horizontal

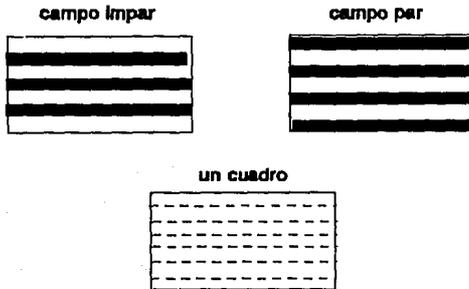


### 1.4.3 Barrido de haz

La persistencia de la luz emitida por el fósforo de la pantalla, permite que el haz incida dos veces en su recorrido y lo concluya antes de que se apague, siendo visible en la pantalla el cuadro completo. A esta remanencia de la luz de los dos campos se le llama **entrelazado**, porque el corrimiento de la media línea entre un campo y otro provoca que las líneas de barrido del primero, caigan exactamente entre las del segundo.

Los campos son llamados **non** y **par**, el primero cubrirá una mitad de la pantalla y el segundo la otra mitad para formar la imagen completa.

El barrido vertical, cubre las primeras y últimas 21 líneas debido a que la inercia no permite que se alcance instantáneamente la velocidad requerida y tampoco llega a cero, de igual forma; es por ello que el barrido horizontal debe cubrir estas áreas.



#### 1.4.4 Entrelazado de Campos

### PARAMETROS DE LA IMAGEN

El sistema de televisión en color es el mismo que el de blanco y negro. En un receptor de color, la señal de color se combina con la señal de brillo para recuperar las señales de video originales de rojo, verde y azul. Así, para la recuperación exacta se deben considerar varios parámetros:

#### Luminancia

Este nombre recibe la señal de brillo e indica la intensidad de luz, que percibe el ojo humano como brillo. En una imagen blanco y negro, las partes más claras tienen más luminancia.

### **Brillo**

El brillo en la pantalla depende de la magnitud de la alta tensión aplicada al tubo de imagen y de su polarización de d.c. en el circuito.

### **Contraste**

El contraste es la diferencia de intensidad entre las partes negras y blancas de la imagen, a diferencia de brillo que significa intensidad media.

### **Detalle**

También llamado resolución o definición depende del número de elementos de imagen que pueden ser reproducidos.

### **Matiz**

Se denomina color de un objeto, es más concretamente su matiz o tinte.

### **Saturación**

Los colores saturados son vívidos, intensos, profundos o fuertes.

### **Crominancia**

Este término se utiliza para indicar el matiz y la saturación. La saturación indica como esta diluido el color por el blanco.

## SEÑAL DE VIDEO COMPUESTA

La señal de video compuesta incluye las siguientes partes:

**a) La señal de cámara**

correspondiente a la información de la señal deseada. La señal de video correspondiente a la imagen completa contiene una sucesión de señales con formas de onda idénticas para cada línea de barrido horizontal.

**b) Impulso de borrado**

para que los retornos no sean visibles. La señal de video contiene impulsos de borrado para que las líneas de retorno sean invisible. Eleva la amplitud de la señal hasta el nivel de negro durante el tiempo en que los circuitos de barrido ejecutan el retorno.

El intervalo de borrado de cada línea horizontal es aproximadamente el 16% del tiempo total de barrido de la línea, esto es:  $63.5 \mu\text{s} \times .16 = 10.2 \mu\text{s}$ . Significa que el retorno de derecha a izquierda debe ser en  $10.2 \mu\text{s}$

**c) Impulso de sincronización**

para sincronizar el barrido del transmisor y del receptor. En el barrido del tubo de la imagen del receptor, el haz debe agrupar nuevamente los elementos de imagen de cada línea horizontal con la misma posición relativa de izquierda a derecha del transmisor.

Se transmite un pulso de sincronización horizontal y vertical, con una frecuencia de 15.750 Hz y 60 Hz respectivamente, para sincronizar el movimiento de barrido.

Las señales de sincronismo no producen exploración, son circuitos generadores de onda de diente de sierra (base tiempo), para la provisión de la deflexión del haz electrónico. Los impulsos del sincronismo hacen posible que la información de la imagen reproducida en la trama se mantenga estable en su posición correcta.

**d) El Burst de sincronismo de color**

El burst de color sincroniza la fase del oscilador de color de 3.58 MHz. Esta etapa reinserta la subportadora de color de 3.58 MHz en los demoduladores sincronos para detectar la señal de crominancia.

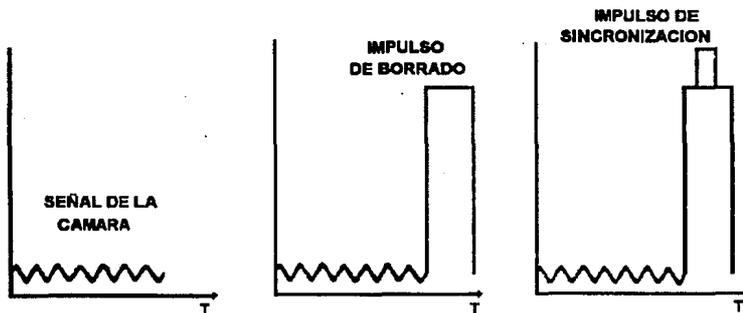
El burst y la señal de color son de 3.58 MHz pero el burst sólo se transmite durante el tiempo de borrado en que no hay información de imagen. La señal de color está presente durante el tiempo de traza visible de la información de color de la imagen.

La presencia o ausencia del burst determina el que un receptor de color reconozca si un programa es en color o monocromático.

Los impulsos de sincronización se transmiten como parte de la señal de imagen, pero son enviados durante los períodos de borrado en que no

se transmite información.

La señal compuesta de video se forma por una línea de exploración horizontal, además para el color se incluye la señal de crominancia de 3.58 MHz y el Burst o impulso de sincronización de color.



#### 1.4.5 Señal de Video

La secuencia en que ocurren estas señales es la siguiente: después de la señal correspondiente a la información de la imagen, el retorno ocurre durante el tiempo de borrado, así los pulsos de sincronización coinciden con el inicio de los retornos.

Esto es, un impulso de borrado inicia con la señal de video en negro; después ocurre la señal de sincronización; la secuencia se aplica a los retornos horizontales y verticales.

En el sistema de televisión a color, las señales recibidas de video son información de la imagen del rojo, verde y azul, los cuales son combinados para formar dos señales equivalentes, una correspondiente al brillo y otra correspondiente al color:

**a) Señal de luminancia.**

Contiene sólo variaciones de brillo de la información de la imagen. Se utiliza para reproducir la imagen en blanco y negro.

**b) Señal de crominancia.**

Contiene la información del color, la señal es transmitida como modulación en una subportadora de 3.58 MHz, por consiguiente la frecuencia para el color es de 3.58 MHz.

Se han inventado muchos sistemas de T.V. a color que se siguen perfeccionando, la diferencia entre sí es la forma como las señales de crominancia modulan la subportadora:

***NTSC (National Television System Committee) E.U***

transmite simultáneamente las dos señales de crominancia, que modulan la subportadora en fase y en amplitud.

***SEMAC ( Sequentiel A Memoire)***

***Francia***

las dos señales de crominancia, modulan alternativamente a la subportadora, mediante un conmutador electrónico, empleado en el receptor de la línea. Una línea de retardo memoriza las señales y las restituye en el momento oportuno.

### **PAL (Phase Alternance Line)**

### **Alemania**

es un perfeccionamiento de NTSC, transmite en un tiempo la señal de crominancia y en otro una señal para compensar los errores de fase, principal problema de la señal NTSC.

## **SEÑALES DE VIDEO DE ROJO, VERDE Y AZUL**

El sistema de televisión maneja las señales de video rojo, verde y azul. En el transmisor la luz de diferentes colores se convierte en las diferentes tensiones de señal de video. El tubo de imágenes del receptor convierte las tensiones de video de color en sus respectivos colores.

Una cámara de color tiene un tubo para cada uno de los colores, rojo, verde y azul. La pantalla del tubo de la imagen tiene fósforo rojo, verde y azul para reproducir estos colores mediante las señales de video.

Cuando la imagen es explorada en la cámara de color, ésta utiliza tubos separados para cada color.

## **LA IMAGEN EN EL PROCESAMIENTO DIGITAL**

El despliegue de una imagen es el último eslabón en la cadena del procesamiento digital de imágenes, el cual produce imágenes en niveles de grises como salida. Después de que todos los procesos se completan, el despliegue transforma la imagen digital en una forma adecuada para la

comprensión humana.

El ojo humano/cerebro es el último consumidor de los frutos del procesamiento digital de imágenes. El ojo es capaz de distinguir alrededor de 40 niveles de grises. Esto significa que el rango entre el negro y el blanco está dividido en 40 niveles equivalentes, los niveles de grises adyacentes parecen idénticos para el ojo humano. De cualquier manera, Hay un proceso de realce en la retina. Hay dos tipos básicos de desplegado, permanente y volátil:

- El desplegado permanente produce una copia de la imagen en papel, filmina u otro medio de grabación.
- El desplegado volátil produce una imagen temporal como en un desplegado en pantalla.

Los componentes básicos de un sistema de desplegado son similares al de un sistema digitalizador "scan-in". El sistema de desplegado difiere en que requiere de una luz de sensado, y la intensidad del rastreo (scan) de puntos se controlan por los valores del nivel de gris de la imagen que es desplegada.

El sistema de desplegado produce una imagen donde la densidad de cada píxel desplegado es controlada directamente por el nivel de gris del píxel correspondiente en la imagen digital.

## CARACTERÍSTICAS DEL DESPLEGADO.

En esta sección, discutiremos algunas características, que determinan la calidad de un sistema de despliegue de imagen digital. Las características de mayor interés son:

- el tamaño
  - fotometría y resolución espacial
  - respuesta a baja frecuencia
  - el ruido en el desplegado
- 
- **Tamaño de la imagen desplegada.**

La característica del tamaño de la imagen de un sistema de desplegado tiene dos componentes:

- El tamaño físico del desplegado, el cual debe ser suficientemente grande que permita una correcta inspección e interpretación de la imagen.
- El tamaño de la imagen digital, que es tan grande como el sistema de despliegue pueda manejar.

El desplegado debe ser adecuado para el número de líneas y el número de muestras por línea a lo largo de la imagen a ser desplegada. Actualmente, el rumbo en el procesamiento digital consiste en manipular imágenes con más y más píxeles.

- **Resolución fotométrica**

Para sistemas de desplegado, la resolución fotométrica se refiere al cuidado con el cual el sistema produce la brillantez correcta o el valor de densidad de cada pixel.

Algunos desplegados manipulan sólo 4 bits de datos, así producen sólo 16 distintas escalas de grises, sin embargo otros sistemas manipulan 8 bits de datos para 256 escalas de grises.

- **Linealidad de la escala de gris**

Otra característica importante del desplegado es la linealidad de la escala de grises. Significa el grado con el cual la brillantez o la densidad es proporcional a la entrada en el nivel de gris.

Cualquier dispositivo de desplegado tiene una entrada de nivel de gris para la salida de la curva de transferencia de densidad o brillantez. Para operaciones idóneas, esta curva debe ser razonablemente lineal y constante desde un despliegue hasta el siguiente.

- **Respuesta a baja frecuencia**

Consideraremos la habilidad de un sistema de desplegado que reproduce largas áreas de un contraste de nivel de gris, o "campo plano" ó región equidistante de amplitud de pixeles. Esta habilidad depende principalmente de la forma de la mancha en el despliegue, el espaciamiento entre las manchas, y la amplitud del ruido y la posición del ruido características del sistema de desplegado.

## ● Ruido

El ruido electrónico en un sistema de desplegado produce variaciones en la intensidad y en la posición de la mancha a desplegar.

- El ruido aleatorio en la intensidad del canal puede producir un efecto de sal y pimienta (punteado, no uniforme) particularmente visible en campos planos.
- Si el ruido es periódico y sincronizado con las señales de deflexión, este puede producir un patrón de barras.

En general la calidad del desplegado es adecuada al ruido, aleatorio y periódico, esta guardado bajo el nivel de gris en la amplitud. En la mayoría de los sistemas, este es muy malo.

## **TECNOLOGIAS DE DESPLEGADO**

- **Desplegado volátil.**

El uso más común es el tubo de rayos catódicos, la intensidad de la mancha del pixel varía con la posición para producir una imagen. Un monitor de televisión ordinario puede actuar como un desplegado de digitalización de imágenes si este está provisto con una señal de video. Desde el desplegado el rastreo es continuo, el desplegado puede ser continuamente "refrescado". El refresco puede ser tomado desde un almacenamiento digital o analógico.

- El desplegado fotométrico usa un tubo de rayos catódicos. El rayo del electrón cambia las características de luz y absorción, la imagen puede ser vista bajo iluminación externa.

- El desplegado láser puede ser construido usando el movimiento de espejos o de otros objetos para la deflexión del rayo y una celda Kerr para modular la intensidad del rayo.

- El desplegado por descarga de gas se hace intercalando una fina red entre dos hojas de vidrio levantando un arreglo rectangular de celdas conteniendo un gas ionizable. Usando técnicas de direccionamiento horizontal y vertical, las celdas pueden ser hechas por calor bajo la influencia de un potencial eléctrico permanente.

- **Desplegado permanente.**

El desplegado permanente más común es el grabador de película CRT. Esto es básicamente una cámara de película, montada frente a un desplegado CRT. Con la juntura abierta, la imagen entera es desplegada, pixel por pixel. Sólo un paso es necesario, no se requiere refresco.

La intensidad del pixel puede ser modulada por el control de brillantez del lugar o el tiempo de duración para cada pixel desplegado. La exposición del pixel en la película es normalmente proporcional al producto de la intensidad de exposición y tiempo de exposición.

Dado que el RTC esta sujeto a una distorsión geométrica y pierde el foco en la periferia, los grabadores de película de alta calidad incorporan corrección de distorsión de geometría dinámica y circuitos de corrección dinámica de foco para mantener la exactitud del foco.

# Capítulo

# III

## Sistema de Captura y Procesamiento

## **II.1 TARJETAS DIGITALIZADORAS DE VIDEO**

En la última década los sistemas de adquisición, procesamiento y despliegue de imágenes han adquirido una gran importancia debido, principalmente, a su enorme aplicación en las diferentes áreas de la ciencia y tecnología.

El procesamiento digital de imágenes exige excepcionales requerimientos del procesador de la computadora. Hasta hace poco tiempo, estas aplicaciones necesitaban de soporte especializado en minicomputadoras y supercomputadoras, siendo estos considerablemente costosos.

Sin embargo, el acelerado desarrollo tecnológico en la fabricación de circuitos integrados, ha permitido producir sistemas de cómputo con alta capacidad de procesamiento y almacenamiento, con lo que se abre la posibilidad de obtener un sistema para adquisición y procesamiento de imágenes a un costo relativamente bajo.

El objetivo principal en el procesamiento digital de imágenes, es permitir al usuario extraer información a través del estudio de los detalles de una imagen, aislando y expandiendo el área de su interés.

## **DISEÑO GENERAL**

El diseño que se presenta, es un dispositivo electrónico instalable en una computadora personal PC/XT/AT y compatible para adquirir imágenes a partir de una cámara de video.

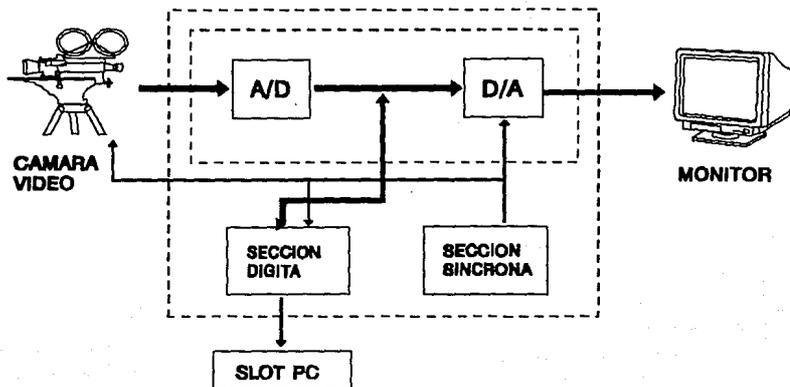
Un sistema, denominado digitalizador o tarjeta digitalizadora, cuenta con un conjunto de circuitos necesarios para poder desplegar en un monitor de video la imagen observada en una cámara de video, o bien una imagen previamente almacenada.

La resolución espacial de adquisición y despliegue, de la tarjeta, es de 512 x 512 pixeles o elementos de imagen, la resolución cromática es de 8 bits, lo cual permite trabajar hasta 256 tonalidades de gris.

El tiempo de adquisición de una imagen completa es de 1/30 segundos, que es el mismo tiempo que tarda la cámara de video en generar los dos campos (non y par) que componen una imagen.

Para el almacenamiento de una imagen, el sistema cuenta con una memoria RAM. Por un lado la memoria está conectada al digitalizador para adquirir y desplegar imágenes, y por el otro, la PC tiene acceso directo a la memoria como si fuera parte de su memoria.

El sistema de adquisición y despliegue se compone de tres partes, Figura II.1.1:



### II.1.1 Diagrama de bloques: Tarjeta digitalizadora de imágenes.

#### ● Sección de Sincronía

El digitalizador opera mediante sincronía interna, genera la señal de sincronía que requiere la cámara, y ésta a su vez genera la señal de video conforme a dicha sincronía. Incluye la lógica para la generación de las señales de control del digitalizador y la señal del reloj.

En esta sección se generan las señales para controlar las diferentes actividades que se realizan en la sección digital y de convertidores. Es

posible realizar la sincronización de la señal de video compuesto, proporcionado por la cámara de video, con el resto del sistema. También se sincroniza el flujo de información con el monitor de video de salida.

El modo de operación del sistema es mediante sincronización interna (free running). En este modo de operación, la cámara, el monitor y el digitalizador de imágenes aceptan las señales de sincronía.

Para su estudio, esta sección ha sido dividida en los siguientes circuitos: Oscilador, Divisor de Frecuencia y Generador de Sincronía.

El circuito generador de sincronías proporciona las señales de tiempo necesarias para manejar la cámara de video. Las señales de tiempo que entrega este circuito son las siguientes:

- **Sincronía compuesta**

se utiliza para sincronizar la cámara de video y el monitor de despliegue.

- **Blanking Compuesto**

esta señal indica el momento preciso en que empieza a desplegarse la información de video.

- **Las señales de blanking horizontal y vertical**

estas señales indican la sincronía por líneas (horizontal) y por campo (vertical) de tal forma que la sección digital está sincronizada con la información de video, tanto por campos como por líneas.

- **Field index**

se presenta al inicio del campo non, e indica el inicio de un cuadro. Sirve para sincronizar la sección digital con la información de video de tal forma que la información almacenada en los campos memoria se almacenen en el siguiente orden: campo non seguido del campo par.



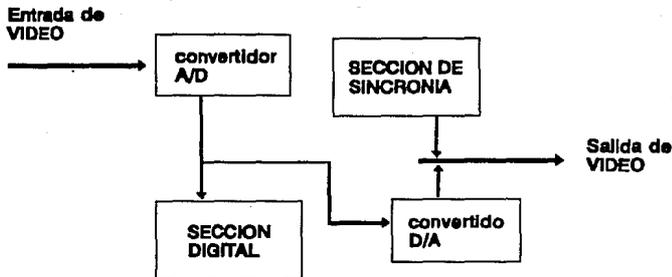
### **II.1.2 Diagrama de bloques de la sección de sincronía**

- **Sección de Convertidores**

Incluye los convertidores A/D y D/A así como la etapa de acoplamiento con la cámara y el monitor.

Esta sección tiene la función de convertir la señal analógica de video de entrada a su equivalente digital y también de convertir la información digital a una señal de video compuesto para manejar el monitor de salida.

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**



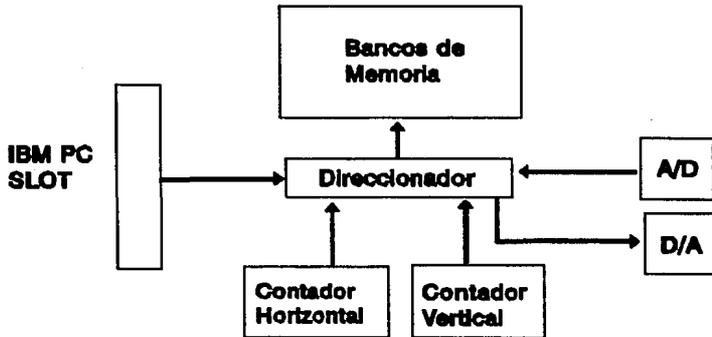
II.1.3 Diagrama de bloque de la sección de convertidores

### ● Sección Digital

La forman la memoria RAM, las interfaces con los convertidores A/D y D/A y la interfase con la PC. Además incluye la lógica de decodificación de la memoria y de los puertos de E/S.

Esta sección realiza la captura de la información digitalizada de la imagen y la almacena en un espacio de memoria. Asimismo, puede enviar la imagen almacenada en memoria a la sección de conversión D/A y de sincronía para generar la señal de video correspondiente.

La captura de la imagen debe estar en concordancia con las señales de sincronía incluidas en la señal de video. Cada línea de información en la señal de video viene precedida por un pulso de sincronía horizontal y cada campo, el cual totaliza la mitad del número de líneas, precedido por un pulso de sincronía vertical.



#### II.1.4 Diagrama de bloques de la sección Digital

El digitalizador puede ser utilizado en los siguientes modos de operación:

- Modo de adquisición.

una imagen es adquirida y almacenada en la memoria.

- Modo de despliegue.

la información que se encuentra almacenada en la memoria es desplegada en el monitor de video externo.

- Modo Normal.

la imagen que observa la cámara es desplegada en el monitor externo. La imagen pasa por el convertidor analógico/digital y digital analógico antes de ser desplegada.

## TARJETAS DIGITALIZADORAS

En la actualidad podemos encontrar una amplia gama de tarjetas digitalizadoras de video, ya que su principal aplicación comercial es la de formar parte de un equipo de Multimedia. Las tarjetas digitalizadoras de video se encuentran en el mercado para las plataformas más importantes, estas son: PC, PS, Macintosh y AMIGA.

Las características más importantes en la elección de una tarjeta son las siguientes:

- Resolución - es la cantidad de pixeles por unidad, existen en el mercado:

320 x 200, 640 x 480, 512 x 480, 512 x 486  
640 x 486, 712 x 486, 1024 x 1024

- Profundidad - se refiere a la cantidad de colores o tonos de grises

8 bits      256  
16 bits     65,536  
24 bits     16.7 M

- Entradas - se refiere al tipo de señal que se va a manejar a la entrada de la tarjeta.

S-video,  
RGB,  
Compuesto,  
Digital NTSC o PAL

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

- Salidas - se refiere al tipo de señal que se va a manejar a la salida de la tarjeta.

Las tarjetas más comunes para PC son:

#### **VIDEO BLASTER**

3 entradas de video compuesto,  
salida VGA  
2M de colores

#### **TARGA +**

Captura de imágenes  
1 entrada de video compuesto  
16.7 colores  
Tiene Alpha Channels para transparencias

#### **TRUE VISION**

32 bits por pixel  
entrada video compuesto  
salidas SVGA

## IMAGEN DIGITALIZADA

La señal analógica de la imagen es digitalizada después de que ésta es capturada por un scanner o una cámara de video. La señal es muestreada a una frecuencia múltiplo de la frecuencia de la señal de video. Por cada bloque de información que representa una señal (frame) que es digitalizada (capturada) el conjunto de datos es un arreglo de pixeles.

Dentro del sistema, cada pixel es un valor numérico que representa la intensidad (valor en la escala de grises) de la imagen en un punto dado. Para imágenes a color, por cada punto son almacenados tres valores.

Estos valores corresponden a la intensidad del componente rojo, verde, y azul. Los valores RGB, son utilizados para el despliegue de la imagen en el monitor de la computadora u otros dispositivos de salida.

El intervalo de muestreo digital aplicado a la señal de entrada determina el detalle con el que puede ser capturado. La **resolución espacial de adquisición y despliegue** describe el número de pixeles en los que ha sido subdividida la imagen. Este parámetro es generalmente expresado como  $H \times V$  (número de pixeles horizontales multiplicado por el número de pixeles verticales).

La escala de grises o la resolución del color, representa la precisión con la cual la intensidad del pixel es descrita en la base de datos. Este parámetro es el número de bits que deben ser almacenados por cada pixel.

La representación completa de una imagen a color requiere tres conjuntos de valores (RGB), así una imagen monocromática contiene sólo un conjunto de valores en la escala de grises. Este número de grises se expresa con la letra **B**. Es usualmente el tercer parámetro en la resolución espacial (**H x V x B**, donde **B** es el número de bits que deben ser almacenados por cada pixel).

Idealmente el paso de la imagen por el sistema de adquisición debe ser suficiente para capturar una imagen en tiempo real. esto es, el tiempo de captura no debe exceder el tiempo que se tarde en transmitir la imagen de la fuente de video.

Los sistemas operan a una velocidad de 30 Hz, la misma que la transmisión de video (NTSC). Sin embargo en el video NTSC, cada frame (conjunto de información transmitida) está subdividida en dos campos (non y par) que contienen las líneas alternadas adquiridas. Los campos son transmitidos a una frecuencia de 60 Hz, y son intercalados para producir una sola imagen a la salida del dispositivo de despliegue.

## DESCRIPCION DE LA TARJETA

Así una tarjeta típica incluye:

- procesador de despliegue
- procesador del frame
- procesador del arreglo

los cuales pueden trabajar independientes o juntos dependiendo de la aplicación. El bus de la PC es utilizado para la comunicación entre los procesos, Interface Process Communication (IPC), así las funciones de la tarjeta con controladas y coordinadas por el procesador de la computadora.

### ● **Procesador de despliegue**

Display Process Image (DPI), almacena los pixeles en la base de datos y provee funciones que permiten al operador manipular la imagen que se va desplegar. Algunas funciones permiten hacer un zoom (aumentar o disminuir la escala de la imagen), reducirla y ejecutar transformaciones de intensidad variando el RGB o la escala de grises.

Las transformaciones no son realizadas directamente en el arreglo de los valores de los pixeles. Por el contrario, estas operaciones son realizadas sobre las Lookup Tables (replica de los arreglos originales) que contienen las direcciones de los pixeles así como sus valores.

Si tenemos una Lookup Table por cada uno de los parámetros se aumenta la flexibilidad de manejo y la velocidad del procesamiento de la imagen.

La DPI básica ofrece una resolución espacial de adquisición y despliegue de 512 x 512 pixeles o elementos de la imagen, la resolución cromática es de 8 bits, lo cual permite trabajar hasta 256 tonalidades.

El tiempo de adquisición de una imagen completa es de 1/30 segundos, que es el mismo tiempo que tarda una cámara de video en generar los dos campos (non y par) que componen una imagen.

- **Procesador de frame**

Frame Process Image (FPI), realiza las funciones necesarias que permiten la entrada de la señal analógica de cualquier fuente. Una entrada común es el formato de video RGB, el cual provee tres componentes de color.

La sincronización de pulsos se maneja en un cuarto canal. También acepta formatos como NTSC (Estándar de Estados Unidos) o PAL (Estándar Inglés).

- **Procesador de arreglo**

Array Process Image (API), está especialmente diseñado para soportar las operaciones comunes en el procesamiento de imágenes, incluyendo operaciones espaciales, funciones aritméticas o lógicas y filtros.

## *Capítulo II.1*

La convolución es el principal algoritmo que es aplicado en el procesamiento de imágenes, el cual es aplicado en las matrices de píxeles.

El proceso de la convolución entrega el valor de un píxel basado en el promedio total de la escala de grises de una región de píxeles. En efecto, la convolución es un filtro que amplifica o resalta píxeles específicos en una imagen.

## **FUNCIONES BASICAS DE UNA TARJETA DIGITALIZADORA.**

Los sistemas de procesamiento de imágenes permiten al observador/usuario aplicar transformaciones a una imagen que ha sido capturada en un formato digital.

Estas transformaciones frecuentemente ayudan para realizar las características de cierta región, como la interpretación de las condiciones del clima a partir de una serie de imágenes de nubes. Algunas de esas transformaciones son las siguientes:

- **Región de interés.**

Una región de interés es un subconjunto de la base de datos que puede ser manipulada y transformada independientemente de las áreas colindantes. Esta área de interés puede ser de forma rectangular o puede tener cualquier forma.

- **Archivar.**

En el procesamiento de imágenes, la función de archivar imágenes provee las ventajas de almacenar y recuperar una imagen hacia o de un dispositivo de almacenamiento.

- **Digitalizar.**

El proceso de digitalizar una imagen, transforma una señal analógica de entrada de video en un arreglo de píxeles

discretos (elementos-imagen).

Digitalizar una imagen en tiempo real significa que una imagen de video puede ser capturada dentro del intervalo de duración de un frame (comúnmente 1/30 seg).

- **Substracción digital.**

En este proceso una imagen es extraída de otra para poder observarla más claramente.

- **Zoom y Minimización.**

Esta función permite agrandar o reducir una región de interés dentro de la imagen desplegada.

- **Pan, Scroll, Roam.**

Estas funciones son relativas al movimiento de la región de interés de un área de la imagen desplegada a otra. El Pan se refiere al movimiento a lo largo del eje X. El Scroll al movimiento a lo largo del eje Y. El Roam involucra ambos ejes.

- **Contraste.**

La distribución de la brillantez dentro de la imagen es el contraste. El análisis del contraste es una serie de operaciones de procesamiento de la imagen que pueden ser usados para alterar la brillantez promedio de una imagen para revelar

detalles o detectar orillas y bordes.

- **Pseudocolor.**

Esto es frecuentemente usado para realzar la densidad de pixeles específicos. Esta técnica ayuda al observador a interpretar las imágenes, ya que la imagen sufre una ligera degradación al ser capturada por los sensores de los dispositivos utilizados.

- **Análisis de la imagen.**

En un análisis de una imagen, la información numérica o gráfica es obtenida de las características básicas de la imagen original.

- **Filtros.**

Un filtro tiene el efecto de transformar la escala de grises (rango de intensidad) de la imagen. Las funciones de filtrado incluyen convolución y erosión o dilatación de la escala de grises.

- **Warp.**

Esta función transforma la geometría o la relación espacial en la imagen. Esta función incluye rotación (warp lineal) y distorsión (warp cuadrática).

- **Editor de imagen.**

**Un sistema procesador de imágenes puede ser usado para cambiar arbitrariamente la imagen. Las funciones de edit incluyen insertar texto, pintar, borrar, copiar, etc.**

## **II.2 SOFTWARE DEL PROCESAMIENTO DE IMÁGENES**

El software en el procesamiento digital de imágenes se organiza de diferentes formas ya que los requerimientos son diferentes según las plataformas de cómputo. Los compiladores y los sistemas operativos gradualmente llegan a ser torpes, debido al gran volumen de información que se maneja simultáneamente.

### **ORGANIZACION DEL SOFTWARE.**

El software en el procesamiento digital de imágenes que esté bien diseñado simplifica el desarrollo de programas y la producción del trabajo.

El software especializado no es estrictamente necesario, ya que el sistema operativo supe las rutinas de utilidad a ser llamadas y ejecutadas por programas almacenados en las librerías del usuario. El sistema que simplifica significativamente esta tarea promueve el procesamiento digital de imágenes entre los usuarios de diversos campos de la ciencia de la computación.

Tres grupos de software están envueltos en el procesamiento digital de imágenes, ellos son:

- **Sistema supervisor**

es un programa que controla la entrada y ejecución de trabajos.

- **Programa controlador**

trabaja con el sistema supervisor simplificando el manejo de datos y la ejecución de programas por parte de los usuarios.

- **Programas de aplicación**

trabajan con la imagen procesada.

- **Programas de aplicación**

Los programas que actualmente desempeñan operaciones de procesamiento digital de imágenes son programas de aplicación almacenados en librerías, éstos están escritos, depurados, documentados y cargados permanentemente en las librerías. El usuario selecciona los programas requeridos para desempeñar un función deseada.

El procesamiento es no destructivo, desde que la imagen de entrada es capturada, ésta es resguardada antes de ser transformada en una imagen de salida. El programa de aplicación usa la imagen de entrada para guiar la generación de una imagen de salida. La imagen es destruida sólo cuando el conjunto de datos es usado por la salida o es borrado al final del trabajo.

- **Programas de control**

En adición a la librería de los programas de aplicación, hay dos programas que comprenden el sistema:

- **Traductor del lenguaje**

lee los comandos de control y los usa para generar el lenguaje de control de trabajo que el sistema supervisor usa para controlar el procesamiento.

- **Supervisor residente**

inicia un trabajo y permanece residente mientras el trabajo no este terminado. Una porción maneja la entrada y la salida durante la ejecución de cada programa de aplicación.

## **ARQUITECTURA DEL SOFTWARE**

El software del procesamiento de imágenes permite la portabilidad en aplicaciones de imágenes. Incluye un conjunto robusto de algoritmos para el procesamiento de imágenes. Hay cientos de algoritmos disponibles, el 90% de todas las aplicaciones usan el 20% de los algoritmos, por ejemplo:

- Transformada geométrica (para rotar y escalar)
- Operadores de puntos (como suma y resta)
- Filtros (como paso-altas y paso-bajas)
- Morfología (como dilatación)

- **Visualización (exhibición).**

En los sistemas de redes existen ligadores entre el host y los recursos de la red en el manejo de imágenes. Requieren de un diseño modular y de un crecimiento fácil en los sistemas de software para interfases con plataformas futuras de hardware y productos de aplicación. Los sistemas también necesitan interpolar con otros sistemas, ejemplos librerías de graficación, interfases con usuarios y sistemas de ventaneo.

La integración de graficación con la manipulación de imágenes ha sido un problema tradicional. En algunos casos de manipulación de imágenes y graficación son desarrollados por compañías diferentes con pocas consideraciones de su conjunta operación. De hecho, la graficación fue desarrollada antes que la manipulación de imágenes, sus alcances no incluyen el procesamiento de píxeles.

Como resultado, los sistemas de graficación contienen inconsistencias básicas con los sistemas de manipulación de imágenes en cuatro áreas:

- 1) Los sistemas de graficación no contienen píxeles primitivos, solo geometría primitiva.
- 2) Los sistemas de coordenadas son básicamente diferentes porque los píxeles tienen dimensión (o área) y la locación de los vértices no.
- 3) Un sistema de graficación por coordenadas es una localización arbitraria en el espacio y no relativo a un

muestreo físico grid (cuadrículado).

- 4) Los modelos de ejecución para sistemas graficadores no son aptos para aplicaciones de manipulación de imágenes.

Algunas arquitecturas han sido propuestas para solucionar esta diferencia. Una aproximación es una combinación de manipulación de imágenes y gráficas en mapas binarios en un sencillo mapa binario virtual y entonces transferir los resultados a un sistema de ventaneo o de impresión.

Una segunda opción es resolver las inconsistencias graficación/manipulación de imágenes tal como gráficas en 2D y manipulación de imágenes. Cuando las gráficas son transformadas, los vértices son transformados y entonces se reubica el pixel en el espacio. Las imágenes son transformadas de acuerdo a este sistema habilitando imágenes-base, texto y gráficas. El orden en el cual las imágenes, el texto y las gráficas sean devueltos determina la visibilidad de cada elemento.

La tercera aproximación consiste en tener el sistema de graficación en pixeles para pasar al sistema de manipulación de imágenes. La aproximación inversa también ha sido usada en los sistemas de manipulación de imágenes donde éstos devuelven una pantalla y pasan al sistema de graficación una función bit-map para el desplegado.

## PRODUCTOS PARA EL MANEJO DE IMÁGENES

Existe Software para la manipulación de imágenes dependiendo de la plataforma en que se este trabajando.

**MACINTOSH** sobresale en el mercado de la digitalización de imágenes pues sus sistemas son muy compatibles en el manejo de imágenes sin consumir grandes recursos, uno de sus software es movie works. Movie works, consta de 4 programas que trabajan en equipo. el programa principal, Composer, esta suplementado por el editor Paint, el editor de sonido y el editor de texto.

Bajo el sistema 7, es necesario cargar solo el modulador Composer, los otros editores se cargan automáticamente, como se van necesitando. La comunicación es bajo escena Apple, un estándar de Macintosh para la comunicación interprogramas.

Los editores de dibujo, sonido y texto proveen los elementos básicos para la manipulación de las formas y datos. Dentro de Movie Works el video y el sonido son agregados como tareas de la presentación. La presentación en si misma siempre es una vista.

**AMIGA** ha desarrollado muchos sistemas en el manejo de imágenes, Scala es un sistema que nos permite manejar imagenes en plataforma Amiga.

**SCALA:** soporta el formato de cine ANIM, formato propio de Amiga, que no incluye sonido pero Scala puede digitar el sonido.

**PC** (Personal Computer) esta plataforma no fue creada para el proceso de imágenes es por eso que cuando se manejan imágenes se necesitan gran cantidad de recursos (memoria RAM, monitor de alta resolución, y gran capacidad en disco). Los equipo PC también cuentan con una gran variedad de productos de software, Animator, Topas, etc.

La mayoría de las aplicaciones de software que corran bajo ambiente Windows incluyen el manejo de imágenes, los procesadores de texto Word y WordPerfect permiten incluir en su trabajos imágenes, el manejador de bases de datos Paradox permite también el uso de imágenes.

Los formatos más utilizados para almacenar imágenes son los siguientes:

#### **PCX**

- maneja mapas de 8 bits
- es generalmente utilizado por programas de Windows
- resolución 1024 x 768

#### **TIFF**

- maneja mapas de 24 bits
- puede ser usado por la plataforma PC o Mac
- asigna un bit adicional a cada cuadro para tener un control de bits.
- 16.7M colores

#### **GIF**

- maneja mapas de 8 bits

- es un formato de intercambio de gráficos
- puede ser utilizado por varios tipos de plataformas
- utiliza formatos compactos

### **EPS**

- encapsulado Post-Script
- intercambia imágenes entre programas y plataformas

### **BMP**

- formato para mapas de bits
- generalmente utilizado por el entorno de Windows y OS/2

Así las imágenes son manipuladas y procesadas, utilizando una gran variedad de software, que en muchos casos su único límite es la imaginación. Actualmente las aplicaciones requieren de muchos recursos de hardware.

## **II.3 CARACTERISTICAS Y SELECCION DEL CODIGO DE BARRAS OPTICO**

### **INTRODUCCION**

Un **Código de barras** es la forma de representar información que contiene números u otros caracteres haciendo uso de una secuencia de barras paralelas, claras y oscuras, anchas y estrechas, las cuales son leídas por medio de equipos de lectura óptica.



#### **II.3.1 Código de Barras**

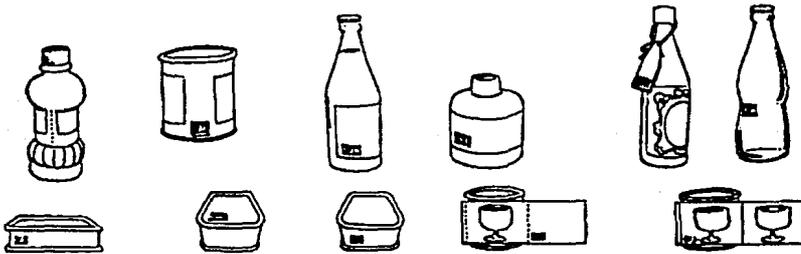
La información escrita mediante el código de barras puede ser la identificación de un producto (aplicación común para el sector industrial y comercial), código de acceso a algún área (usado como reloj checador o sistema de seguridad), o cualquier otro dato para una computadora.

● **Ventajas del uso de códigos de barras**

● La alta confiabilidad de los datos leídos y enviados a la computadora, gracias a que la lectura se hace por medios electrónicos y no por medios manuales los cuales tienen un alto porcentaje de error.

● La lectura de la información codificada es rápida y automática ya que se hace por medio de lectores ópticos que envían directamente la información a la computadora o bien la almacenan en algún dispositivo para después procesarla.

● Cada carácter está compuesto de barras paralelas y la interpretación que hacen los lectores no depende de la relación entre lo largo y ancho que éstas sean sino de la relación, entre las barras anchas y estrechas, claras y oscuras.



**11.3.2 Productos comerciales que utilizan códigos de barras.**

La información escrita mediante el código de barras puede ser utilizada en una gran variedad de aplicaciones, pero principalmente se utilizan en tres sectores:

- Automatización comercial
- Control de inventarios
- Sistemas de control de acceso, asistencia y productividad.

Existen varios patrones internacionales referentes al código de barras, por lo tanto, cuando se piensa en implantar un sistema donde los datos colectados se basen en este tipo de código, es necesario tomar en consideración algunos factores como:

- El tipo de dato que se va a manejar, ya sea numérico o alfanumérico, así como la cantidad de caracteres que éste contenga.
- El medio o material en que serán impresos los datos codificados, deberá tener una resistencia, y durabilidad adecuada a su fin.

La técnica de impresión utilizada debe ser capaz de generar códigos dentro de las tolerancias de anchura de las barras y de las propiedades ópticas del sistema.

En el caso específico de los equipos de control de acceso, éste no es mayor problema pues se provee de un software capaz de generar los códigos e imprimirlos en una impresora de matriz de puntos sobre papel común, para posteriormente ser leídos por el dispositivo de control de acceso y registro de asistencia. Esto lógicamente permite al cliente una

gran autonomía en la generación de documentos con códigos de barras.

El espacio disponible donde físicamente será puesto el código de barras. Se debe tomar en cuenta que todo código de barras requiere de marcas de START y STOP, de una zona de silencio inicial y Zona de silencio final, antes y después de los datos codificados.

Las zonas de silencio generalmente son un múltiplo de la anchura de un elemento angosto, son áreas que obligatoriamente deben estar libres de impresión y que anteceden y siguen a las marcas START y STOP.

En el ambiente de los negocios de comercialización de productos, se debe ser capaz de actualizar sus inventarios, clasificar sus productos, registrar ingresos por volumen, etc.

Para que esto sea posible los sistemas automatizados de comunicación están actualmente disponibles no sólo para las grandes corporaciones, sino para empresas medianas y pequeñas y muy fuertemente en el sector industrial.

Es por esta razón que surgieron los códigos de barras, existen varios tipos de códigos de barras, el estándar en México y Europa es el código EAN (European Article Numering).

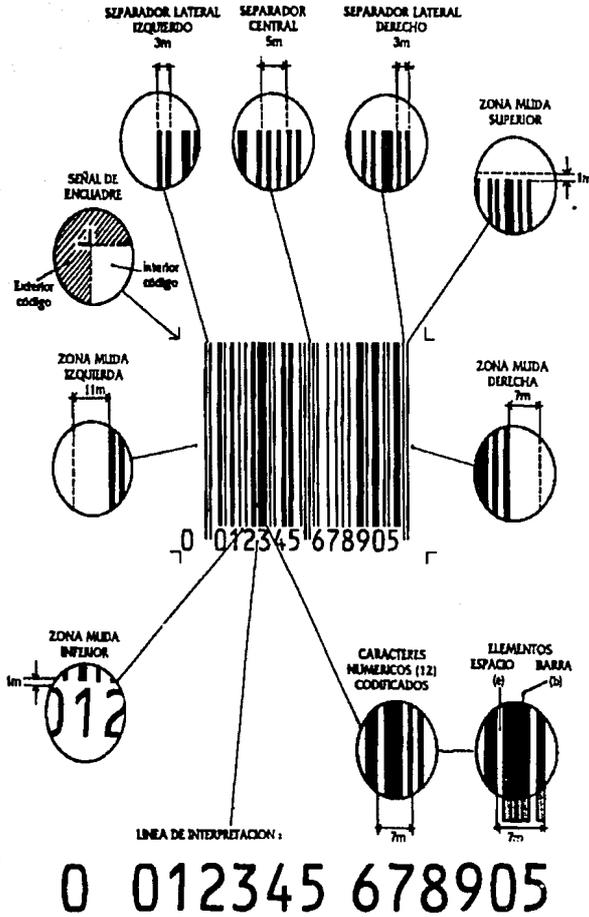
## **PATRONES DEL CODIGOS DE BARRAS**

Existen varios patrones internacionales de código de barras referentes al ancho y altura de las barras en el código, a la forma en que serán dispuestas unas detrás de otras y la manera en que serán interpretadas.

- EAN (European Article Numering)
  
- UPC (Universal Product Code)
  - UPC A
  - UPC E
  - UPC Truncado
  
- Código 2 de 5 (ó código 25)
  - Código 2 de 5 industrial
  - Código 2 de 5 matricial
  - Código 2 de 5 intercalado
  
- Código 3 de 9 (ó código 39)

### **Código EAN.**

Es un sistema de codificación constituido por series de barras y espacios, paralelos, de ancho variable, donde por lo general las barras son oscuras y los espacios claros; consta de una cantidad fija de barras (30 en total) y espacios (29 en total) que codifican información.



11.3.3 Código EAN.

Permite codificar teóricamente en 1000 países y 10000 industrias distintas y a cada una de ellas 100000 productos o formas de representación de los mismos, lo cual representa una enorme cantidad de posibles combinaciones.

Siendo éste un sistema común a varios países y productos dentro y fuera del mercado común europeo disponible, como es lógico, de un indicativo nacional llamado "Flag" para cada país que identifica al organismo nacional de codificación que asignará los códigos localmente.

#### **Características :**

- El carácter numérico es un número de un dígito (0 1 2 3 4 5 6 7 8 9); con estos números se forma el código EAN también conocido como EAN 13 pues precisa 13 caracteres.
- Cada carácter numérico se representa por 2 barras + 2 espacios, ubicados alternativamente, o sea 4 elementos para cada carácter; el ancho y ubicación de los elementos difieren a un carácter de otro.
- Se codifica cada módulo/espacio = 0. El ancho de cada carácter es fijo y mide 7 módulos (módulo es la unidad de menor ancho que forma los elementos).
- Los 4 elementos que forman un carácter tienen un ancho total de 7 módulos, es así que cada barra y/o espacio podrán tener un ancho como mínimo de 1 módulo, y como máximo de 4 módulos, siendo así un código de estructura compleja.

- Definimos la ubicación o posición de cada carácter en el código, mirándolo de frente, posición #1: la primera a la derecha del código y posición #13: la última a la izquierda del código.

### **Código UPC (Universal Product Code):**

Este tipo de código es uno de los más populares y adoptado por la industria norteamericana en 1973 para su lectura en las cajas registradoras de los supermercados (punto de venta); existen tres versiones:

- UPC A
- UPC E
- UPC truncado

### **Código UPC A**

Los 12 caracteres que forman el código, son representados e impresos por barras y espacios. Cada carácter numérico se representa por 2 barras mas 2 espacios, ubicados alternativamente, o sea 4 elementos para cada carácter; el ancho y ubicación de los elementos diferencia a un carácter de otro.

Significado de cada carácter:

- carácter # 12:  
categoría del producto (medicinal, alimenticio, etc.)
- carácter # 11, 10, 9, 8, 7:

identificación del fabricante del producto.

- carácter # 6, 5, 4, 3, 2:  
identificación del producto
- carácter # 1:  
dígito de verificación del código (check carácter).

Los caracteres # 1 y 12 se imprimen con barras más largas que las demás.



### II.3.4 Código UPC A

#### Características

- El ancho de cada carácter es fijo y mide 7 módulos (módulo es la unidad de menor ancho que forma los elementos).
- Los 4 elementos que forman a un carácter tienen un ancho de 7 módulos, es así que cada barra y/o espacio podrán tener un ancho como mínimo de 1 módulo, y como máximo de 4 módulos.
- Los criterios sólo se aplican a los 12 caracteres numéricos que

se codifican en el sistema UPC A, y no se aplican a los separadores ni en zonas mudas.

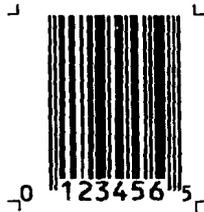
- Definimos la ubicación o posición de cada carácter en el código, mirándolo de frente, posición #1: la primera a la derecha del código y posición # 12: la última a la izquierda del código.

### **Código UPC E (código reducido)**

Esta versión se llama "Cero Suprimido" (zero suppressed) ya que elimina por lo menos 4 ceros en el código. No siempre es posible su uso ya que esto dependerá del número de fabricante y el número de productos asignados. Existen 4 formas de supresión de ceros, dependiendo de los tipos de números que le fueran asignados al fabricante y producto. Los artículos que podrán disponer de un código reducido UPC E, son por ejemplo:

- 1000 productos podrán ser codificados con UPC E, si el número de fabricante termina en 00, precedido por 0,1 o 2;
- 100 productos podrán codificarse, si el número de fabricante termina en 00, precedido por 3 al 9;
- 10 números de producto podrán ser asignados, si el número de fabricante termina en 0;
- 5 productos podrán utilizar la versión reducida, si el número del fabricante no termina en cero.

Aún así, todas estas condiciones, el número asignado al producto también debe comenzar con algunos "ceros" para que la reducción del código sea factible. Esta versión reducida UPC E del código UPC A se basa en un concepto de implementación distinto al que permite convertir un código EAN 13 en un EAN 8 y no deben ser confundidos.



### II.3.5 Código UPC E

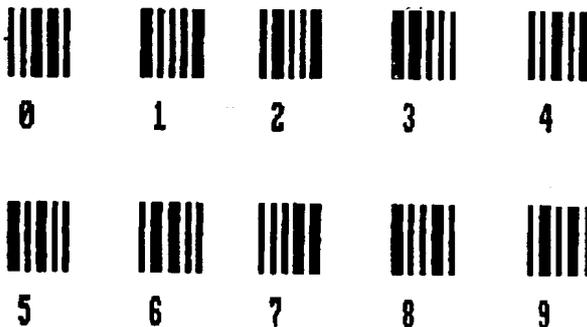
#### **Código UPC truncado**

Se trata de recortar la altura de las barras en la parte superior del código sólo se recomienda como última solución cuando la superficie disponible para el código es pequeña y la estructura del código no permite la implementación del código UPC E.

Cuando un código es truncado, no debe ser reducido también, porque esto reduce aún más la eficiencia del scanner, hace más lenta la lectura del código y reduce o anula la omnidireccionalidad en la lectura.

● **Código 2 de 5 (código 25)**

Este código recibe su nombre por las barras utilizadas de las cuales sólo 2 son anchas.



11.3.6 Código 2 de 5

**Características:**

- En el código 25 sólo hay posibilidad de trabajar con datos numéricos del 0 al 9.
- Para la codificación de cada dígito se utilizan cinco elementos de los cuales dos son anchos y el resto angostos. En la siguiente tabla se muestran la manera general en que estos 9 dígitos son codificados.

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

- El grueso de un elemento ancho, típicamente es de dos a tres veces el grueso de un angosto.

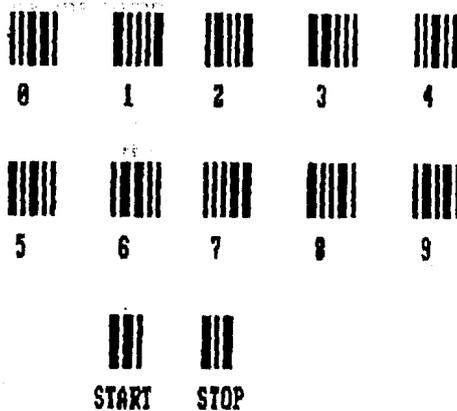
El código 2 de 5 cuenta con tres patrones de uso: industrial, matriz e intercalado.

### **Código 2 de 5 (Industrial)**

El código 25 industrial es el más antiguo de la familia del código 25. En este patrón cada carácter es representado por cinco barras separadas entre si por espacios angostos. En el código 25 industrial los caracteres son representados combinando barras anchas y angostas con espacios angostos los cuales no tienen ningún uso mayor que el de separar.

#### **Características:**

- Cada carácter se separa del siguiente por medio de un espacio, de tal forma que nunca dos barras oscuras angostas quedarán juntas pues podría causar confusión con una barra ancha.
- Las marcas de START y STOP, a diferencia del resto de los caracteres, se componen tan sólo por tres barras.
- La marca de START se forma por: dos barras anchas, una angosta.
- La marca STOP es formada por: una barra ancha, una angosta, una ancha



### 11.3.7 Código 2 de 5 Industrial

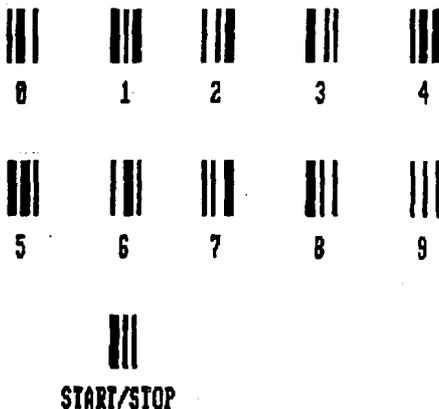
El código 2 de 5 usa dos tipos de resoluciones:

- 2 a 1, tendremos que los grueso de los elementos anchos es el doble que el grueso de los elementos angostos.
- 3 a 1, los elementos anchos son el triple de grueso que los elementos angostos.

#### Código 2 de 5 (Matriz)

Con este patrón de codificación del código 25 es posible tener una mayor densidad de información al compactar los caracteres utilizando, para representarlos, tanto las barras como los espacios. El código 25

matriz logra una compactación de los caracteres representados al permitir utilizar los espacios angostos dándoles un valor para representarlos.



### II.3.8 Código 2 de 5 Matricial

Esta técnica de codificación aprovecha los espacios que en el código 25 industrial no tienen más fundamentación que la de separar barras, dándoles un valor de tal forma que sean utilizados para generar y representar un mismo carácter de forma más compacta ahorrando de 28% a un 33% de espacio aproximadamente con respecto al código 25 industrial.

### **Características:**

- Cada carácter es formado por tres barras y dos espacios que bien pueden ser anchos, angostos o combinados; además, al igual que en código 25 industrial los caracteres son separados por espacios angostos.
- Las marcas **START** y **STOP** se simplifican a una misma codificación quedando formada por una barra ancha y dos angostas separadas por dos espacios angostos.

### **Código 2 de 5 (Intercalado)**

El código 25 intercalado logra una compactación aún mayor no sólo al permitir utilizar tanto espacios angostos como anchos sino al darles un valor, de tal forma que los caracteres no sólo se puedan representar con barras sino también con los espacios que separan las barras. Esto trae un ahorro aproximado del 36 % al 42 % respecto al código 25 industrial y de un 10 % al 12.5 % con respecto al de matriz.

El código 25 intercalado se logra insertando caracteres representados con barras y caracteres representados por espacios. En el código 25 intercalado el primer dígito del campo es codificado en barras y el segundo en espacios que separan a las barras del primer dígito, el tercer dígito sería en barras y el cuarto en espacios y, así sucesivamente.

### **Características:**

- Las señales de START y STOP también cambian:
  - La marca de START se compone de: una barra angosta, y un espacio angosto, una barra angosta y un espacio angosto
  - La marca de STOP se compone de: una barra ancha, un espacio angosto y una barra angosta.

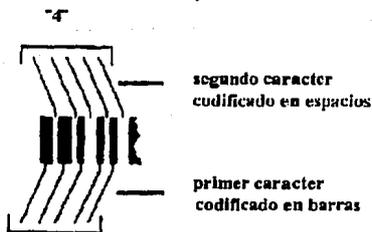


### **II.3.9 Códigos 2 de 5**

- El código intercalado es el más complejo para su interpretación y decodificación.
- El intercalado de caracteres también requiere que la información

codificada se componga de un número par de dígitos, pues en este proceso, por la forma de codificación de caracteres, sólo puede trabajar los dígitos de dos en dos.

- Si la cantidad de números de la información es impar, entonces se deberá agregar un cero (0) a la izquierda para completar una cantidad par.



### 11.3.10 Marcas: START , STOP

- **Código 3 de 9 (codigo 39)**

El código de barras sería muy limitado si sólo trabajara con números, por lo que se han establecido patrones de codificación que permiten el uso de caracteres alfanuméricos.

De estos patrones el más popular es el código 3 de 9 o código 39. Este código fue creado en 1974 y adoptado por el Departamento de Defensa (LOGMARS), Administración de Servicios (GSA) e Industria automovilística (AIAG) en Norteamérica aproximadamente en 1982, adoptado el SCS (shipping container system) luego llamado UNIFORM CONTAINER SYMBOL para depósito y distribución; diseñado originalmente para ser impreso sobre cartón corrugado.

El código 39 es capaz de representar 44 caracteres, los cuales comprenden: los números del 0 al 9, las letras mayúsculas de la A a la Z y algunos caracteres especiales.

**Características:**

- Para la codificación de los caracteres el código 39 utiliza 9 elementos, cinco barras y cuatro espacios. De esos 9 elementos tres son anchos (ya sean barras, espacios o ambos), de ahí su nombre "3" de "9".
- El código 39 es un código discreto porque para separar los caracteres representados, requiere de espacios que normalmente son angostos.

- Al igual que cualquiera de los patrones del código 2 de 5, éste también requiere de señales de START y STOP sólo que como en el caso de código 25 matricial, el código 39 utiliza una sola codificación, representada por un asterisco (\*), para indicar inicio y/o fin de dato.

1		0		I		R		-	
2		A		J		S		.	
3		B		K		T			
4		C		L		U		*	
5		D		M		V		\$	
6		E		N		W		/	
7		F		O		X		+	
8		G		P		Y		%	
9		H		Q		Z			

### 11.3.11 Código 3 de 9

Finalmente, después de diferentes pruebas en el sistema, se seleccionó el código 3 de 9 para la impresión de las credenciales generadas por el sistema.

## **II.4 CARACTERISTICAS Y SELECCION DEL SISTEMA LECTOR DE CODIGO DE BARRAS OPTICO**

Existen diversas tipos de sistemas lectores o scanners, fundamentalmente dependiendo de la forma de lectura, el destino del scanner y el tipo de fuente lumínica.

### **LA LUZ**

La luz, visible, a la que estamos acostumbrados es un fenómeno natural que nos provee diariamente el sol y las estrellas, la misma llega directamente o reflejada a los distintos objetos que la absorben y reflejan de distintas maneras, permitiéndonos "ver" y distinguir colores, tonalidades, etc.

Una aplicación particular de este mismo principio nos permitirá recoger y procesar las informaciones almacenadas en los códigos de barras, iluminándolos con cierta luz especial y viéndolos con un ojo electrónico muy particular que se llama scanner.

La luz se propaga con una forma de onda sinusoidal y se le identifica por su frecuencia (número de ciclos por segundo: c/s, o Hz) o por su longitud de onda (distancia entre el principio y fin de una onda o ciclo),

una es inversamente proporcional a la otra. la longitud de onda se mide en metros (m), o por sus múltiplos.

La luz del sol forma todo el espectro de colores visibles que van del color rojo al violeta (comúnmente identificados con el arco iris, teniendo en cuenta que el blanco es la suma de todos los colores y el negro es la ausencia de ellos) más allá del rojo y violeta el espectro es invisible y se llama infrarrojo y ultravioleta.

El espectro visible abarca longitudes de onda comprendidas entre 400 nm (violeta) y 700 nm (rojo); se llama infrarrojo a las longitudes de onda cercanas y mayores que 700 nm y ultravioleta a las cercanas y menores que 400 nm.

Los sistemas de scanning aplicados a los códigos de barras, utilizan por lo general, un rango de longitudes de onda que va del color rojo visible hasta el infrarrojo. A estas frecuencias se dan los fenómenos de absorción y reflexión de la luz por las barras y espacios del código dentro de los colores y contrastes que se especifican para este uso.

## **EL ESPECTRO ELECTROMAGNETICO.**

Los espectros visibles e infrarrojo se ubican, dentro de un rango de longitudes (o frecuencias) que se muestran en el siguiente esquema, comparándolas con las demás radiaciones electromagnéticas.

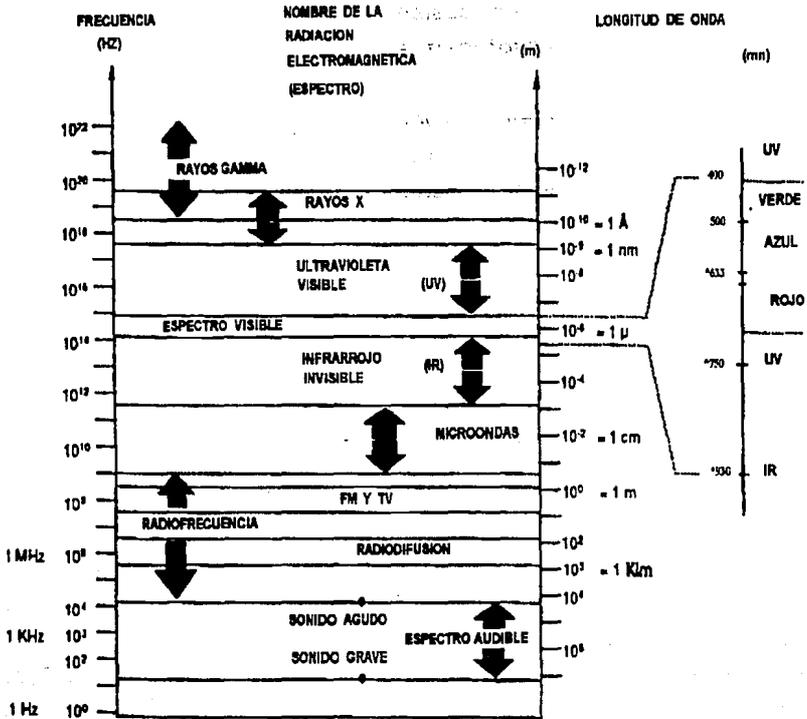
Los scanners se utilizan para los códigos de barras tradicionales, emiten y reciben en longitudes de onda fijas y predeterminadas que son básicamente:

**Rojo** : 633 nm - la más difundida para codificaciones comerciales, permiten el uso de scanners gaseosos (HeNe), es visible al ojo humano.

**Intermedio** : 750 nm - similar al rojo, parte de su espectro es IR, permite usar los diodos láser (de estado sólido) pero vuelve muy difícil la lectura de los colores verdes, azul, es relativamente invisible al ojo humano.

**Infrarrojo** : 930 nm - permite leer "a través" de ciertos filtros de seguridad y contaminantes (aceites, grasas), requiere tintas con alto contenido de carbón, no sirve para papel térmico orgánico, es totalmente invisible al ojo humano.

La elección de estas longitudes de onda son relativamente independientes del tipo de scanner y fuertemente dependientes del tipo de sustrato, impresión y objeto del código y su implementación.



### II.4.1 Espectro

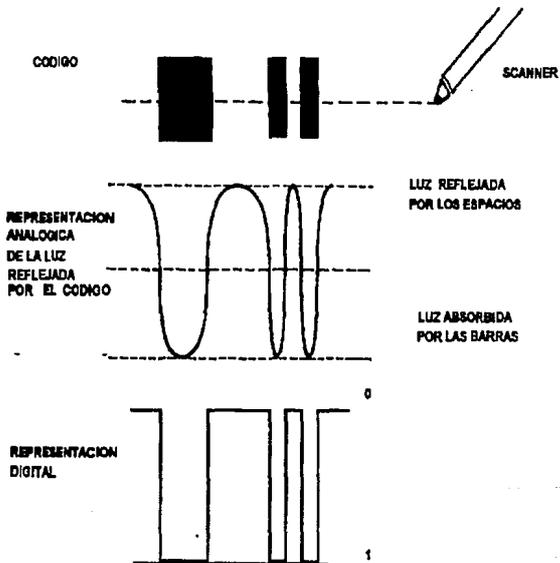
El scanner lee al código impreso de la siguiente forma: primero lo ilumina con una luz de frecuencia comprendida entre el color rojo y el

infrarrojo barriendo el símbolo horizontalmente, esta luz es absorbida por las barras (obscuras) y reflejada por los espacios (claros); el mismo scanner recibe la luz reflejada, transformando en digital la señal analógica.

Por lo tanto, el buen funcionamiento del conjunto "código impreso-scanner" depende de las características ópticas del símbolo (color, contraste, reflectancia) y la longitud de onda del scanner (rojo, intermedio, infrarrojo).

## CLASIFICACION DE LOS SCANNERS

Se llama **scanner** al instrumento lector óptico electrónico capaz de emitir y recibir un haz de luz rojo, infrarrojo, de tipo no-coherente (LED) o coherente (LASER); el scanner explora el símbolo iluminándolo a lo ancho y recibe el haz reflejado en un transductor óptico convirtiendo así la onda electromagnética (visible o no), en una señal eléctrica analógica.



### II.4.2 Scanner

Se llama decodificador al circuito electrónico que transforma esta

señal en digital (sucesión de 1 y 0 en forma de pulsos) y la procesa hasta descifrar su contenido y verificar que corresponde 100% a un código de barras para el cual fue programado, identificando al código y transmitiéndolo a un computador central. En algunos casos, el circuito decodificador se encuentra dentro del mismo scanner.

El objeto del scanner es la captura de datos (data collection) o el control de la calidad del símbolo. La combinación inseparable de scanner-objeto se divide básicamente en dos grupos, según el scanner sea portable o fijo, según se detalla a continuación.

## **A. SCANNER PORTABLE (HAND-HELD)**

Es aquel donde el objeto queda inmóvil y el scanner o la luz, se desplaza en forma manual o automáticamente.

### **a.1 manual, portable**

El scanner es desplazado en forma manual (manual wand), es de bajo precio y consumo, ideal para pequeños volúmenes de trabajo.

Permite una lectura por vez solamente (aproximadamente 1 lectura/segundo), requiere mínimo entrenamiento para mantener la velocidad y el ángulo de inclinación constante; no lee bien símbolos si no son de alta calidad y puede leerlos de cualquier longitud, su apertura es aproximadamente de 0.1 a 0.4 mm.

### a.1.1 scanner manual de contacto (contact wand)

Generalmente tiene forma de lápiz con transductores de estado sólido (LED), contiene uno a cuatro LED emisores de luz y normalmente una fotocélula (receptora).



### II.4.3 Scanner manual de contacto

Este es el tipo de scanner más económico y simple; sólo lee códigos planos (en dos dimensiones) y debe estar en contacto real con el código, su consumo energético es mínimo (3 - 10 mA/5 Vcc).

Se utiliza en la gran mayoría de los analizadores/verificadores portables ya que al estar en contacto directo con el código, los errores de enfoque casi son despreciables.

### **a.1.2 scanner manual, de aproximación (non-contac wand)**

Su costo es de 3 a 6 veces mayor que un scanner de contacto. El código debe mantenerse dentro de un estrecho rango de profundidad de campo y deben controlar 3 dimensiones, por lo tanto debe leer superficies planas e irregulares.

No lee bien símbolos pequeños ni truncados y puede leer objetos en movimiento en algunos casos, también sirve para símbolos en ubicaciones difíciles de alcanzar.

### **a.2 automático, portable**

El objeto queda inmóvil y la luz del scanner se desplaza en forma automática, sea por medios mecánicos o electrónicos (automatic hand-hel). Permite múltiples lecturas por segundo y lecturas redundantes. No requiere entrenamiento muy especializado y lee códigos impresos en mediana calidad.

#### **a.2.1 scanner automático**

Donde la luz se propaga en forma no-coherente. Esto quiere decir que se propaga simultáneamente en varias direcciones, no en fase y en algunos casos con distintas longitudes de onda, ejemplos: algunos diodos foto emisores (LED), las lámparas incandescentes, la luz del sol o de una vela. La energía se dispersa, por lo tanto se obtienen rangos de profundidad estrechos.

**a.2.1.1 scanner LED automático portable de luz no-coherente**

Los diodos fotoemisores son transductores de estado sólido, el scanner equipado con LED's posee una profundidad de campo limitada, menor de 15 cm y un ancho de lectura usualmente menor de 10 cm.

También pueden requerir de ciertos ajustes según la densidad del código, o sea que conviene utilizarlo para un tipo único de código rutinariamente.

**a.2.1.2 scanner CCD (descarga acoplamiento capacitivo) automático portable de luz no-coherente**

Es un dispositivo de contacto virtual, esto significa que lee por contacto real o hasta aproximadamente 2.5 cm de distancia permitiendo leer códigos sobre superficies irregulares o curvas.

Es generalmente muy liviano, pequeño con o sin decodificador incluido, con indicador de lectura auditivo y/o visuales.

Se activa por un interruptor o por presencia de códigos y viene en distintos anchos de campo, según el tipo de código. Ofrece en general un buen funcionamiento, sin entrenamiento previo y su promedio de lectura por segundo es de 50 a 100 según la marca.

Ilumina el código con un frente plano de luz roja (663 nm) y recibe el haz plano reflejado sobre un papel de micro-celdas donde por descarga capacitiva transforma la luz en señal electrónica a decodificar.



#### **II.4.4 Scanner CCD**

##### **a.2.2 automático portable láser**

Donde la luz se propaga en forma coherente (láser) o sea que todas las ondas son iguales, de la misma frecuencia, amplitud y además están en fase. Se produce una alta concentración de la energía y una dispersión tan pequeña que en condiciones adecuadas se podría iluminar la luna desde la tierra.

Por estos motivos, el scanner puede enfocar con presión a diferentes distancias permitiendo un alto rango de profundidad de campo, leyendo superficies irregulares o curvas.

El haz de luz se mueve por medios ópticos-mecánicos propios por lo cual su uso no requiere entrenamiento previo. Puede leer códigos de

alta, y baja densidad o truncados, por lo general sin ajustes, y la apertura típica del haz es 0.2-0.25 mm o sea una resolución media.

#### a.2.2.1 scanner láser, portable de estado sólido (LED)

Se basa en dispositivos electrónicos semiconductores conocidos como diodos foto emisores. Proporciona muy buena profundidad de campo (hasta 45 cm) y ancho de campo (hasta 28 cm).



#### II.4.5 Scanner láser

Es liviano, pequeño y generalmente trae el decodificador ya incorporado; de muy bajo consumo: 0.5-0.75 W, la luz es generalmente infrarroja (750-930 nm).

### **a.2.2.2 scanner láser, portable de estado gaseoso**

El transductor es un tubo lleno de gas, (generalmente helio-neón) donde se produce una alta excitación molecular, que genera un haz de ondas electromagnéticas coherentes, o láser; la luz usualmente roja (663 nm) y la profundidad y ancho de campo obtenibles son iguales o superiores a las del diodo láser.

El tamaño y consumo es obviamente mayor que las láser de estado sólido, y la vida útil del tubo láser es de aproximadamente 20000 horas.

## **B. SCANNER FIJO (FIXED MOUNT SCANNER)**

Es aquel donde el scanner queda inmóvil y al objeto se desplaza en forma manual o automática.

### **b.1 scanner fijo**

El objeto es desplazado en forma manual

#### **b.1.1 scanner fijo de haz simple-lineal (single beam).**

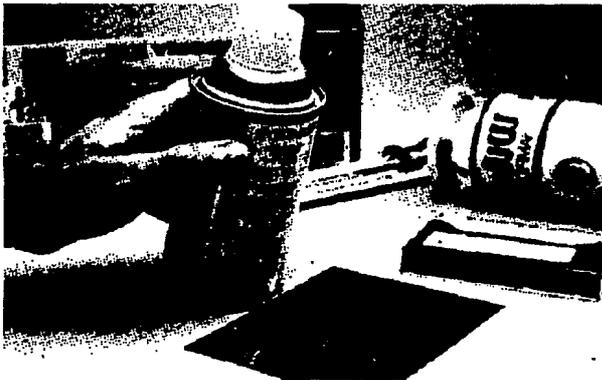
El haz de luz barre en línea recta al campo visual, produciendo una línea roja visible que permite enfocarlo correctamente. Generalmente es compacto, ocupa poco espacio y es fácil de ubicar incluso de mesa. Provee una gran profundidad de campo (60 cm) y generalmente opera con todos los códigos y en todas las densidades habituales.

**b.1.2 scanner fijo de haz multi-axial, de ranura (omni\_slot scanner)**

Es el tipo de scanner normalmente utilizado para punto de venta (POS), por ejemplo las cajas de supermercado. El haz de luz barre el campo visual en 3 o 4 líneas rectas simultáneamente (mult\_axial) y por lo tanto no requiere generalmente una orientación especial del símbolo.

Usualmente se ubica en la mesa de la caja registradora, horizontalmente, hacia arriba: también se ubica lateralmente. La ventana o ranura debe ser limpiada regularmente y reemplazada anualmente.

Procesan hasta 15000 ítems/hora, la profundidad de campo típica es hasta 25 cm de la ventana y la velocidad es de aproximadamente 500 lecturas/segundo. Su vida útil se estima en 10 años de uso normal.



**II.4.6 Scanner fijo de luz multi-axial**

**b.1.3 scanner fijo de ranura, económico (slot scanner)**

Es similar al anterior pero de bajo precio, a cambio de sacrificar parcialmente su utilidad, por ejemplo: el objeto debe permanecer semiorientado al scanner y no en cualquier posición.

**b.2 scanner fijo**

El objeto es desplazado en forma automática; generalmente para aplicaciones industriales.

**b.2.1 scanner fijo de alta velocidad, omni direccional (omni\_scanner)**

Este scanner permite alta capacidad de procesamiento para símbolos en cualquier posición a muy alta velocidad, especialmente en cintas transportadoras donde tamaño y posición del símbolo son impredecibles. El precio es alto pero garantiza alta productividad independiente de la orientación del código de barras.

**b.2.2 scanner fijo de alta velocidad unidireccional (raster-scan)**

Permite procesar símbolos a alta velocidad y con un costo significativamente más bajo, pero requiere una orientación del símbolo de forma que la línea de scanner intercepte necesariamente un juego completo de barras. Permite la lectura hasta 1 metro de distancia a alta velocidad.

**b.2.3 scanner fijo lineal (single line)**

#### **Capítulo II.4**

**Es un scanner de velocidad media, para cintas transportadoras de baja velocidad. Bajo precio comparado con los ejemplos anteriores y buena profundidad de campo. La posición del símbolo en el objeto se vuelve crítica para el scanner.**

**Se seleccionó el scanner manual de contacto (contac wand), ya que preciso y costo es bajo.**

# Capítulo

# III

# Diseño y Construcción

### **III.1 CARACTERISTICAS Y SELECCION DE LA TARJETA MANEJADORA DE VIDEO**

La tarjeta digitalizadora de imágenes que utilizamos es la Professional Image Board (PIB), esta tarjeta no fue seleccionada por sus características, como ya se dijo anteriormente, la Coordinación de la carrera de Ingeniería en Computación la compró como apoyo didáctico a la materia de digitalización de señales. Existían dos tarjetas, las cuales no eran utilizadas, así el propósito de la tesis es dar una aplicación, a estas para que las utilice la coordinación.

Las características fundamentales de la tarjeta son las siguientes:

- La tarjeta PIB es un dispositivo para capturar imágenes en "vivo" desde una cámara de video u otros dispositivos, con el propósito de salvar, almacenar, manipular y desplegar esas imágenes capturadas.
- La tarjeta PIB tiene una resolución de 512x256 y 256x256 píxeles.
- La tarjeta PIB usa uno de los dos estándares de video, NTSC o PAL. NTSC es el estándar comúnmente usado en Estados Unidos. En Europa y en muchas otras partes del mundo, el estándar es PAL.

Los requerimientos de software y hardware son los siguientes:

- Sistema de computadora personal: IBM PC, IBM XT, IBM AT, ATI-286, 368SX o 386, y cualquier sistema compatible con IBM, 640Kb de memoria RAM
- Tarjeta PIB
- Cable PIBCON
- Dispositivo de entrada: cámara de video, videocassetera, u otro dispositivo de video que presente la señal de video compuesta con el estándar NTSC o la señal RGB, que presente la señal de video compuesta PAL o la señal analógica RGB.
- Dispositivo de despliegue analógico RGB, monitor multisincrono que maneje el estándar NTSC de la frecuencia de barrido horizontal 15.75 Hz (15.62Hz para PAL) o un televisor estándar equipado con entrada de señal de video compuesta.
- Mouse
- Tarjeta manejadora de entrada/salida serial
- Sistema Operativo MS-DOS 2.0 o mayor.
- Alta capacidad de almacenamiento en el disco duro.

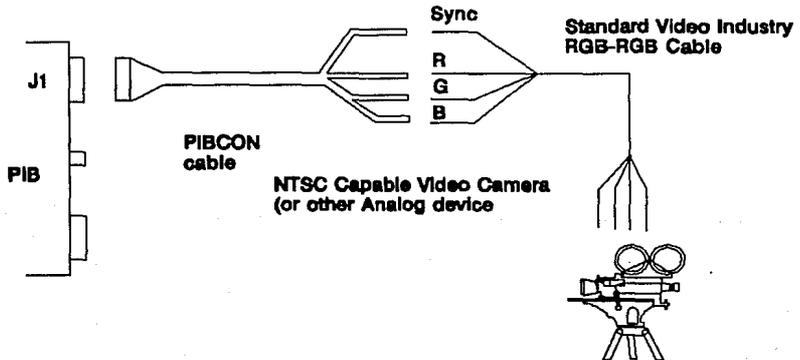
## ENTRADAS

Existen tres posibilidades de entrada a la tarjeta PIB.

- Entrada analógica RGB
- Entrada de video compuesto
- Entrada EGA.

### Entrada analógica RGB.

Algunas cámaras proveen de una dispositivo para manipular la señal. Estas cámaras (o los adaptadores) usualmente tienen 4 conectores etiquetados con Video, R, G y B.



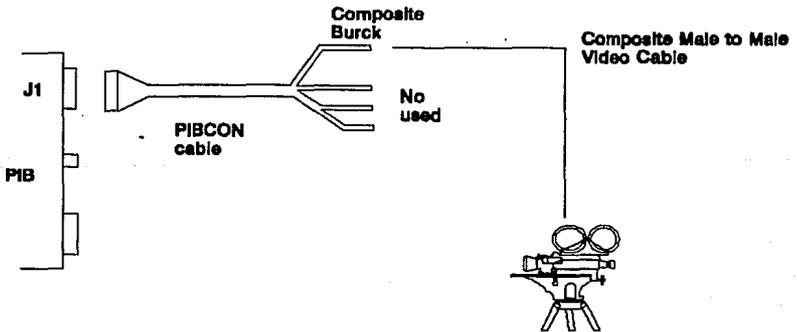
### III.3.1 Entrada Analógica

También tienen un conector macho al final. En la industria del video hay una impedancia estándar SEÑAL DE VIDEO.

Los cuatro conectores hembra del cable se conectan a los respectivos del machos del cable PIBCON. El cable del video es negro, los otros 3 rojo, verde y azul se conectan respectivamente (Red, Green, Blue). El otro terminador del cable PIBCON es DB9 macho conector J1.

### Entrada de video compuesto standard NTSC.

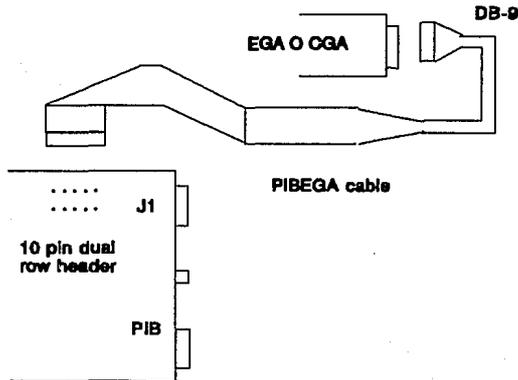
Algunas cámaras y otros dispositivos tienen el estándar NTSC para el manejo de señales. El video compuesto es uno de los usados por la televisión. Los dispositivos estándar de señal compuesta usualmente tienen un conector macho del tipo RCA etiquetado con VIDEO o COMPOSITE. Este tipo de conexión fue el usado en el sistema.



### III.3.2 Entrada de video compuesta NTSC

## **Entrada EGA (Enhanced Graphics Adapter).**

Es posible usar como un monitor EGA directamente en la tarjeta PIB, usando el conector PIBEGA que va del DB9 del monitor al conector J3 de la tarjeta. Este tipo de monitor debe ser analógico de otra manera no hay sincronía en la imagen.

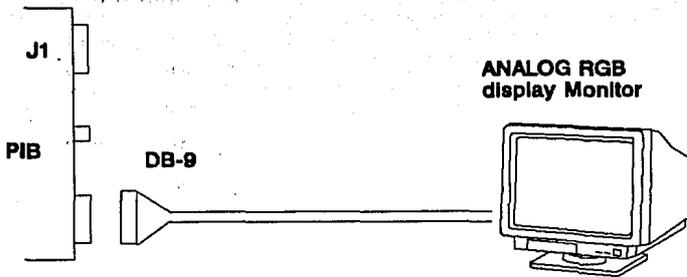


### **III.3.3 Entrada EGA**

## **SALIDAS**

### **Salida Analógica RGB.**

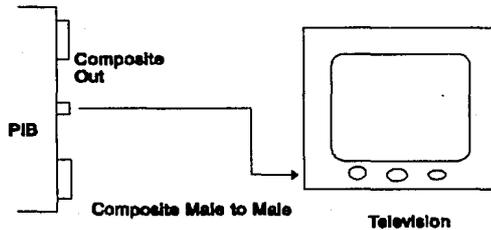
En esta configuración un monitor analógico puede usarse como desplegado de imágenes, conectando el DB9 macho en el monitor RGB directamente en el conector DB9 de salida (J2).



### III.3.1. Salida Analógica

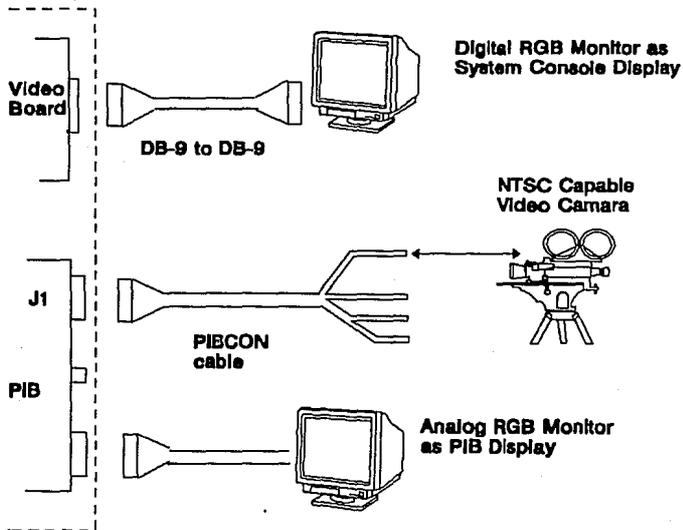
#### Estandar Compuesto NTSC.

Este metodo permite que PIB mande la salida a cualquier dispositivo de video compuesto, estandar NTSC en el conector de entrada. Se usa la entrada compuesta de video a través de un conector RCA.



### III.3.5 Salida Compuesta NTSC

La configuración usada es la llamada de escenario doble. El sistema de consola opera las funciones y las aplicaciones normales que corren en la tarjeta de video de la máquina. PIB esta conectado en un monitor RGB que se dedica al desplegado de imágenes PIB. Este configuración permite el manejo de imágenes libremente para el usuario, mientras que los comandos se usan para el manejo de imágenes son desplegados en sistema normal.

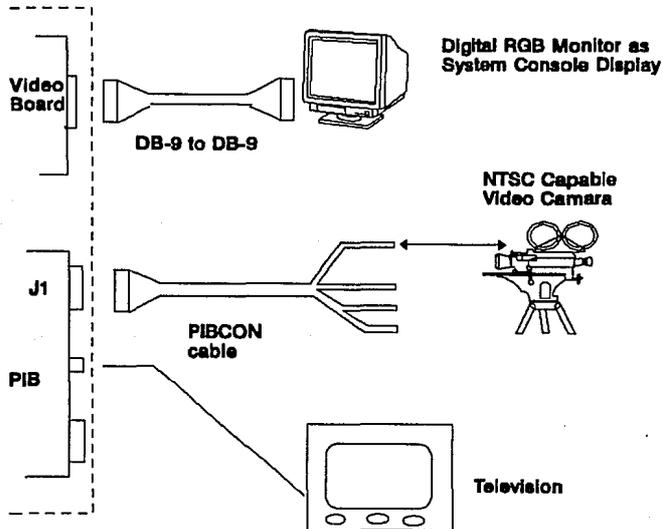


### III.3.6 Escenario con dos monitores

### Configuración Empleada

Finalmente se implemento el modelo de entrada de Video Compuesto Estándar NTSC utilizando una cámara de video, y de salida el modelo de Video Compuesta NTSC Estandar utilizando un televisor.

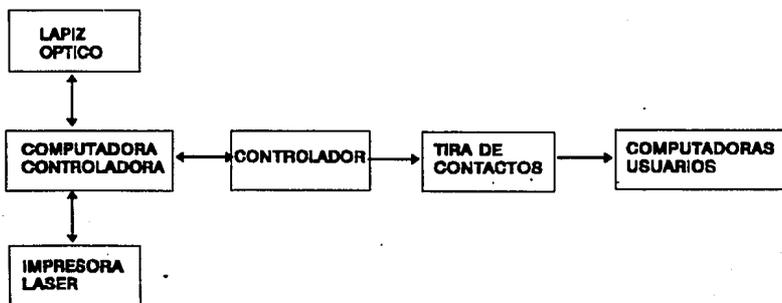
En un principio se intentó eliminar el televisor, teniendo como único medio de despliege un monitor, pero las pruebas fracasaron ya que no fue posible conseguir un monitor analógico RGB.



III.1.7 Configuración Final

## **III.2 DISEÑO Y CONSTRUCCION DEL CONTROLADOR DE EQUIPO**

El sistema en conjunto se resume con el siguiente diagrama de bloques, esto es: la computadora central, que se encargará del manejo de la base de datos de los alumnos y asignación de tiempos, esta a su vez interactúa con la impresora que imprime las credenciales, y el lápiz óptico que lee el código de barras para facilitar el trabajo al operador.



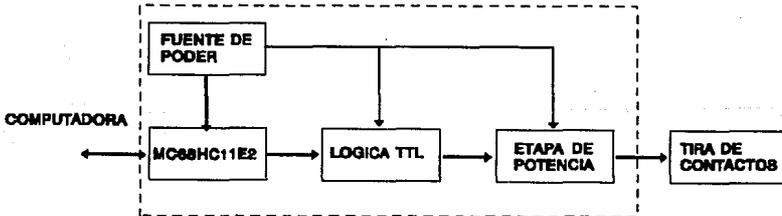
### **III.2.1 Diagrama de bloques del Sistema de captura, de control y asignación de equipo.**

Los otros componentes son: el controlador y la tira de contactos. El

controlador recibe información de la computadora central, que le indican que computadora tendrá voltaje para su funcionamiento y cual no. La tira de contacto es parte de los periféricos del controlador.

Finalmente este sistema conservará la integridad de las computadoras de los usuarios.

En este punto se explica detalladamente el diseño del controlador de equipo, este módulo cuenta con una fuente de poder, un microcontrolador MC68HC11E2, un bloque lógico de selección y una etapa de potencia. El sistema controlador es una interfase entre la computadora central (lleva el control de acceso) y las computadoras (para usuarios) que forman el centro de cómputo.



III.2.2 Diagrama de bloques del controlador

## **MICROCONTROLADOR**

Para la implementación de la tarjeta se seleccionó como unidad de control y procesamiento un microcontrolador de la familia 68HC11 de Motorola, tomando en cuenta los siguientes factores:

- Velocidad de operación
- Número de instrucciones disponibles
- Facilidad de adquisición
- Información disponible

Las ventajas de usar un microcontrolador es que incluye en el mismo chip los siguientes circuitos:

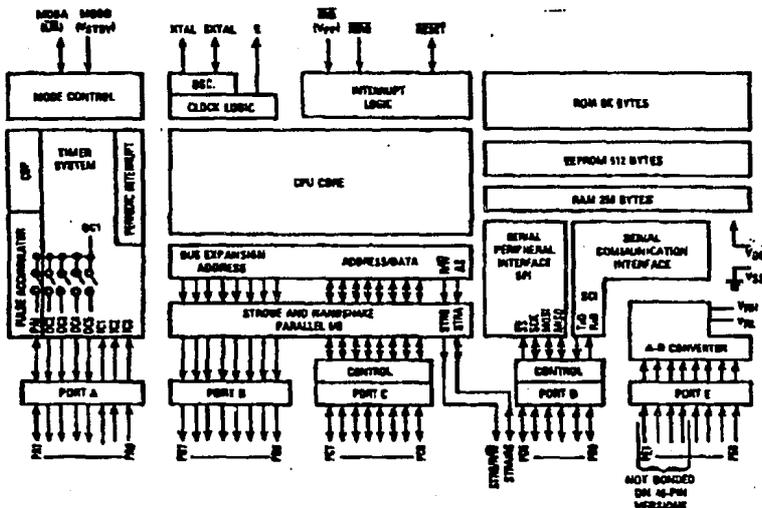
- Puertos de entrada y salida
- Unidad de transmisión asíncrona, (para comunicación con un puerto RS-232).
- Conversores análogo-digitales

Se seleccionó el 68HC11E2 de Motorola, que trabaja con un bus de 8 bits a una velocidad de 2MHz.

Los sistemas incluidos en el circuito 68HC11E2 son:

- Cinco puertos programables de entrada/salida.
- Un convertor análogo-digital de 8 bits de 4 entradas.
- Una memoria interna RAM de 2K bytes.
- Un puerto de comunicaciones serie.

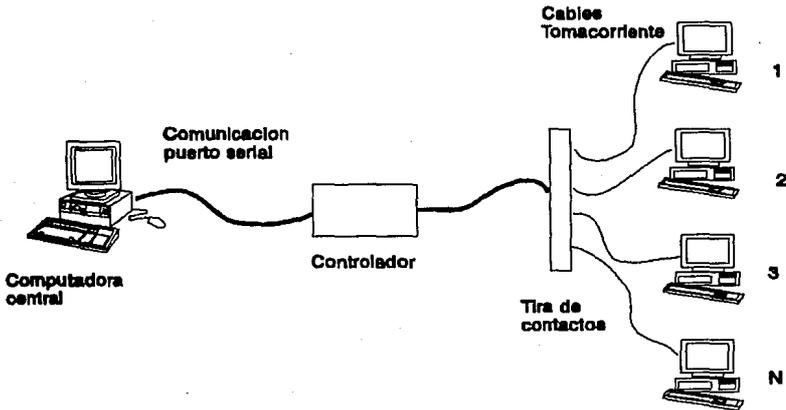
- Una EEPROM interna de 2K bytes.



### III.2.1 Diagrama de bloques del Microcontrolador 68HC11

El sistema que se diseñó tiene como propósito controlar el acceso a un centro de cómputo, para preservar la integridad del equipo. El

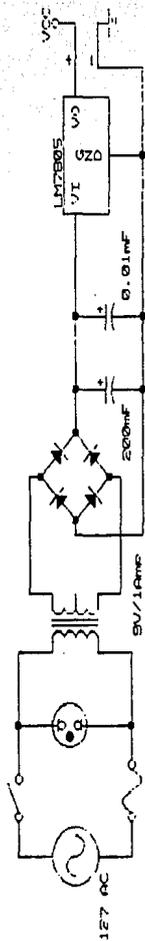
sistema protege al equipo, restringiendo el paso de corriente a las clavijas de las computadoras.

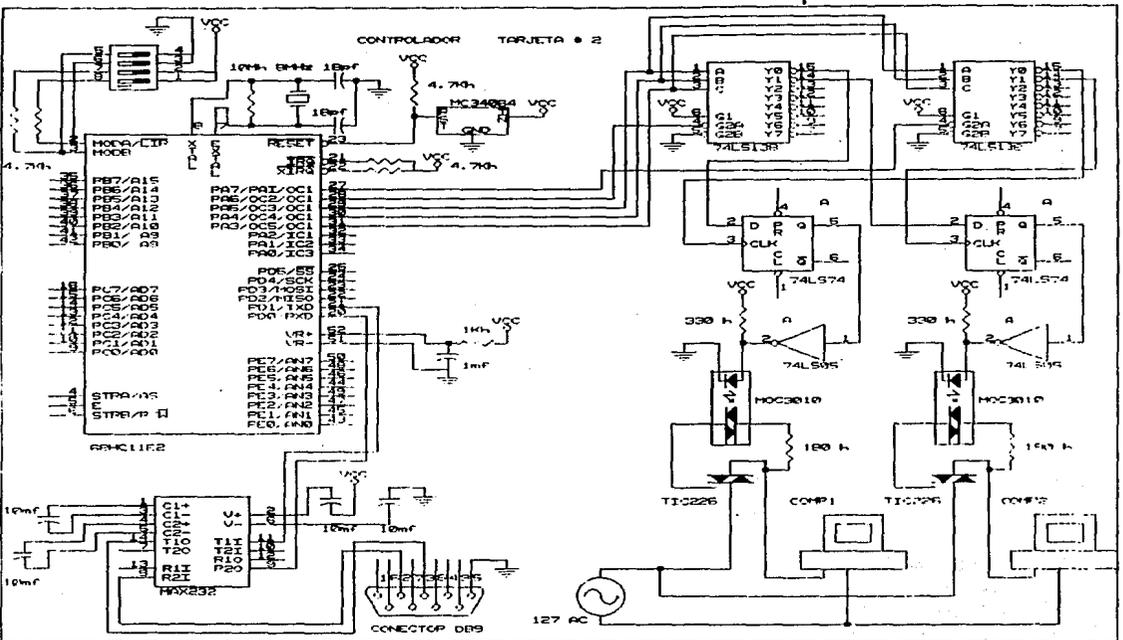


### III.2.2

El microcontrolador está funcionando en Modo extendido, esto quiere decir que se tiene acceso a una memoria externa (EPROM), de la cual se van a leer las instrucciones a ejecutar, en nuestro caso no utilizamos ninguna memoria externa, se utilizó la EEPROM interna debido a que su utilización resultó mas factible. Cuando se utiliza una memoria externa es indispensable tener acceso a un grabador y un borrador de EPROM's, ya que en el momento de programar es común que surjan cambios y los datos ya almacenados en la memoria deban ser sustituidos por otros. Así, el microcontrolador se alambrió en modo extendido, (se anexa diagrama de conexión).

FUENTE DE PODER 5V/1Amp TABUETA • 1





Para implementar la comunicación serial, entre el microcontrolador y la PC, se utilizó el Puerto D del microcontrolador, que tiene la lógica necesaria para el manejo de la comunicación asíncrona, los pines son PDO(RxD) y PD1(TxD).

El puerto serial de la computadora, utiliza los voltajes de -12v y +12v, y el circuito TTL de 0v y +5v. Este problema se solucionó utilizando el circuito integrado MAX232 el cual hace los cambios de voltaje necesarios, la ventaja que se observa, es que el circuito se polariza con 0v y +5v, lo cual permite un ahorro de recursos.

## **BLOQUE LÓGICO DE SELECCIÓN**

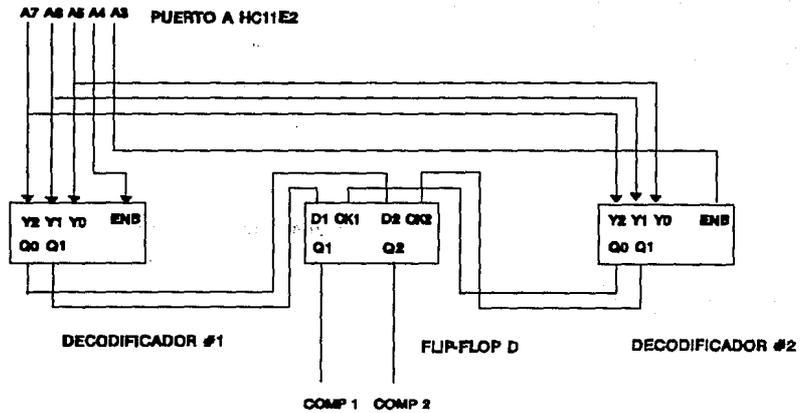
A la salida del Microcontrolador, se utilizaron 2 decodificadores 3x8, y un flip-flop D; el primer decodificador recibe los datos para seleccionar una de las computadoras, esta salida seleccionada del decodificador se encuentra conectada a la entrada de dato "D" del flip-flop D; el segundo decodificador tiene los mismos datos del primero, pero su salida esta conectada al "reloj" CK del flip-flop D.

### **Ejemplo:**

Quando se selecciona la computadora #1 de los usuarios (quiere decir que el decodificador #1, tiene un dato en la salida "Q0", que tendríamos en la entrada del decodificador  $C=0, B=0, A=0$ ) el flip-flop D correspondiente a esa computadora tendrá un dato en "D" (este dato indica prender=0, apagar=1), el flip-flop cambiará a su estado correspondiente sólo si existe un pulso de reloj.

En este momento interviene el decodificador #2, que tiene los mismos datos de selección que le decodificador #1 y su salida "Q0" está conectada al reloj "CK" del flip-flop D, el decodificador se habilita y deshabilita para simular el pulso de reloj.

Así con los datos de selección y el control de las entradas de habilitación de los decodificadores #1 y #2 (salidas A3,A4,A5,A6,A7), se selecciona una de las computadoras, para prenderla o apagarla.

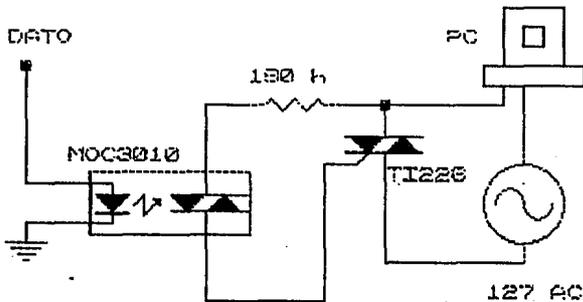


III.2.5 Diagrama de bloques Sección lógica de selección

## SECCIÓN DE POTENCIA

La salida "Q" del flip-flop está conectada a la entrada de un inversor de colector abierto (74LS05), el cual se utiliza para obtener una corriente de 300mA para la señal deseada.

La salida de dicho inversor está conectada al (TRIAC DRIVER) circuito integrado MOC-3010 y este a su vez se encuentra conectado al Triac TIC226, que junto con la toma de corriente de la computadora forman el sistema que va a permitir o no el paso de corriente.



**III.2.7 Diagrama de bloques Sección de Lógica**

## FUENTE DE PODER

El módulo del controlador es externo a la computadora, por lo tanto se fabricó una fuente de poder, de 5 V a 1Amp. Este módulo es portátil con el fin de no ocupar espacio dentro de la computadora. El diagrama de conexiones es la Figura III.2.7.

En el diagrama de conexiones, tenemos un transformador de 127V a 19V a 1Amp, un puente de diodos de 6 Amp, Un regulador de voltaje de 5v (LM7805), un capacitor 200  $\mu$ F y uno de 0.01  $\mu$ F.

El circuito correspondiente a la fuente se armó en una tableta independiente a la tableta del controlador, lo anterior fue debido al diseño del gabinete, el cual es pequeño para ahorrar espacio.

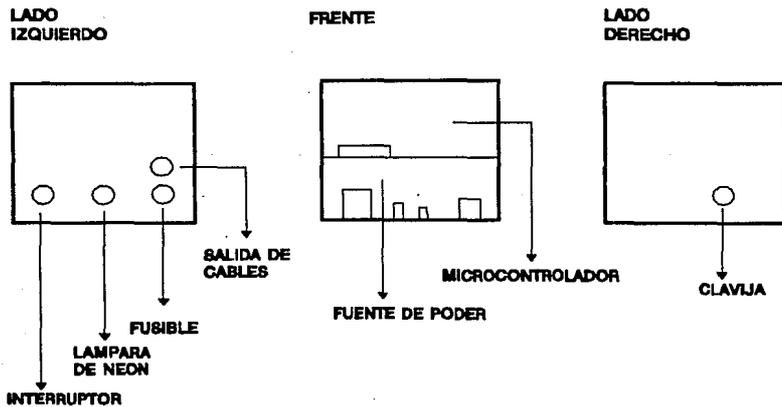
### Gabinete

En conjunto el módulo del controlador se instaló en un gabinete de acrílico. Se utilizó este material por la facilidad que ofrece para unirlo, se utiliza un líquido que disuelve el acrílico permitiendo unir una pieza con otra.

El gabinete tiene las siguientes medidas: largo 11 cm, ancho 9 cm y alto 9 cm., tiene unos pequeños rieles por los que se deslizan dos tabletas, en la parte superior está la tableta correspondiente al microcontrolador, en la parte inferior la correspondiente a la fuente.

A continuación se presenta un diagrama del gabinete con los

diferentes componentes que lo conforman:



### III.3.8 Gabinete

## PROGRAMAS

El módulo del controlador utilizó dos programas:

El primero está en lenguaje "Emsamblador", este programa se encuentra en la EEPROM interna del microcontrolador, se usó el programa PCBUG.EXE para acceder a la memoria, este programa me permitía borrar código anterior o almacenar código nuevo.

Cuando se enciende el circuito, el microprocesador se inicializa automáticamente, esto se hizo utilizando el transistor MC34064, que al detectar voltaje cambia el nivel a cero volts y después lo estabiliza.

Después de que se inicializa el microcontrolador, se ejecutan las siguientes instrucciones:

- inicializa los registros que va a utilizar en el programa,
- configura el protocolo de comunicaciones, esto es:

longitud del byte	= 8 bits
velocidad de transmisión	= 9600 bauds, esto se determina por el cristal que se esté utilizando en nuestro caso es de 8MHz
protocolo de la palabra	= 1 bit de start, 8 bits de datos, 1 bit de stop

- se determina que se va a trabajar sin interrupciones

Después de que están configurados los registros, se inicia un ciclo, esto es:

- Revisa una bandera para determinar si el microcontrolador ha recibido algún dato o no, entra en un ciclo hasta que lea algo.
- Lee el dato del registro que almacena el dato que recibe de la computadora,
- En este momento debemos estar seguros que el dato se recibió bien, para esto, se envía nuevamente el dato a la computadora, este es otro ciclo que revisa otra bandera.
- Después se ejecuta la rutina que escribe al puerto A donde se encuentra conectada la lógica de selección que ya fue descrita anteriormente.

El programa se queda en este ciclo (revisando y escribiendo) hasta que se desconecta el circuito.

El programa tiene la función de revisar constantemente si la computadora central le ha enviado algún dato, de ser así, lee el registro que contiene el dato, el programa interpreta la información y pone en el puerto A (PORT A) la siguiente información (ya se explicó anteriormente el significado de esta información):

PA7	PA6	PA5	PA4	PA3
END	END	C	B	A
D #1	D #2	datos de selección de los decodificadores 3x8		

Habilitación de

## **Capítulo III.2**

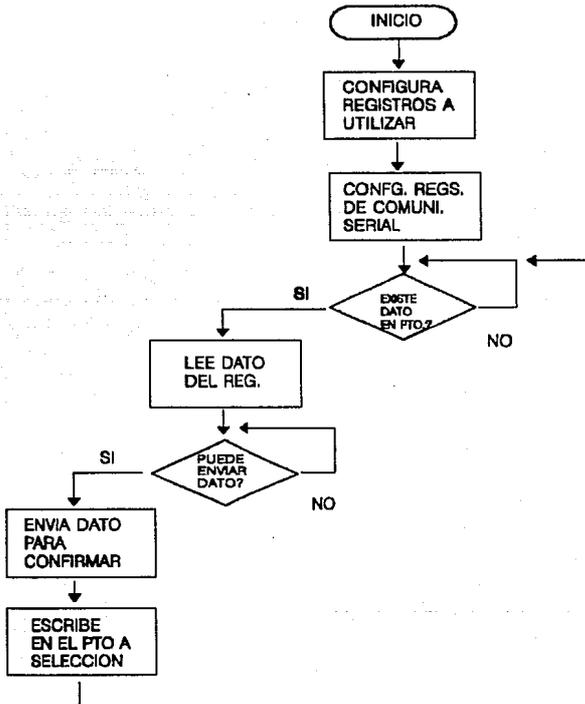
### **los decodificadores**

Las salidas del decodificador son: el reloj (CK) y el dato (D) del flip-flop D.

CK                    D  
para el manejo del flip- flop D

**Para mayor información, revisar el apéndice B, Programa 1.**

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*



**III.2.8 Diagrama de flujo, programa en Ensamblador HC11E2**

El segundo programa está en lenguaje "C", para entablar la comunicación serial, utilizando interrupciones. El programa recibe información de la aplicación en PARADOX y este a su vez envía por el puerto serial esta información al microcontrolador.

El programa hace uso de interrupciones al BIOS (Basic Input Output System). La interrupción que se utilizó es la siguiente, los valores de los registros determinan su función

### OX14 - manejo del puerto serial

#### inicia la comunicacion

dx = 0      en este caso dx determina el puerto al cual se hace referencia, esto es: 0 - puerto serial 1, 1 - puerto serial 2.  
ah = 0  
al = OXE3    significa, baud = 9600, 1 bit de start, 1 bit de stop, logitug del dato 8 bits

#### revisa el puerto

dx = 0  
ah = 3  
al = status

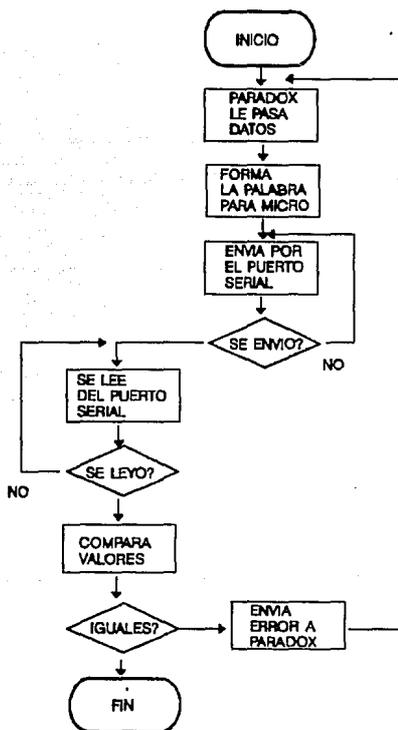
#### lee dato del puerto

dx = 0  
ah = 2  
al = dato leído

#### escribe dato en el puerto

dx = 0  
ah = 1  
al = dato a escribir

Finalmente el programa recibe como argumentos: el número de la máquina a la cual se desea hacer referencia, y estado en que se desea tener, esto es habilitar y deshabilitar.



III.2.9 Diagrama de flujo, programa en lenguaje "C"

El programa se ejecuta desde Paradox, con la instrucción EXECUTE, la cual permite ejecutar un programa fuera del ambiente de Windows, hecho en cualquier lenguaje de programación. Sólo se ejecuta una vez, para cada operación de habilitación o deshabilitación, este programa no se hizo residente, debido a que no convive un programa residente de DOS con Windows.

Por lo anterior se maneja como una librería adicional. En un principio surgió la duda de su compatibilidad, debido a que Windows maneja eventos, y debido a esto tal vez el tiempo asignado al programa de comunicaciones no sería suficiente para que este terminara. Pero se hicieron pruebas y funcionó, se solucionó aumentándole el tiempo asignado para su ejecución.

Para mayor información, hacer referencia al apéndice B, Programa 2.

### **III.3 DESARROLLO DE UTILERIAS DE SOFTWARE PARA EL MANEJO, INTEGRACION E IMPRESION DE CREDENCIALES CON SU RESPECTIVO CODIGO DE BARRAS.**

El desarrollo del sistema fue en Paradox para Windows, elegido principalmente porque tiene integrado el manejo de imágenes. La tarjeta PIB dentro del software tiene una demostración hecha en dBase III plus que utiliza los programas dbpib que permiten la interfase con la tarjeta.

El programa en su totalidad es desplegado en la televisión esclavizando la PC pero no usándola para el desplegado por esta razón se decidió usar otro software que permitirá el uso del sistema aun cuando la televisión no este presente (solo indispensable su presencia cuando se captura la imagen).

Se pretendió prescindir del uso de la televisión pues los diagramas de conexión en los manuales mostraban la opción del no uso de una televisión en el desplegado de imágenes, desafortunadamente se requería de un monitor analógico que ya no se consiguen.

Paradox utiliza para la programación de sus sistemas ObjectPAL (lenguaje orientado a objetos) que permiten crear aplicaciones compiladas a partir de los objetos que tiene definidos que son las herramientas de

programación. Cada herramienta de ObjectPAL tiene características y posibilidades diferentes.

ObjectPAL, al igual que C y Pascal, es un lenguaje compilado (no interpretado). Es decir, el compilador de ObjectPAL traduce el código de ObjectPAL que el programador escribe al código de máquina para que la computadora pueda ejecutar.

En Paradox todo es un objeto, desde los botones y los campos creados mediante las herramientas de la barra rápida, hasta las tablas y los archivos de texto almacenados en disco, o hasta los menús creados con código.

Paradox reconoce dos tipos de objetos: los objetos de diseño, u objetos que se colocan en las fichas; y los objetos de datos, entre los que se incluyen archivos, tipos de datos y estructuras de programación. Según la definición formal, un objeto esta formado por datos y código.

En ObjectPAL, los objetos tienen propiedades (como color, posición, tipo y grosor de línea) y métodos (códigos que definen la forma en que se comporta el objeto): las propiedades son datos y los métodos, código.

El entorno ObjectPAL se basa totalmente en objetos. Los objetos de Paradox tienen código estándar incorporado, por lo que la mayor parte del trabajo de programación ObjectPAL puede considerarse mas una personalización y modificación desde cero. No es necesario controlar todo el programa; basta con controlar ciertos aspectos.

Un lenguaje basado en objetos mejora la programación en el sentido de que proporciona herramientas que reflejan con mayor exactitud la forma en que concebimos el mundo.

Con la programación basada en objetos, es posible reproducir la complejidad de los sistemas del mundo real y desarrollar aplicaciones a gran escala de la forma mas eficaz posible.

En un lenguaje basado en objetos, se deja que el procesador de lenguaje decida el método que debe utilizarse según el tipo de objeto que llama al método: el código de programación subyacente esta escrito para los objetos de forma especifica.

Los objetos de ObjectPAL se organizan por tipos. Los objetos de un tipo determinado tienen las mismas propiedades y los mismos métodos.

La creación de aplicaciones de Paradox consiste en gran medida en colocar objetos en fichas y escribir métodos con ObjectPAL que definan el comportamiento de dichos objetos.

En un programa tradicional no interactivo basado en procedimientos, el código del programa controla lo que ocurre y el momento en que ocurre.

La ejecución suele ser lineal. Un programa de este tipo podría, por ejemplo, mostrar una tabla, realizar un calculo, mostrar el resultado, etc. todo bajo el control del programa.

**ObjectPAL es diferente. Las aplicaciones elaboradas con ObjectPAL otorgan el control al usuario, que actúa sobre los objetos (botones, tablas, menús, etc.) en el orden que desea. Aquí no puede aplicarse el enfoque basado en procedimientos: es preciso adoptar una estrategia basada en objetos.**

**La interacción de un usuario con una aplicación genera sucesos y, como respuesta a estos se ejecutan los métodos anexados a los objetos. Un suceso esta formado por datos y código, por lo que encaja en la definición de objeto.**

**Los datos incluyen el tipo de suceso (por ejemplo, pulsación de un botón del ratón o una tecla), lo que ocurrió y porque ocurrió (si, por ejemplo, se trata de algo que hizo el usuario, o de un suceso generado desde dentro de ObjectPAL). El código es el método usado por ObjectPAL para extraer datos.**

**Los objetos de diseño contienen el código que se ejecuta como respuesta a los sucesos. Los objetos de diseño pueden ser menús, páginas, botones, cuadros, marcos de tabla u otros objetos de las fichas (incluida la propia ficha) sobre los que el usuario actúa.**

**Los gestores de visualización son los contenedores (Un objeto es contenedor de otro si el segundo se encuentra totalmente dentro de los bordes del primero.**

**La situación de un objeto en la jerarquía de contenedores determina si las variables, los métodos y los procedimientos están disponibles para el o no.) de los objetos de diseño. Los gestores de visualización son**

ventanas que muestran los datos, como formas y reportes.

Los tipos de datos básicos permiten declarar variables para almacenar y manipular datos en las tablas, además de realizar cálculos y operaciones sin necesidad de obtener datos de las tablas o mostrar los resultados al usuario.

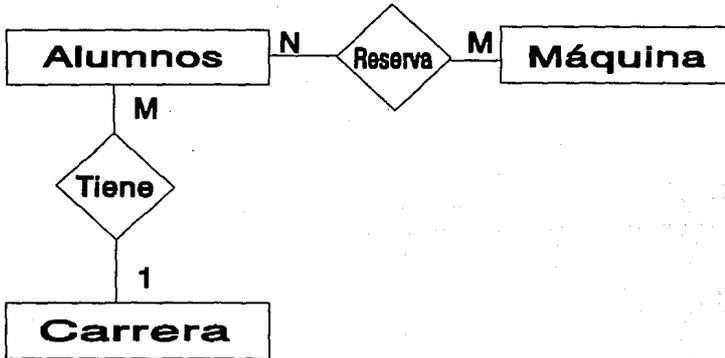
Los objetos de datos del sistema almacenan y acceden a datos acerca de Windows, DOS, contadores del usuario, pero no a los datos almacenados en las tablas.

Cuando se utiliza un aplicación de Paradox, se generan sucesos al actuar sobre los objetos de diseño, contenidos en los gestores de visualización. Los objetos de diseño forman, junto con los gestores de visualización, la interfase de la aplicación.

Los objetos de tipos de datos pueden utilizarse para declarar variables con el fin de realizar cálculos y manipular los datos de las tablas, y los objetos de datos del sistema para proporcionar datos sobre el sistema. ObjectPAL puede trabajar con objetos a cualquier nivel.

## DISEÑO ENTIDAD RELACION.

Por tratarse de un sistema típico de bases se procedió a definir antes que nada el diagrama entidad relación que dando de la siguiente manera:



III.3.1 Diagrama Entidad Relación

Los campos de los que consta cada una de las tablas que componen este sistema son los siguientes:

**ALUMNOS (Entidad)**

<b>NOMBRE</b>	<b>TIPO</b>	<b>TAMAÑO</b>	<b>DESCRIPCION</b>
Cuenta	Alfanumérico	9	Llave principal, Número de cuenta del alumno.
Nombre	Alfanumérico	20	Nombre o nombres del alumno.
Apellidos	Alfanumérico	20	Apellidos del alumno.
Dirección	Alfanumérico	40	Domicilio particular del alumno.
Teléfono	Alfanumérico	10	Teléfono particular del alumno (si tiene).
Cve_carrera	Alfanumérico	2	Clave de la carrera a la cual pertenece.
Fotografía	Gráfico		Fotografía del alumno, (capturada a través de la tarjeta PIB).
Fecha de Inscripción	Fecha		Fecha en la que el alumno se inscribió al laboratorio.
Credencial	Alfanumérico	1	Indica si ya se imprimió la credencial

**CARRERA (Entidad)**

<b>NOMBRE</b>	<b>TIPO</b>	<b>TAMAÑO</b>	<b>DESCRIPCION</b>
Cve_carrera	Alfanumérico	2	Llave principal, Clave de las carreras.
Carrera	Alfanumérico	20	Nombre de la Carrera.

**MAQUINA (Entidad)**

<b>NOMBRE</b>	<b>TIPO</b>	<b>TAMAÑO</b>	<b>DESCRIPCION</b>
Máquina	Alfanumérico	2	Llave principal, identificador de la máquina.
Tipo	Alfanumérico	15	Tipo de máquina (XT, AT etc.)
Servicio	Alfanumérico	1	Indica se la máquina esta en servicio o no.

### RESERVA (RELACION)

NOMBRE	TIPO	TAMAÑO	DESCRIPCION
Cuenta	Alfanumérico	9	Llave foránea, número de cuenta del alumno.
Asistio	Alfanumérico	1	S - si asistio.
Máquina	Alfanumérico	2	Llave foránea, número de máquina.
Fecha	Fecha		Fecha en que el alumno reservo el equipo.
Hora	Alfanumérico	1	Hora en que se reservo el equipo.

Una vez diseñado el sistema se procede a programar el sistema, primero creando las tablas, después diseñando las formas, el sistema consta de 3 formas:

**FORMA DATOS DEL ALUMNO:** Esta es la forma principal del sistema, en ella aparecen los datos del alumno y incluyendo su fotografía, cuenta con 6 botones para realizar diferentes acciones.

**NUEVO:**

Este botón nos permite ingresar a la base de datos un alumno a inscribir, solicita sus datos personales y al momento de llegar al campo de la fotografía se activa el programa que nos permite capturar la imagen (Este programa esta realizado en 'C' y ensamblador e interactúa directamente con la tarjeta). la fecha de inscripción se toma directamente del sistema. En los campos de Cuenta, nombre, apellidos, dirección y clave de carrera no acepta información nula. Una vez ingresada la información Paradox ordena los datos por la llave principal (Número de cuenta).

**BUSCAR:**

Permite hacer una consulta, solicita el número de cuenta del alumno, y lo busca en base de datos de existir la información se despliega en pantalla sino manda un mensaje de error, cabe mencionar que para búsqueda de información Paradox tiene herramientas auxiliares representadas por flechas que permiten movernos de registro en registro.

**CREDECIAL:** Al seleccionar esta opción se mandará a imprimir, las credenciales de los alumnos que aún no hayan sido impresas, se detectan por la información guardada en el campo credencial, una vez impresas se modifica esta información.

**RENOVACION:** Si se da el caso de que un alumno extravíe su credencial, es posible solicitar una reposición, se modificará el campo de credencial para identificar al usuario como que aún no tiene credencial.

**APARTADO:** Un alumno con su credencial podrá reservar tiempo, esta opción manda llamar a la forma apartado de tiempo.

**CONFIRMACION:** Cuando el alumno desee hacer uso de su derecho de uso de equipo previamente reservado, con esta opción se invocará a la forma de confirmación, para proceder a asignar el equipo.

<b>Número de Cuenta:</b>	<input type="text"/>	<b>Nuevo</b>
<b>Nombre:</b>	<input type="text"/>	<b>Buscar</b>
<b>Apellidos:</b>	<input type="text"/>	<b>Credencial</b>
<b>Dirección:</b>	<input type="text"/>	<b>Reposición</b>
<b>Teléfono:</b>	<input type="text"/>	<b>Apartado</b>
<b>Clave Carrera:</b>	<input type="text"/>	<b>Confirmación</b>
<b>Fecha de inscripción:</b>	<input type="text"/>	
<b>Fotografía:</b>	<input type="text"/>	

### III.3.2 Forma Principal -Datos del Alumno.

**FORMA APARTADO:** Esta forma como se acaba se mencionar es invocada desde la forma principal Datos del alumno, solicita el número de cuenta de existir aparecerán su nombre y apellidos, después solicitará la fecha en la cual quiere reservar, esta información esta validada para que únicamente se reserve con una semana de anticipación y de lunes a viernes, los días festivos también están considerados. Una vez ingresada la fecha solicita la hora en que se desea ocupar el equipo, esta debe darse en el formato de 24 horas (ej. 16:00 hrs. en lugar de 4:00 p.m.), si se da el caso de que se reserve tiempo para la fecha actual el sistema

detectara que el alumno no tenía reservado previamente, queda a consideración de la persona que maneje el sistema el asignarle mas tiempo o no según la demanda, pues un alumno solo puede reservar una hora diaria. En esta forma existen 2 botones u acciones.

**ACEPTAR:** Si los datos ingresados están correctos se procede a actualizar la información y se regresa a la forma principal.

**CANCELAR:** Cuando la información es incorrecta o simplemente no se quiere realizar la operación de apartado, esta opción nos devuelve a la forma principal.

Número de Cuenta:

Nombre:

Apellidos:

Fecha:

Hora:

Máquina
01

### III.3.3 Forma de Apartado de Tiempo

**FORMA CONFIRMACION:** Esta forma también es invocada desde la forma principal, su tarea es proceder a asignar el equipo previamente reservado con hasta 45 minutos de retraso, si un alumno quiere ocupar un equipo de una persona que reservo pero no asistió, primero deberá ir a la forma de apartado y luego a la de confirmación, para poder hacer uso. La forma ejecuta una consulta para seleccionar a los alumnos que en ese momento tienen reservación (transparente al usuario), inmediatamente solicita el número de cuenta del alumno de estar en la lista de reservación

se procede a asignarle el equipo, dando paso a la energía eléctrica en la máquina que el tiene reservada. Solo tiene un botón.

**CANCELAR:** Para cancelar la operación y regresar a la forma principal.

## DIAGRAMA DE FLUJO DE DATOS

El seguimiento del sistema es por disparo de métodos, por tratarse de programación orientada a objetos, el código se encuentra repartido en los diferentes objetos, como la forma principal es la forma de captura de datos del alumno de ahí partiremos para hacer el seguimiento del programa, al final se pondrá el árbol de objetos de cada forma.

El código que forma cada método se anexará al final en el apéndice A identificándolos por un número de programa, cuando se refiera a alguno lo haremos a través de su número de programa.

Número de Cuenta:

Nombre:

Apellidos:

Máquina:

### III.3.4 Forma de Confirmación de Apartado de Tiempo.

Al abrir la forma principal se dispara el método open de la forma, en este método se tienen definidos los estándares del sistema, programa 1.1:

- Formato de Fecha y hora.
- Definición de atributos de la forma.

Una vez definidos se presenta la forma -figura III.3.2- como no se ha definido alguna operación en la forma se vacía la información del primer usuario en la base de datos. En este momento se puede realizar cualquiera de las 6 acciones definidas por los botones.

- 1. Nuevo:** Este botón como todo lo que existe en la forma es un objeto, tiene definido el método pushbutton, programa 1.2, que se dispara en el momento en que el usuario hace click con el mouse y automáticamente se limpia la forma para permitir el ingreso de datos.

En primer lugar se posiciona en objeto número de cuenta de alumno, este tiene definido el método action, programa 1.3, que contiene la rutina del dígito verificador, no permite avanzar si el número de cuenta esta mal.

Una vez dado correctamente el número de cuenta nos permitirá capturar el resto de los datos personales del alumno (todo se captura en mayúsculas para dar uniformidad a la información), el nombre el apellido y la dirección no pueden estar vacíos por lo que no se avanzará a la siguiente información si éstos tienen información nula.

El teléfono puede no tener información. En el caso de la

clave de la carrera, por default tiene el valor de 32 (clave de la carrera de Ingeniero en Computación), pero únicamente se pueden capturar las claves de 21 al 32, y del 71 al 73. La correspondencia de claves es de la siguiente manera:

<b>Clave</b>	<b>Carrera</b>
21	Civil
22	Minas y Metalurgia
23	Geología
24	Petróleo
25	Topógrafo y geodesta
26	Mecánico electricista
27	Area mecánica
28	Area industrial
29	Area electrónica
31	Geofísico
32	Computación
71	Especialidad
72	Maestría
73	Doctorado

La siguiente información es la captura de la fotografía del alumno, programa 1.4, en este momento se invoca al programa que maneja la tarjeta, este programa esta formado por rutinas hechas en Lenguaje 'C' y en ensamblador programa 1, apéndice B. Una vez capturada la imagen se incorpora a los datos del alumno.

El último dato es la fecha de inscripción que toma directamente del sistema, programa 1.5. Con esto el registro es incorporado automáticamente a la tabla de alumnos y el sistema vuelve al punto inicial, donde la forma tiene los datos del primer registro, esperando nuevamente una acción.

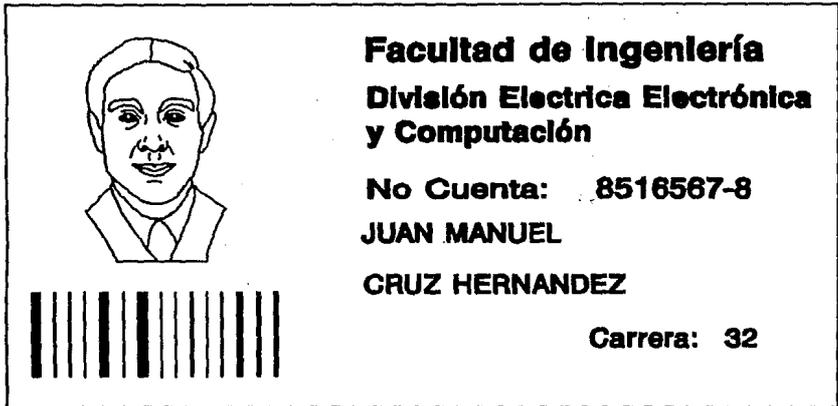
**2. Buscar:**

En este objeto como en todos los botones se tiene definido el método pushbutton, programa 1.6, con un click con el mouse en este botón aparecerá una ventana pidiendo introducir el número de cuenta del alumno a buscar, de existir en la forma se vaciará con los datos correspondientes, sino se manda un mensaje de error.

**3. Credencial:**

Quando se dispara el método pushbutton de este objeto se mandarán a imprimir todas aquellas credenciales de los alumnos que en ese momento todavía no las tengan o aquellos que hayan pedido reposición de la misma, programa 1.7. El diseño es

Diseño y Construcción de un sistema de captura de imágenes y control de asignación de equipo  
el siguiente:



### III.3.5 Formato de Credencial

#### 4. Reposición:

El método pushbutton de este objeto permite modificar la información de la tabla, solicita el número de cuenta del alumno, de existir modifica la tabla para que cuando se solicite la acción de Credencial se incluya en la impresión de las mismas, programa 1.8.

**5. Apartado:** Este botón permite invocar a la forma de apartado de tiempo, figura III.3.3, programa 1.9, donde automáticamente pide el número de cuenta del alumno, el objeto cuenta con el método action que permite comprobar el número dado con el algoritmo del dígito verificador programa 2.1, y si este es correcto se procede a hacer la búsqueda en la tabla, de existir el nombre y los apellidos aparecerán en la forma.

Después pedirá la fecha para la cual el alumno desea ocupar el equipo, este objeto (Fecha) tiene definido los métodos, arrive que inicia modo edición, CanDepart que no permite salir de el si la fecha es incorrecta y error para fecha incorrectas programas 2.2, pues solo se puede apartar con una semana de anticipación de lunes a viernes y contempla días festivos, (5 de febrero, 21 de marzo, 1 de mayo, 5 de mayo, 16 de septiembre, 12 de octubre, 1 de noviembre, 20 de noviembre y 12 de diciembre).

De aceptarse el dato de la fecha, se procede a solicitar la hora, también tiene definido el método CanDepart para permanecer en el objeto hasta no teclear un dato correcto, programa 2.3, el horario de 7 a 22, (la hora se debe introducir en el formato de 24 horas).

Para seleccionar la máquina existe una pequeña tabla con barra de desplazamiento vertical que nos permite localizar la máquina deseada con la fechas o el mouse

haciendo click en dicha barra.

Esta forma tiene definidos botones, Aceptar y Cancelar.

**Aceptar:** Permite registrar la información, siempre y cuando estén todos los datos correctos, y regresa a la forma principal, programa 2.4.

**Cancelar:** Se regresa a la forma principal ignorando si hubo o no información, se cancela la operación de apartado, programa 2.5.

**6. Confirmación:** Invoca a la forma de confirmación de tiempo, programa 1.10. Esta forma tiene definido el método de arrive, programa 3.1, que se dispara al entrar, en el se ejecuta una consulta sobre la tabla de reserva, para obtener un subconjunto de alumnos, aquellos que tienen apartado para la fecha y hora actual.

Paradox crea una tabla de resultados cuando se ejecuta una consulta, en el sistema esta tabla se llama Consulta. El método que utiliza Paradox para realizar consultas es a través del sistema de Query By Example (QBE).

La forma solicita el número de cuenta, este objeto como los anteriores objetos donde se pide número de cuenta del alumno tiene validada la rutina del dígito verificador para los números de cuenta de la U.N.A.M., se hace la búsqueda del número de cuenta introducido en el resultado de la consulta, programa 3.2.

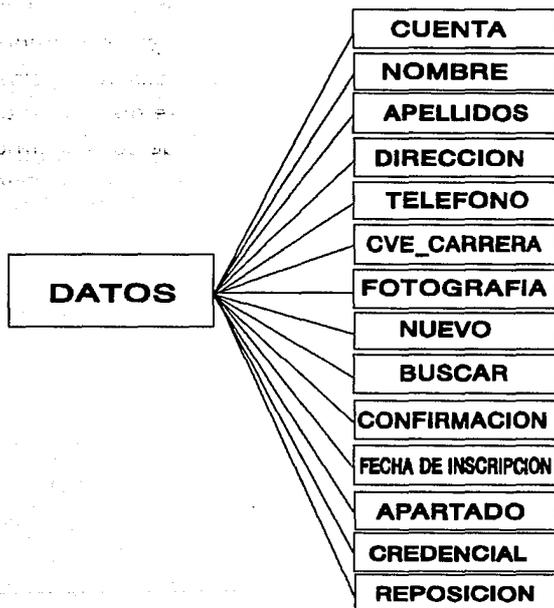
De tener tiempo reservado (hasta con 45 minutos de retraso) se procede a desplegar su nombre, apellidos y la máquina que tiene reservada, para inmediatamente invocar a la rutina que permite comunicarse con el microprocesador para que este permita el paso de corriente al equipo señalado. Después se regresa a la forma principal.

Se diseñó un programa residente que dará aviso al usuario 5 minutos antes de que su tiempo se acabe programa xx.

Esta forma tiene un botón auxiliar.

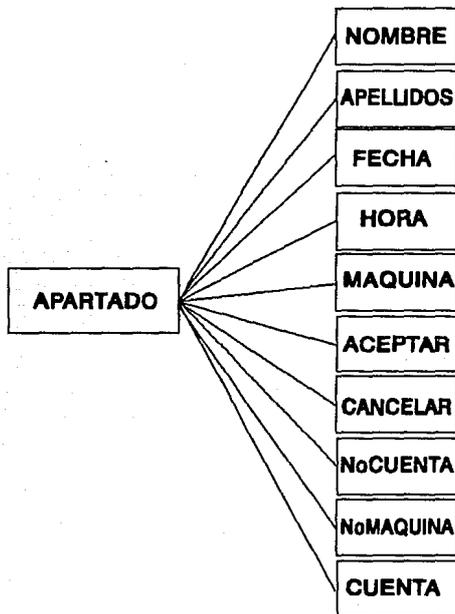
**Cancelar:** Este botón permite cancelar la operación de confirmación y regresar a la forma principal, programa 3.3.

A continuación se incluye el árbol de objetos de cada una de las formas:



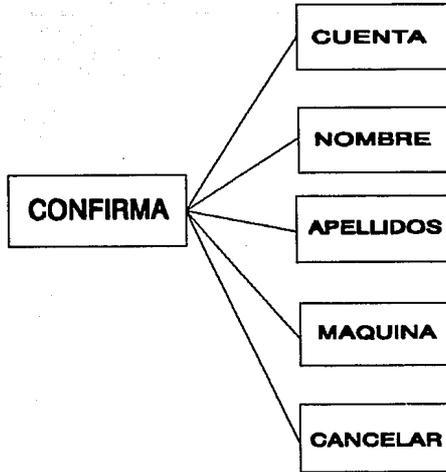
### III.3.6 Arbol de Objetos de la Forma de Captura de Datos

En este árbol de objetos observamos que existe el objeto NoCuenta y Cuenta, en ambos casos se refiere al número de cuenta del alumno, pero el primero objeto esta asociado a la tabla de Reserva mientras que el segundo solo nos permitirá obtener el dato para ir a buscarlo posteriormente a la tabla de alumnos, ambos están presentes en la forma pero solo es visible Cuenta, mientras que NoCuenta esta oculto, no lo visualizamos pero es necesaria su presencia pues el nos permite guardar la información de apartado de tiempo (guardar la información en la tabla Reserva).



### III.3.7 Arbol de Objetos de la Forma de Apartado de Tiempo

De la misma manera observamos el objeto NoMáquina y Máquina, ambos asociados con el número de equipo, el primero corresponde a la tabla de Reserva (oculto), mientras que el segundo forma parte de la tabla auxiliar con desplazamiento vertical -figura III.3.3- que nos permite seleccionar el equipo a ocupar.



**III.3.8 Arbol de Objetos de la Forma de Confirmación**

# Conclusiones

## **CONCLUSIONES**

En las instituciones y centros educativos es de primordial importancia contar con recursos humanos y técnicos, que se encuentren a la vanguardia de los avances tecnológicos. Como ya es sabido, los cambios tecnológicos se dan a pasos agigantados, y es por lo anterior que es de vital importancia actualizarse.

Pero también, encontramos el caso contrario, en el que no existen metas, ni el espíritu de superación entre las personas que conforman los recursos humanos, y tampoco se explotan los recursos técnicos, lo que trae como consecuencia el atraso.

La idea que originó la realización de este proyecto fue el darle una aplicación a la tarjeta PIB adquirida por el departamento de Computación para la materia de procesamiento digital de imágenes, pues su única función era la realización de prácticas de dicha materia, con este trabajo se apoyó a la materia mostrando una aplicación práctica, y como este existen en potencia muchos otros proyectos de gran beneficio para la facultad. Cabe aclarar que antes de iniciar las pruebas con la tarjeta, su uso era inadecuado debido a que estaba mal configurada y el software de aplicación estaba mal instalado.

Sobre el tema de digitalización de imágenes existe mucha información, dispersa en varios documentos (libros, revistas, manuales etc.), en este trabajo se hizo un compendio de toda la información que

### *Conclusiones*

recabamos al respecto y que tenía que ver con la aplicación que se estaba realizando, no solo por tener un trabajo completo sino por respaldar con bases teóricas y de esa manera se comprenda mejor el funcionamiento del sistema.

Personalmente este trabajo nos trajo muchos beneficios, pues nos permitió conocer mas afondo dos de los temas de mas actualidad, la digitalización de imágenes, que como se sabe es una de las partes principales en la tecnología Multimedia; y el Control Digital a través del microprocesador HC11E2, que nos permite manejar en conjunto los conocimientos de Hardware y Software, que es una de las principales características que distingue al Ingeniero en Computación de las demás carreras afines. Entonces aparte de conocer ambos temas logramos integrarlos en un sistema, imponiéndonos retos y obteniendo beneficios.

En cuanto a la rentabilidad del sistema hicimos un análisis de costos obteniendo lo siguiente:

**Nuevos Pesos**

<b>Computadora Central (*)</b>	<b>7,000.00</b>
<b>386DX25</b>	
<b>8K RAM</b>	
<b>DD 200Mb</b>	
<b>Tarjeta digitalizadora de video (*)</b>	
<b>PIB adquirida en 1988</b>	<b>1,500.00</b>
<b>Cámara de video (*)</b>	
con funciones de edición	<b>3,000.00</b>
<b>Television (entrada de video)</b>	<b>1,200.00</b>
<b>Lector óptico</b>	<b>1,400.00</b>
<b>Impresora Láserjet IIP (*)</b>	<b>3,000.00</b>
<b>Font (Código de barras)</b>	<b>400.00</b>
<b>Controlador (Módulo)</b>	<b>300.00</b>
<b>Paradox (Manejador de B.D.)</b>	<b>600.00</b>
<b>Etiquetas (800 unidades)</b>	<b>120.00</b>
<b>Cables y conectores</b>	<b>50.00</b>

-----  
**18,570.00**

El precio no resulta ser muy elevado, pero se debe tomar en cuenta que parte del equipo se adquirió previamente (\*), para el apoyo de diferentes materias. Este mismo sistema puede reconstruirse con equipo más económico, esto es:

*Conclusiones*

Tarjetas Digitalizadoras que incluyen además sonido, se consiguen hoy en día más económicas aproximadamente	900.00
Cámara de video blanco y negro sin funciones adicionales	1,000.00
Impresora de matriz.	600.00
Televisión blanco y negro con convertidor a entrada de video	350.00
	-----
	2,850.00

Estos artículos suman 8,700.00 con el equipo actual, haciendo una diferencia de 5,850.00, 67.25% menos.

Pero la ventaja principal es la automatización del trabajo, el tiempo que antes se tardaban en hacer credenciales se reduce considerablemente pues prácticamente es instantánea la generación de esta, además que se controla el acceso al uso del equipo, detectando usuarios que hacen mal uso del laboratorio.

# **Apéndice**

## **A**

### **Programas Paradox**

## APENDICE A

### Programa 1.1

method open(var eventInfo Event)

```
*****  
; METODO OPEN FORMA DATOS DEL ALUMNO  
; CONFIGURACION DEL AMBIENTE  
; PROGRAMA 1.1  
; DIANA ANABEL HERNADEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****  
  
formatadd("horitas","TOW")           ;Se define el formato  
formatSetTimeDefault("horitas")      ;de la hora en 24 horas  
maximize()                            ;Se maximiza el tamaño de  
                                       ;la ventana  
  
endmethod
```

### Programa 1.2

method pushButton(var eventInfo Event)

```
*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON NUEVO  
; SE DISPONE A CAPTURAR DATOS  
; PROGRAMA 1.2  
; DIANA ANABEL HERNADEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****  
  
action(DataBeginEdit)                 ; Empezar a editar  
action(datainsertrecord)              ; Insertar Informaci≤n  
  
endmethod
```

### Programa 1.3

```
method action(var eventInfo ActionEvent)
```

```
*****  
; METODO ACTION FORMA DATOS DEL ALUMNO, OBJETO: NUMERO DE CUENTA  
; CAPTURA Y VERIFICA EL NÚMERO DE CUENTA DEL ALUMNO  
; DIGITO VERIFICADOR Y QUE EL ALUMNO NO ESTE REGISTRADO  
; PROGRAMA 1.3  
; DIANA ANABEL HERNADEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****
```

```
var
```

```
; *** Variables auxiliares en el calculo  
; *** en el calculo del digito veficador
```

```
sump,           ; suma de digitos pares  
sumn,           ; suma de digitos nones  
i,              ; indice para recorrer el arreglo  
dig number     ; digito verificador calculador  
ncta,          ; n·mero de cuenta  
cadena string  ; dígito convertido a cadana para comparaci≤n
```

```
endvar
```

```
if eventInfo.id() = FieldForward then ;Si quiere abandonar el campo  
;toma el valor a verificar
```

```
sump = 0  
inicializan variables  
sumn = 0  
ncta = Cuenta.value
```

```
for i from 1 to 7 step 2 ;Suma de digitos nones  
sumn = numval(substr(ncta,i,1)) + sumn  
endfor
```

```
for i from 2 to 6 step 2 ;Suma de digitos pares  
sump = numval(substr(ncta,i,1)) + sump  
endfor
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
dig = sump*3 + sumn*7           ;Dígito verificador
cadena = strval(dig)           ;Se convierte a cadena
; *** Si no son iguales los dígitos mensaje de error
; *** y regresa a solicitarlo nuevamente

if (substr(cadena,cadena.size()-3,1) <> substr(ncta,9,1)) then
    beep()
    msgInfo(" E R R O R !!", "N·mero de cuenta incorrecto")
    disabledefault
    Cuenta.moveto()
    Cuenta = space(9)
endif

; *** Si el n·mero de cuenta ya existe mensaje de error
; *** y regresa a solicitarlo nuevamente

if Cuenta.locate("Cuenta",ncta) then
    beep()
    msgInfo(" E R R O R !!", "El alumno ya esta registrado")
    disabledefault
    Cuenta.moveto()
    Cuenta = space(9)
endif

endif
endmethod
```

## Programa 1.4

```
method action(var eventInfo ActionEvent)
;*****
; METODO ACTION FORMA DATOS DEL ALUMNO, OBJETO: FECHA DE
; INSCRIPCION
; ASIGNA LA FECHA DEL SISTEMA A LOS DATOS
; PROGRAMA 1.4
; DIANA ANABEL HERNADEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
;*****
if eventInfo.id() = FieldForward then
    fecha_de_inscripcion.value = date()           ;Asigna fecha del sistema
    disabledefault                               ;Terminamos de insertar datos
    action(DataEndEdit)
endif
endmethod
```

## **Programa 1.5**

```
method action(var eventInfo ActionEvent)
```

```
.....  
; METODO ACTION FORMA DATOS DEL ALUMNO, OBJETO: FECHA DE  
; INSCRIPCION  
; ASIGNA LA FECHA DEL SISTEMA A LOS DATOS  
; PROGRAMA 1.5  
; DIANA ANABEL HERNADEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
.....
```

```
var  
  TCCarrera TCursor  
  carrera string  
endVar  
carrera = self.value  
TCCarrera.open("Carrera.db")  
if eventInfo.id() = FieldForward then  
  if not TCCarrera.locate("Cve_Carrera",carrera) then  
    MsgStop("¡ERROR! ", "Clave de Carrera no valida")  
    disabledefault  
    self.value = blank()  
    cve_carrera.moveto()  
  else  
    disabledefault  
    fecha_de_inscripcion.value = date() ;Asigna fecha del sistema  
    credencial1.value = "N" ;Terminamos de insertar datos  
    action(DataEndEdit)  
  endif  
endif  
endmethod  
method action(var eventInfo ActionEvent)
```

## Programa 1.6

method pushButton(var eventInfo Event)

```
*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON BUSCAR  
; LOCALIZA LOS DATOS DE UN ALUMNO A TRAVES DE SU NUMERO DE CUENTA  
; PROGRAMA 1.6  
; DIANA ANABEL HERNADEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****
```

```
var  
  numcuenta,          ;N·mero de Cuenta del alumno  
  vacia string        ;Cadena vacia  
endVar
```

```
numcuenta = space(9)   ;Se inicializan variables  
vacía = space(9)
```

```
numcuenta.view("Introduzca el n·mero de cuenta") ;Pide numero de cuenta  
if numcuenta < > vacia then ;Si no es cadena vacia  
  if not cuenta.locate("Cuenta",numcuenta) then ;Si no encuentra el dato  
    beep()  
    ;mensaje de error  
    message(" No ha sido localizado: ",numcuenta)  
    sleep(1000)  
  endif  
endif  
endmethod
```

## **Programa 1.7**

```
method pushButton(var eventInfo Event)
```

```
;*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON: CREDENCIAL  
; SELECCION DE ALUMNOS SIN CREDENCIAL PARA IMPRESION  
; PROGRAMA 1.7  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
;*****
```

```
var  
Credenciales Report  
DatosCredencial ReportPrintInfo  
Nuevas String  
TCAlumnos TCursor  
opcion String  
EndVar  
TCAlumnos.open("alumnos.db")  
opcion = "N"  
  
Nuevas = "Query\n\n" +  
"alumnos|Cuenta|Nombre|Apellidos|Cve_carrera|Fotografia|Credencial|\n" +  
" |check |check |check |check |N |\n\n" +  
"EndQuery"  
if executeQBESString(Nuevas) then  
Credenciales.open("Credenc.rsl")  
DatosCredencial.nCopies = 1  
DatosCredencial.name = "Credenc"  
DatosCredencial.queryString = Nuevas  
if Credenciales.print(DatosCredencial) then  
if TCAlumnos.locate("Credencial",opcion) then  
TCAlumnos.edit()  
TCAlumnos.credencial = "S"  
while TCAlumnos.locateNext("Credencial",opcion)  
TCAlumnos.credencial = "S"  
endwhile  
TCAlumnos.endEdit()  
endif  
else
```

*Apéndice A*

```
MsgStop("error", "no se puede ejecutar consulta")
endif
else
MsgStop("Error", "no se puede Imprimir")
endif
endmethod
```

## **Programa 1.8**

```
method pushButton(var eventInfo Event)
```

```
*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON BUSCAR  
; LOCALIZA LOS DATOS DE UN ALUMNO A TRAVES DE SU NUMERO DE CUENTA  
; Y MODIFICA LA TABLA PARA QUE SE GENERE UNA NUEVA CREDENCIAL  
; PROGRAMA 1.8  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****
```

```
var  
    numcuenta,          ;N·mero de Cuenta del alumno  
    vacia string       ;Cadena vacia  
endVar
```

```
numcuenta = space(9)   ;Se inicializan variables  
vacía = space(9)
```

```
numcuenta.view("Introduzca el n·mero de cuenta") ;Pide numero de cuenta  
if numcuenta <> vacia then ;Si no es cadena vacia  
    if not cuenta.locate("Cuenta",numcuenta) then ;Si no encuentra el dato  
        beep()  
        ;mensaje de error  
        message(" No ha sido localizado: ",numcuenta)  
        sleep(1000)  
    else  
        action(DataBeginEdit)  
        Credencial1.value = "N"  
        action(DataEndEdit)  
        MsgInfo("Credencial", "Dato modificado proximo a imprimir ....")  
    endif  
endif  
endmethod
```

## Programa 1.9

method pushButton(var eventInfo Event)

```
;*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON APARTADO  
; INVOCA A LA FORMA DE APARTADO DE TIEMPO  
; PROGRAMA 1.9  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
;*****
```

var

```
    nombre,          ; Nombre logico de la tabla alumnos  
    fecha table     ; Nombre logico de la tabla reserva  
    reservacion form ; Forma de apartado  
endvar
```

```
nombre.attach("Alumnos.db") ;Se asocia nombre con alumnos  
fecha.attach("reserva.db") ;Se asocia fecha con reserva
```

```
if reservacion.open("aparta") then ; Si se puede abrir la forma  
    dlgVal = reservacion.wait()
```

```
else  
    msgInfo("Problema", "El cuadro no puede abrirse") ; De otra manera mensaje  
                                                ; de error.
```

endif

endmethod

## **Programa 1.10**

```
method pushButton(var eventInfo Event)
```

```
;  
;*****  
; METODO PUSHBUTTON FORMA DATOS DEL ALUMNO, BOTON CONFIRMACION  
; INVOCA A LA FORMA DE CONFIRMACION DE TIEMPO APARTADO  
; PROGRAMA 1.10  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
;*****
```

```
var  
  confirmacion form ;Forma de confirmacion  
endvar
```

```
if confirmacion.open("confirma") then ;Si se puede abrir la forma  
  dlgVal = confirmacion.wait()  
else  
  msgInfo("Problema", "El cuadro no puede abrirse") ;De otra manera mensaje  
  ;de error.  
endif
```

## Programa 2.1

```
method canDepart(var eventInfo MoveEvent)
```

```
*****  
; METODO CANDEPART FORMA RESERVACION DE TIEMPO, OBJETO: CUENTA  
; (numero de cuenta)  
; CHECA NUMERO DE CUENTA CORRECTO Y SI EXISTE EN LA BASE DE DATOS  
; PROGRAMA 2.1  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
*****
```

```
var  
    sump,           ;Suma de digitos pares  
    sumn,          ;Suma de digitos nones  
    i,             ;Indice sobre el arreglo  
    dig number     ;Digito verificador calculado  
    ncta,          ;Numero de cuenta  
    cadena string  ;Digito verificador convertido en cadena  
    TCAlumno TCursor ;Apuntador a registro, se asocia a la tabla de alumnos  
endvar
```

```
sump = 0           ;Se inicializan variables  
sumn = 0  
ncta = Cuenta.value  
TCAlumno.open("alumnos.db")
```

```
if eventInfo.reason() = UserMove then ;Si se pretende abandonar el campo  
if ncta.size() <> 9 then ;Si el tama±o es incorrecto  
msgStop("ERROR","N·mero de Cuenta Incorrecto");Mensaje de error  
Cuenta = blank() ;Se limpia el campo  
Cuenta.moveTo() ;Se regresa al campo  
else ;Tama±o correcto
```

```
; *** Detecta que sean caracteres validos
```

```
for i from 1 to 9  
if NOT (substr(ncta,i,1) >= "0" AND substr(ncta,i,1) <= "9");AND i <> 8 then  
msgStop("ERROR","N·mero de Cuenta Incorrecto")  
Cuenta = blank()
```

```
Cuenta.moveTo()
quitloop
endif
endifor

; *** Suma de digitos nones
for i from 1 to 7 step 2
  sumn = numval(substr(ncta,i,1)) + sumn
endifor

; *** Suma de digitos pares
for i from 2 to 6 step 2
  sump = numval(substr(ncta,i,1)) + sump
endifor

dig = sump*3 + sumn*7 ;Calculo del digito verificador
cadena = strval(dig)

; *** Detecta que este correcto el numero
; *** Si esta mal regresa pedirlo nuevamente
if (substr(cadena,cadena.size()-3,1) <> substr(ncta,9,1)) then
  beep()
  msgInfo(" E R R O R !!", "N-mero de cuenta incorrecto")
  disabledefault
  Cuenta.moveTo()
  Cuenta = space(9)
endif

; *** Localiza el dato en la tabla de alumnos
; *** Si no lo encuentra mensaje de error y vuelve a pedir el dato
if not TCAlumno.locate("Cuenta",ncta) then
  beep()
  msgInfo(" E R R O R !!", "El alumno no esta registrado")
  disabledefault
  Cuenta.moveTo()
  Cuenta = space(9)

; *** Si lo encuentra despliega nombre y apellido
else
  Nombre.value = TCAlumno.nombre
  Apellidos.value = TCAlumno.Apellidos
  action(DataEndEdit)
```

Apéndice A

```
Fecha.MoveTo() ;Se mueve al campo fecha, siguiente dato a pedir
endif
endif
endif
endmethod
```

## Programa 2.2

```
method arrive(var eventInfo MoveEvent)
```

```
*****
; METODO ARRIVE FORMA RESERVACION DE TIEMPO, OBJETO: FECHA
; CHECA NUMERO DE CUENTA CORRECTO Y SI EXISTE EN LA BASE DE DATOS
; PROGRAMA 2.2
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
*****
```

```
action(DataBeginEdit) ;Para editar e insertar dato
action(DataInsertRecord)
```

```
endmethod
```

```
method canDepart(var eventInfo MoveEvent)
```

```
*****
; METODO CANDEPART FORMA RESERVACION DE TIEMPO, OBJETO: FECHA
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
*****
```

```
var
dia, ;Numero de dia
mes smallint ;Numero de mes
endvar
```

```
if eventInfo.reason() = UserMove then ;Si se quiere abandonar el campo
if self.value < today() then ;No puede ser antes de la fecha
;en curso
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado!", "La fecha no puede ser antes del día de hoy ... \n
Hoy es" + strval(today()))
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

dia = dowOrd(self.value) ;Se obtiene numero de dia y mes
mes = month(self.value)

if self.value > today() + 7 then ;Solo con una semana de anticipacion
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Solo con una semana de anticipaci≤n")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

if dia = 1 OR dia = 7 then ;No puede ser Sabado o Domingo
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Día no Laborable \n Fin de Semana")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** Dias festivos
; *** Primero de enero

if dia = 1 AND mes = 1 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Día no Laborable \n Primer día del año")
disableDefault
fecha.moveTo()
self.value = blank()
```

## Apéndice A

```
    action(DataBeginEdit)
endif

; *** 5 de febrero
if dia = 5 AND mes = 2 then
eventInfo.SetErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Constituci≤ n Mexicana")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 21 de marzo
if dia = 21 AND mes = 3 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Natalicio de Benito Juárez")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** Primero de mayo
if dia = 1 AND mes = 5 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Døa del Trabajo")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 5 de mayo
if dia = 5 AND mes = 5 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Batalla de Puebla")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif
```

```
; *** 10 de mayo
if dia = 10 AND mes = 5 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Døa de las Madres")
.disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 16 de septiembre
if dia = 16 AND mes = 9 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Døa de la Independencia")
.disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 12 de octubre
if dia = 12 AND mes = 10 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Døa de la Raza")
.disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** Primero de noviembre
if dia = 1 AND mes = 11 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Informe Presidencial")
.disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 2 de noviembre
if dia = 2 AND mes = 11 then
```

```
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Døa de Muertos")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif

; *** 20 de noviembre
if dia = 20 AND mes = 11 then
eventInfo.setErrorCode(CanNotDepart)
msgStop(" ¡Cuidado! ", "Døa no Laborable\n Revoluci≤ n Mexicana")
disableDefault
fecha.moveTo()
self.value = blank()
action(DataBeginEdit)
endif
endif
endmethod

method error(var eventInfo ErrorEvent)
;*****
; METODO ERROR FORMA RESERVACION DE TIEMPO, OBJETO: FECHA
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
;*****

; *** Si ocurre un error critico (fechas inexistentes)

if eventInfo.reason() = ErrorCritical then
msglnfo("Error Critico",errorMessage())
disableDefault
fecha.value = blank()
fecha.moveto()
endif
endmethod
```

## Programa 2.3

```
method canDepart(var eventInfo MoveEvent)
;*****
; METODO ERROR FORMA RESERVACION DE TIEMPO, OBJETO: FECHA
; CHECA QUE LA HORA SEA CORRECTA
; PROGRAMA 2.3
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
;*****

var
  hora string      ; Hora del sistema
  tiempo string    ; Hora en que solicita el servicio
endvar

hora = StrVal(time())
tiempo = substr(self.value,1,2)
if eventInfo.reason() = UserMove then ;Si quiere abandonar el campo
if fecha.value = today() then ;Servicio para el día en curso?
  if tiempo = substr(hora,1,2) then ;Servicio en la hora en curso?
    tiempo = substr(hora,4,2)
    if tiempo > "45" then ;Solo con 45 minutos de retraso
      eventInfo.setErrorCode(CanNotDepart)
      msgStop(" ¡Cuidado! ", "La hora esta mal son: " + hora)
      disabledefault
      self.value = blank()
      Hora.moveTo()
      action(DataBeginEdit)
    endif

; *** De no se la hora en curso
; *** Verifica que no sea una hora menor a la actual
else
  if self.value < substr(hora,1,5) then
    eventInfo.setErrorCode(CanNotDepart)
    msgStop(" ¡Cuidado! ", "La hora esta mal son: " + hora)
    disabledefault
    self.value = blank()
    Hora.moveTo()
    action(DataBeginEdit)
```

*Apéndice A*

```
    endif  
  endif  
endif  
endmethod
```

## Programa 2.4

```
method pushButton(var eventInfo Event)
;*****
; METODO PUSHBUTTON FORMA RESERVACION DE TIEMPO, BOTON ACEPTAR
; VERIFICA LA INFORMACION, PARA PROCEDER A HACER EL APARTADO
; PROGRAMA 2.4
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
;*****

var
equipo TCursor                ;Apuntador a registro de la tabla maquina
sirve,                        ;Esta en servicio el equipo
elige string                  ;Selección de opciones
ocupa TCursor                ;Apuntador a registro de la tabla reserva
tablaindex Table             ;Tabla indexada de reservacion
endvar

equipo.open("maquina.db")    ;Asignacion de tablas
ocupa.open("reserva.db")
tablaindex.attach("reserva.db")

index tablaindex            ;Creacion del indice
primary
on "cuenta", "maquina", "fecha", "hora"
to "busca"
endindex

action(DataBeginEdit)

if fecha.isblank() then      ;Si no hay fecha Error
msgStop("f Error !", "Falta Fecha")
fecha.moveto()
else
if hora.isblank() then      ;Si no hay hora Error
msgStop("f Error !", "Falta Hora")
hora.moveto()
else
equipo.locate("maquina", maquina.value)
sirve = equipo."servicio"
```

Apéndice A

```
if sirve = "N" then ;Si el equipo seleccionado no funciona error
  msgStop("Problema", "El equipo no funciona")
else
  if ocupa.locate("cuenta", cuenta.value) then ; BUSCAR EL ALUMNO EN RESERVAR
  if ocupa.maquina = maquina.value then ; BUSCAR SI YA TIENE ESA MAQUINA
  if ocupa.fecha = fecha.value then ; BUSCAR EN QUE FECHA
  if ocupa.hora = hora.value then ; BUSCAR A QUE HORA
  elige = msgQuestion(" Maquina ya asignada",
    nombre.value + " " + apellidos.value + "\n\n" +
    "Reservo previamente ¿ Quiere cancelar la reservaci≤n ?")
  fecha.moveto()
    action(DataDeleteRecord)
  if elige = "Yes" then ;Para cancelar reservacion
    ocupa.edit()
      ocupa.deleteRecord()
    ocupa.endEdit()
  action(DataEndEdit)
  cuenta.moveto()
  formreturn("Cancelar")
  close()
  endif
else
  if fecha.value = today() then ; Advertencia si el alumno ya trabajo ese
dia
  elige = msgQuestion("¡ Cuidado !",
    "El alumno ya tiene reservado para hoy...\n
  ¿ Se continua con la reservaci≤n ?")
  if elige = "Yes" then
    NoCuenta.value = cuenta.value
    NoMaquina.value = maquina.value
  else
  fecha.moveto()
    action(DataBeginEdit)
    action(DataDeleteRecord)
    action(DataEndEdit)
    formreturn("Cancelar")

  close()
  endif
; *** Solo una hora por dia
else
  msgStop("¡f Error !", "El usuario ya tiene maquina asignada")
```

```
fecha.moveto()
action(DataBeginEdit)
action(DataDeleteRecord)
action(DataEndEdit)
formreturn("Error")
close()
endif
endif
else

; *** Verificacion de que el equipo no este asignado
if ocupa.locate("maquina",maquina.value) then
if ocupa.fecha = fecha.value then
if ocupa.hora = hora.value then
msgStop("f ERROR !", "Lo siento el equipo ya esta reservado")
fecha.moveto()
action(DataBeginEdit)
action(DataDeleteRecord)
action(DataEndEdit)
formreturn("Error")
close()
endif
endif
endif
endif
else

; *** Tiene reservado el alumno en esa fecha y hora en otra maquina
if fecha.value = ocupa.fecha then
if hora.value = ocupa.hora then
msgStop(" f ERROR ! ", "El usuario ya tiene la m&quina No. " +
ocupa.maquina + " Para el " + strVal(fecha.value) + " a " + hora.value
+ " Se cancela la reservacion")
fecha.moveto()
action(DataBeginEdit)
action(DataDeleteRecord)
action(DataEndEdit)
formreturn("Cancelar")
close()
else

; *** Advertencia para otra reservacion en el dia curso
```

Apéndice A

```
if fecha.value = today() then
  elige = msgQuestion("¿Cuidado", "El alumno ya tiene reservado para hoy...\n"
    + "\t\t Se continua con la reservaci≤n ?")
if elige = "Yes" then
  NoCuenta.value = cuenta.value
  NoMaquina.value = maquina.value
else
  fecha.moveto()
  action(DataBeginEdit)
  action(DataDeleteRecord)
  action(DataEndEdit)
  formreturn("Error")
close()
endif
else

; *** Solo una hora por dia
  msgStop("¡ ERROR !", "Lo siento el usuario ya reservo previamente" +
    "\n en la maquina No. " + ocupa.maquina)
  fecha.moveto()
  action(DataBeginEdit)
  action(DataDeleteRecord)
  action(DataEndEdit)
  formreturn("Error")
close()
endif
endif
endif
else

; *** Verificacion de que el equipo no este asignado
if ocupa.locate("maquina", maquina.value) then
if ocupa.fecha = fecha.value then
if ocupa.hora = hora.value then
  msgStop("¡ ERROR !", "Lo siento el equipo ya esta reservado")
  fecha.moveto()
  action(DataBeginEdit)
  action(DataDeleteRecord)
  action(DataEndEdit)
  formreturn("Error")
close()
endif
endif
endif
endif
endif
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
endif
endif
endif
endif
NoCuenta.value = cuenta.value
NoMaquina.value = maquina.value
formreturn("Aceptar")
endif
endif
endif
endmethod
```

## Programa 2.5

```
method pushButton(var eventInfo Event)
```

```
;*****  
; METODO PUSHBUTTON FORMA RESERVACION DE TIEMPO, BOTON CANCELAR  
; CANCELA CAMBIOS Y REGRESA A LA FORMA PRINCIPAL DATOS DEL ALUMNO  
; PROGRAMA 2.5  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
;*****
```

```
action(DataDeleteRecord)  
formreturn("Cancelar")  
close()  
endmethod
```

## **Programa 3.1**

```
method arrive(var eventInfo MoveEvent)
```

```
;*****  
; METODO ARRIVE FORMA CONFIRMACION DE APARTADO  
; SELECCIONA ALUMNOS QUE RESERVARON SEGUN LA FECHA Y LA HORA  
; PROGRAMA 3.1  
; DIANA ANABEL HERNANDEZ ALONSO  
; GABRIELA DE JESUS MAGALLANES GONZALEZ  
;*****
```

```
var  
    consulta query ;Consulta para seleccionar segun fecha y hora en curso  
    hoy date       ;Fecha actual  
    hora String    ;Hora en curso  
endVar
```

```
hoy = today()  
addAlias("aparta", "Standard", "c:\\pdxwi\\trabajo")  
hora = SubStr(StrVal(time()), 1, 2) + ":45"  
; *** Consulta que selecciona alumnos que reservaron segun la fecha  
; *** y hora del sistema  
consulta = Query
```

```
:aparta:reserva.db| Cuenta | Maquina | fecha      | hora      |  
                  | Check | Check  | Check today | check < ~hora|
```

```
EndQuery
```

```
executeQBE(consulta, "Consulta.db") ;Se ejecuta consulta  
action(DataBeginEdit)  
endmethod
```

## Programa 3.2

```

method setFocus(var eventInfo Event)
;*****
; METODO SETFOCUS FORMA CONFIRMACION DE APARTADO, OBJETO: CUENTA
; (Numero de Cuenta)
; VERIFICACION DEL NUMERO DE CUENTA Y BUSQUEDA EN RESERVACION
; PROGRAMA 3.2
; DIANA ANABEL HERNANDEZ ALONSO
; GABRIELA DE JESUS MAGALLANES GONZALEZ
;*****

var
    ncta,                ;Numero de Cuenta
    cadena string       ;Digito verificador convertido en cadena
    sump,               ;Suma de digitos pares
    sumn,               ;Suma de digitos nones
    i,                  ;Indice para el arreglo
    dig number          ;Digito verificador
    TCAlumno,           ;Apuntador a registro de la tabla alumnos
    TCAparta,           ;Apuntador a registro de la tabla consulta
                        ;creada por el metodo arrive de la forma
    TCReserva TCursor  ;Apuntador a registro de la tabla reserva
endvar

ncta = Cuenta.value      ;Inicilizacion y asignacion de tablas
TCAlumno.open("Alumnos.db")
TCAparta.open("Consulta.db")
TCReserva.open("Reserva.db")
sump = 0
sumn = 0
ncta = Cuenta.value
if ncta.size() <> 9 then  ;Si no es correcto el tama±o
    msgStop("ERROR", "Numero de Cuenta Incorrecto")
    disableDefault
    Cuenta = blank()
    Cuenta.moveTo()
else
; *** Caracteres validos
for i from 1 to 9

```

```
if NOT (substr(ncta,i,1) >= "0" AND substr(ncta,i,1) <= "9")AND i <> 8 then
msgStop("ERROR","N·mero de Cuenta Incorrecto")
disableDefault
Cuenta = blank()
Cuenta.moveTo()
quitloop
endif
endfor
```

```
if substr(ncta,8,1) <> "-" then
msgStop("ERROR","N·mero de Cuenta Incorrecto")
disableDefault
Cuenta = blank()
Cuenta.moveTo()
else
```

```
; *** Suma de digitos nones
for i from 1 to 7 step 2
sumn = numval(substr(ncta,i,1)) + sumn
endifor
```

```
; *** Suma de digitos pares
for i from 2 to 6 step 2
sump = numval(substr(ncta,i,1)) + sump
endifor
```

```
dig = sump*3 + sumn*7 ;Digito verificador
cadena = strval(dig)
```

```
; *** Checar digitos
if (substr(cadena,cadena.size()-3,1) <> substr(ncta,9,1)) then
beep()
msgInfo(" E R R O R !!", "N·mero de cuenta incorrecto")
disabledefault
Cuenta.moveto()
Cuenta = blank()
else
```

```
; *** Localizacion del alumno en la tabla
if not TCAlumno.locate("Cuenta",ncta) then
beep()
msgInfo(" E R R O R !!", "El alumno no esta registrado")
```

## Apéndice A

```
disabledefault
Cuenta.moveto()
Cuenta = blank()
else

; *** Localizacion en las reservaciones
if TCAparta.locate("Cuenta",ncta) then
Nombre.value = TCAlumno.nombre
Apellidos.value = TCAlumno.apellidos
Maquina.value = TCAparta.maquina
sleep(1000)
action(DataEndEdit)
confirma = msgQuestion("Confirmado", "¿ Se procede a asignar?")
if confirma = "Yes" then
; se asigna
else
TCAparta.Edit()
TCAparta.deleteRecord()
TCAparta.endEdit()
TCReserva.locate("Cuenta",ncta)
endif

; *** Si no existe reservacion
else
msgStop("Error",TCAlumno.Nombre + " " + TCAlumno.Apellidos +
"\nNo tiene reservado")
disableDefault
Cuenta.moveTo()
Cuenta = blank()
endif
endif
endif
endif
endmethod
```

# **Apéndice**

## **B**

### **Programas Controlador**



## Apéndice B

```
edo = atoi(argv[2]);
compu = formato(p,&edo);
transmisor(compu);
}

/* ----- Procedimiento Inicia la Comunicacion ----- */
/* se define el protocolo de comunicacion, el puerto por el cual */
/* se hara la comunicacion dx=0 es com 1 */
*/

inicia_com()
{
union REGS r;
r.h.ah = 0;
r.h.al = 0XE3; /* baub = 9600 1b = start 1b = stop longitud = 8b */
r.x.dx = 0;
int86(0X14,&r,&r);
return(r.h.ah);
}

/* ----- Interrupcion Receptor ----- */

receptor()
{
int caracter;
union REGS r;
r.h.ah = 2;
r.x.dx = 0;
int86(0X14,&r,&r);
caracter = r.h.al;
return(caracter);
}

/* ----- Interrupcion transmisor ----- */

transmisor(p)
char p;
{
int caracter,cond;

caracter = p; /* p es el dato que contiene
la computadora y el estado */
envia_caracter(caracter);
```

```
/* funcion que hace la trnasmision del caracter
utilizando le interrup. 14 funcion 1*/
```

```
return(caracter);
}
```

```
/* ----- Procedimiento que transmite caracter -----*/
```

```
envia_caracter(caracter)
```

```
int caracter;
```

```
{
union REGS r;
r.h.ah = 1;
r.x.dx = 0;
r.h.al = caracter;
int86(0X14,&r,&r);
return(r.h.ah);
}
```

```
/* ----- formato_dato_a_transmitir ----- */
```

```
char formato(comp,estado)
```

```
char comp;
```

```
int *estado;
```

```
{
comp--;
comp&= 15;
if (*estado)
comp|= 8;
comp<<= 3;
return comp;
}
```

## Programa 2

\*\*\*\*\*

- \* Programa en ensamblador, almacenado en la memoria EEPROM del
- \* microcontrolador MC68HC11E2.
- \* El programa recibe un dato del puerto serial de la PC central
- \* lo examina y envia este dato por el Puerto A para seleccionar
- \* la computadora que recibirá corriente eléctrica.

\* Realizado : Diana Anabel Hernández Alonso  
 \* Gabriela de Jesús Magallanes González

julio/93

\*\*\*\*\*

### Habilitar Registros.

PORTA	EQU	\$1000	*Asigna Loc. 1000 al PORT A
PORTD	EQU	\$1008	*Asigna Loc. 1008 al PORT D
PACTL	EQU	\$1026	*Asigna Loc. 1026 al reg PACTL

### Configurar Registros.

ORG	\$F8D0	*Lectura en la EPROM
LDS	#\$FF	*Tope de la RAM
LDX	#\$1000	*Apuntador registros internos.
		*Dirección relativa
LDA A	#\$02	*Configura registro DDRD, TXD=1, RXD=0
STAA	\$09,X	*Se carga al registro DDRD
LDA A	#\$88	*Carga A Config. bit 3 del puerto A
STAA	PACTL	*Puerto A (bit 3) Config. de salida
LDA A	#\$B0	*Configuracion a 9800 BAUDS,
		*ion 8, 1bstart,1bstop
STAA	\$2B,X	*Se carga al registro BAUD

```

*****
*          PROGRAMA PRINCIPAL
*****
INICIO    JSR      SERIAL
          JSR      RECIBE
          JSR      PUERTOA
          JMP      INICIO
*****
SERIAL    LDD      #$000C    * SCCR1 = $00, SCCR2 = $0C,
          STD      $2C,X     * sin interrup, sin razgos
          RTS          * manda comando $102C
*****
RECIBE    BRCLR   $2E,X     $20      RECIBE
          LDAA    $2F,X
          STAA    $2F,X     * Checa que haya información en el pto.
TRANS    BRCLR   $2E,X     $80      TRANS
          RTS          * Checa si ha transmitido
*****
PUERTOA  ORAA    #$80
          STAA    PORTA
          JSR     PAUSA
          ANDA    #$7F
          STAA    PORTA
          JSR     PAUSA
          ORAA    #$80
          STAA    PORTA
          JSR     PAUSA
          RTS
*****
PAUSA    PSHA
          PSHB
          PSHX
          LDX     #$FFFF
REG      DEX
          BNE     REG
          PULX
          PULB
          PULA
          RTS

```



```
ORG    $FFFE  
DW     $F8D0  
END
```

# **Apéndice**

## **C**

### **Programas Tarjeta**

## APENDICE C

### Programa 1

```
/* Programa para el manejo de la tarjeta PIB */
```

```
/* pibtool.c */
```

```
/*
```

This program contains c routines to interface ATronics PIB/PIB+ Board. Those are provided only for the convenience of the programmer. ATronics makes no warranty on the correctness of those routines.

Language

Microsoft C Compiler V 3.00 or V 4.00 Large model

History

when	who	version	what
04/01/87	w.l.h	V 1.00	PIB interface
06/23/87	w.l.h	V 1.10	Adjust the resolution to 512 by 240
07/05/87	w.l.h	V 1.20	fixed bugs in WaitOdd and WaitEven
09/15/87	w.l.h	V 1.30	1. Use same blanking and sync parameters for both internal and external modes. 2. modify FillPibRect
04/14/88	c.y.c	V1.32	Add SetUnderScan() SetVerBlank() SetHorBlank()
05/25/88	c.y.c	V1.32	routines to take care overscan Combine NTSC and PAL into single package
12/27/88	c.y.c	V1.53	Stablized VCR input for PAL Standard
01/19/89	c.y.c	V1.54	Stablized NTSC PIB output for VCR

```
*/
```

```
#include "conio.h"
```

```
#include "dos.h"
```

```
unsigned io_base=0X3a8 ;  
unsigned seg_addr=0XD000;  
int XMAX=512;
```

Apéndice C

```
int YMAX = 241;  
int page = 0;  
int source = 0; /* NTSC / PAL */
```

```
/* PIB registers, offset from IO base address ( io_base ) */
```

```
#define MODE_R io_base  
#define EXTRA_MODE_R MODE_R + 1  
#define INDEX_R MODE_R + 2  
#define DATA_R MODE_R + 3  
#define MEM_S_R MODE_R + 4  
#define STATUS_R MODE_R + 7
```

```
/* PIB internal registers index */
```

```
#define H_SYNC_L 0  
#define H_SYNC_H 2  
#define H_END_BLANK_L 4  
#define H_END_BLANK_H 6  
#define H_START_BLANK_L 8  
#define H_START_BLANK_H 10  
#define H_TOTAL_L 12  
#define H_TOTAL_H 14  
#define V_SYNC_L 16  
#define V_SYNC_H 18  
#define V_END_BLANK_L 20  
#define V_END_BLANK_H 22  
#define V_START_BLANK_L 24  
#define V_START_BLANK_H 26  
#define V_TOTAL_L 28  
#define V_TOTAL_H 30  
#define DSP_UPDATE_L 32  
#define DSP_UPDATE_H 34  
#define DSP_START_L 0X24  
#define DSP_START_H 0X26  
#define VSC_CONT1_L 0X2c  
#define VSC_CONT1_H 0X2e  
#define VSC_CONT2_L 0X30  
#define VSC_CONT2_H 0X32  
  
#define BRIGHT_R 0X85  
#define MASK_R_L 0X86  
#define MASK_R_H 0X87  
#define HUE_R 0X88
```

```
#define      CONTRAST_R      0X89
#define      SAT_R           0X8A

#define      M_TO_S         0X2000
#define      S_TO_M         0X1000
```

```
static char *pibtoolid = "PibToolId";
int  hor_start=306, hor_end=50;
int  ver_start=258, ver_end=17;
int  new_pal_hor=0, palver=0;
```

```
/* SetNewPAL1989 */
SetNewPAL1989()
{
    new_pal_hor=8;
} /* SetNewPAL1989 */
```

```
/* SetSource (source)
   int source;
   mode/2 = 0 (NTSC)
   mode/2 = 1 (PAL )
*/
```

```
int SetSource (mode)
int mode;
{
    if (mode <= 3)
        source = mode/2;
    if (source)
        /* palver=1; */
        palver=0;
    else
        palver=0;
} /* SetSource */
```

```
int GetSource ()
{
    return(source);
} /* GetSource */
```

Apéndice C

```
/* SetUnderScan (mode,resolution)
int mode;
    Under scan monitor over scan
    mode = 0 (normal 512X241 or 512X482
    mode = 1 (underscan 480X200 or 480X400
        is compatible with HaloVision
int resolution;
    0 512x256 (display page 0)
    1 512x256 (display page 1)
    2 512x512 (only one page)
    3 256x256 (display page 0)
    4 256x256 (display page 1)
*/
SetUnderScan(mode,resolution)
int mode;
int resolution;
{
if (GetSource() == 1) {
switch (mode) { /* PAL */
case 1 : /* under scan */
    XMAX = (resolution <= 2)?(480):(240);
    YMAX = (resolution == 2)?(420):(210);
    hor_start=294; hor_end = 54;
    ver_start=270; ver_end = 60;
    break;
case 0 : /* normal */
default:
    XMAX = (resolution <= 2)?(512):(256);
    YMAX = (resolution == 2)?(506):(253);
    hor_start=310; hor_end = 54;
    ver_start=308; ver_end = 55;
    break;
}
} else {
switch (mode) { /* NTSC */
case 1 : /* under scan */
    XMAX = (resolution <= 2)?(480):(240);
    YMAX = (resolution == 2)?(400):(200);
    hor_start=290; hor_end = 50;
    ver_start=237; ver_end = 37;
    break;
case 0 : /* normal */
```

```
default:
    XMAX = (resolution <= 2)?(512):(256);
    YMAX = (resolution == 2)?(482):(241);
    hor_start=306; hor_end = 50;
    ver_start=258; ver_end = 17;
    break;
}

} /* else */
SetHorBlank(hor_start+new_pal_hor,hor_end+new_pal_hor);
SetVerBlank(ver_start,ver_end);
}
```

```
SetHorBlank(start,end)
int start,end;
{
    outp(INDEX_R, H_END_BLANK_L);
    outp(DATA_R , end&0xFF);
    outp(INDEX_R, H_END_BLANK_H);
    outp(DATA_R , (end > > 8)&0xFF);
    outp(INDEX_R, H_START_BLANK_L);
    outp(DATA_R , start&0xFF);
    outp(INDEX_R, H_START_BLANK_H);
    outp(DATA_R , (start > > 8)&0xFF);
}
```

```
SetVerBlank(start,end)
int start,end;
{
    outp(INDEX_R, V_END_BLANK_L);
    outp(DATA_R , end&0xFF);
    outp(INDEX_R, V_END_BLANK_H);
    outp(DATA_R , (end > > 8)&0xFF);
    outp(INDEX_R, V_START_BLANK_L);
    outp(DATA_R , start&0xFF);
    outp(INDEX_R, V_START_BLANK_H);
    outp(DATA_R , (start > > 8)&0xFF);
}
```

```
SetDspUpdate(val)
int val;
```

## Apéndice C

```
{  
  outp(INDEX_R, DSP_UPDATE_L);  
  outp(DATA_R , val&0xFF);  
  outp(INDEX_R, DSP_UPDATE_H);  
  outp(DATA_R , (val >> 8)&0xFF);  
}
```

SetHorTotal(total)

```
int total;  
{  
  outp(INDEX_R, H_TOTAL_L);  
  outp(DATA_R , total&0xFF);  
  outp(INDEX_R, H_TOTAL_H);  
  outp(DATA_R , (total >> 8)&0xFF);  
}
```

SetVerTotal(total)

```
int total;  
{  
  outp(INDEX_R, V_TOTAL_L);  
  outp(DATA_R , total&0xFF);  
  outp(INDEX_R, V_TOTAL_H);  
  outp(DATA_R , (total >> 8)&0xFF);  
}
```

SetHorSync(sync)

```
int sync;  
{  
  outp(INDEX_R, H_SYNC_L);  
  outp(DATA_R , sync&0xFF);  
  outp(INDEX_R, H_SYNC_H);  
  outp(DATA_R , (sync >> 8)&0xFF);  
}
```

SetVerSync(sync)

```
int sync;  
{  
  outp(INDEX_R, V_SYNC_L);  
  outp(DATA_R , sync&0xFF);  
  outp(INDEX_R, V_SYNC_H);  
  outp(DATA_R , (sync >> 8)&0xFF);  
}
```

```
}
```

```
/*
```

```
SetScreen ( mode )
```

```
int mode;
```

```
SetScreen set the screen resolution and page.
```

```
Input
```

```
0 512x256 ( display page 0 )
```

```
1 512x256 ( display page 1 )
```

```
2 512x512 ( only one page )
```

```
3 256x256 ( display page 0 )
```

```
4 256x256 ( display page 1 )
```

```
*/
```

```
SetScreen ( mode )
```

```
int mode;
```

```
{
```

```
int c;
```

```
switch ( mode )
```

```
{
```

```
case 0 :
```

```
    XMAX = 512;
```

```
    YMAX = (GetSource() == 1)?(254):(241);
```

```
    c = inp ( EXTRA_MODE_R );
```

```
    c = ( c & 0XAF );
```

```
    outp ( EXTRA_MODE_R, c );
```

```
    c = inp ( MODE_R );
```

```
    c = ( c & 0XF6 );
```

```
    SetInterlace ( 0 );
```

```
    break;
```

```
case 1 :
```

```
    XMAX = 512;
```

```
    YMAX = (GetSource() == 1)?(254):(241);
```

```
    c = inp ( EXTRA_MODE_R );
```

```
    c = ( c & 0XAF ) | 0x10;
```

```
    outp ( EXTRA_MODE_R, c );
```

```
    c = inp ( MODE_R );
```

```
    c = ( c & 0XF6 ) | 0x08 ;
```

```
    SetInterlace ( 0 );
```

```
    break;
```

```
case 2 :
```

```

XMAX = 512;
YMAX = (GetSource() == 1)?(508):(482);
c = inp ( EXTRA_MODE_R );
c = ( c & 0XAF );
outp ( EXTRA_MODE_R, c );
c = inp ( MODE_R );
c = ( c & 0XF6 ) | 0x01;
SetInterlace ( 1 );
break;
case 3 :
XMAX = 256;
YMAX = (GetSource() == 1)?(254):(241);
c = inp ( EXTRA_MODE_R );
c = ( c & 0XAF ) | 0x40 ;
outp ( EXTRA_MODE_R, c );
c = inp ( MODE_R );
c = ( c & 0XF6 ) ;
SetInterlace ( 0 );
break;
case 4 :
XMAX = 256;
YMAX = (GetSource() == 1)?(254):(241);
c = inp ( EXTRA_MODE_R );
c = ( c & 0XAF ) | 0x50;
outp ( EXTRA_MODE_R , c );
c = inp ( MODE_R );
c = ( c & 0XF6 ) | 0x08 ;
SetInterlace ( 0 );
break;
}
outp ( MODE_R, c );
}

```

/\*

CaptureFrame ()

This routine captures a frame of image to the display memory. If no external source is attached, it will return a time out error message.

result

0	No external source
1	OK

\*/

CaptureFrame ()

```
{
int c, d, i;
int mode;
int sts;

    outp ( INDEX_R, VSC_CONT1_H );
    c = inp ( DATA_R );
    c = ( c & 0XFE ) | 0X01;
    outp ( DATA_R, c );

    c = inp ( MODE_R );
    if ( c & 0X01 ) mode = 2;
    else if ( c & 0x08 ) mode = 1;
    else mode = 0;

    c = ( c & 0X8F );
    outp ( MODE_R, c );

    if ( sts = WaitVsync () )
    {
        outp ( INDEX_R, DSP_START_L ); /* display start from row 255 */
        if (GetSource())
            outp ( DATA_R, 0XF8 );
        else
            outp ( DATA_R, 0XFC );
        outp ( INDEX_R, DSP_START_H );
        outp ( DATA_R, 3 );

        outp ( INDEX_R, VSC_CONT1_L );
        c = inp ( DATA_R );
        d = ( c & 0XBF ) | 0X40;
        outp ( DATA_R, d );
        outp ( MEM_S_R, 0 );
        outp ( MEM_S_R, 1 );
        SaveData ( seg_addr, S_TO_M, 0 );

        if ( mode == 2 )
            WaitVsync();
            WaitVsync();
    }
}
```

```

        outp ( MEM_S_R, 0 );
        outp ( INDEX_R, VSC_CONT1_L );
        outp ( DATA_R, c );
    }
    outp ( INDEX_R, DSP_START_L );    /* display start from row 0 */
    outp ( DATA_R, 0 );
    outp ( INDEX_R, DSP_START_H );
    outp ( DATA_R, 0 );
    return ( sts );
}

```

```

/*
SetInDispMode ()
SetInDispMode sets PIB to the internal Sync display mode and
also enables the direct memory access.

```

**NOTE :** The sequence to set the registers is crucial. Don't try to change the sequence.

```

*/
SetInDispMode ()
{
    unsigned char c;
    int ss, ee;

    if (GetSource() == 0)
        SetVerSync(2);
    SetVerBlank(ver_start, ver_end);
    SetHorBlank(hor_start + new_pal_hor, hor_end + new_pal_hor);
    c = inp ( MODE_R );
    c = ( c & 0X8F ) | 0X30;
    outp ( MODE_R, c );

    outp ( INDEX_R, VSC_CONT1_H );
    c = inp ( DATA_R );
    if (GetSource() == 1)
        c = ( c & 0XFE );
    else
        c = ( c & 0XFE );
    outp ( DATA_R, c );
}

```

```
/*  
SetExtDispMode ()  
    SetExtDispMode sets PIB to the display memory with external sync  
    and enables direct access.
```

NOTE : The sequence to set the registers is crucial. Don't try to change the sequence.

```
*/  
SetExtDispMode ()  
{  
    unsigned char c;  
  
    if (GetSource() == 0)  
        SetVerSync(3);  
    SetVerBlank(ver_start + palver, ver_end + palver);  
    SetHorBlank(hor_start-2, hor_end-2);  
  
    outp ( INDEX_R, VSC_CONT1_H );  
    c = inp ( DATA_R );  
    c = ( c & 0XFE ) | 0x01;  
    outp ( DATA_R, c );  
  
    c = inp ( MODE_R );  
    c = ( c & 0X8F ) | 0X70;  
    outp ( MODE_R, c );  
}
```

```
/*  
SetMixMode ()  
    SetMixMode sets to the mixed mode and direct memory access.
```

NOTE : The sequence to set the registers is crucial. Don't try to change the sequence.

```
*/  
SetMixMode ()  
{  
    unsigned char c;  
  
    if (GetSource() == 0)  
        SetVerSync(3);  
    SetVerBlank(ver_start + palver, ver_end + palver);
```

## Apéndice C

```
SetHorBlank(hor_start-2,hor_end-2);
```

```
    outp ( INDEX_R, VSC_CONT1_H );  
    c = inp ( DATA_R );  
    c = ( c & 0XFE ) | 0X01;  
    outp ( DATA_R, c );
```

```
    c = inp ( MODE_R );  
    c = ( c & 0X8F ) | 0X50 ;  
    outp ( MODE_R, c );
```

```
}
```

```
/*
```

```
SetLiveMode ()
```

SetLiveMode sets to the live mode and direct memory access.

NOTE : The sequence to set the registers is crucial. Don't try to change the sequence.

```
*/
```

```
SetLiveMode ()
```

```
{  
    unsigned char c;
```

```
    if (GetSource() == 0)
```

```
        SetVerSync(3);
```

```
    SetVerBlank(ver_start + palver,ver_end + palver);
```

```
    SetHorBlank(hor_start-2,hor_end-2);
```

```
    outp ( INDEX_R, VSC_CONT1_H );
```

```
    c = inp ( DATA_R );
```

```
    c = ( c & 0XFE ) | 0X01;
```

```
    outp ( DATA_R, c );
```

```
    c = inp ( MODE_R );
```

```
    c = ( c & 0X8F ) | 0X10;
```

```
    outp ( MODE_R, c );
```

```
}
```

```
/*
```

**SetHue ( value )**

int value;

SetHue sets input hue to the value specified.

input

value        0 -- 255

\*/

**SetHue ( value )**

int value;

{

    outp ( INDEX\_R, HUE\_R );

    outp ( DATA\_R, value );

}

/\*

**SetContrast ( value )**

int value;

SetContrast sets input contrast to the value specified.

input

value        0 -- 255

\*/

**SetContrast ( value )**

int value;

{

    outp ( INDEX\_R, CONTRAST\_R );

    outp ( DATA\_R, value );

}

/\*

**SetSaturation ( value )**

int value;

SetSaturation sets input saturation to the value specified.

input

value        0 -- 255

\*/

*Apéndice C*

**SetSaturation ( value )**

```
int value;  
{  
    outp ( INDEX_R, SAT_R );  
    outp ( DATA_R, value );  
}
```

/\*

**SetBright ( value )**

```
int value;
```

SetBright set input brightness to the value specified.  
value        0 -- 255

\*/

**SetBright ( value )**

```
int value;  
{  
    outp ( INDEX_R, BRIGHT_R );  
    outp ( DATA_R, value );  
}
```

/\*

**SetInterlace ( mode )**

```
int mode;
```

SetInterlace enable or disable the interlace

input

mode 0	Non_interlace
1	interlace

\*/

**SetInterlace ( mode )**

```
int mode;
```

```
{  
    unsigned char c;  
  
    outp ( INDEX_R, VSC_CONT1_H );  
    c = inp ( DATA_R );  
    c &= 0XFD ;  
    if ( mode ) c |= 0X02;  
    outp ( DATA_R, c );  
}
```

```
/*  
SetMask ( mask )  
unsigned mask;  
  
SetMask sets the two bytes mask register to the value specified.  
If a bit of the mask register is set to 0, then the subsequence  
write to a pixel will not change the corresponding bit of that  
pixel.
```

```
*/  
  
SetMask ( mask )  
unsigned mask;  
{  
int c;  
  
c = ( mask & 0Xff );  
outp ( INDEX_R, MASK_R_L );  
outp ( DATA_R, c );  
c = ( mask >> 8 ) & 0Xff;  
outp ( INDEX_R, MASK_R_H );  
outp ( DATA_R, c );  
}
```

```
/*  
SetCamera ( mode )  
int mode;  
  
SetCamera enable the NTSC or RGB source.
```

```
input  
mode 0    NTSC  
      1    RGB  
mode 2    PAL  
      3    RGB
```

```
*/  
SetCamera ( mode )  
int mode;  
{  
unsigned char c;  
  
c = inp ( EXTRA_MODE_R );
```

## Apéndice C

```
c &= 0X1D;
if ( mode%2 ) c |= 0X02;
outp ( EXTRA_MODE_R, c );
```

```
}
```

```
/*
```

```
SetPIBorEGA ( mode )
```

```
int mode;
```

SetPIBorEGA decides output is from PIB or EGA.

```
input
```

mode 0	EGA
1	PIB

```
*/
```

```
SetPIBorEGA ( mode )
```

```
int mode;
```

```
{
```

```
unsigned char c;
```

```
c = inp ( EXTRA_MODE_R );
c &= 0X1E;
if ( mode ) c |= 0X01;
outp ( EXTRA_MODE_R, c );
```

```
}
```

```
/*
```

```
SetBlank ( mode ) .
```

```
int mode;
```

SetBlank blank or show PIB screen.

```
input
```

mode 0	blank the screen
1	show the screen

```
*/
```

```
SetBlank ( mode )
```

```
int mode;
```

```
{
```

```
unsigned char c;
```

```
    outp ( INDEX_R, VSC_CONT2_H );
    c = inp ( DATA_R );
    c &= 0XDF;
    if ( mode ) c = 0X20;
    outp ( DATA_R, c );
}

/*
void FillPibRect ( color, x1, y1, x2, y2, mode )
unsigned color ;    color to fill
int x1, y1;
int x2, y2;
int mode;
```

#### Description

FillPibRect fills the rectangle area specified by the two corners x1, y1 and x2, y2 with the color.

```
*/
FillPibRect ( color, x1, y1, x2, y2, mode )
unsigned color ;
int x1, y1;
int x2, y2;
int mode;
{
    int xstart, ystart;
    int xend, yend;
    int i;

    xstart = x1;
    ystart = y1;
    xend = x2;
    yend = y2;
    if ( x2 < x1 )
        { xstart=x2; xend = x1; };
    if ( y2 < y1 )
        { ystart = y2; yend = y1; };

    for ( i = ystart ; i <= yend ; i++ )
        FillPibRow ( color, xstart, xend, i , mode);
}

WaitEven ()
```

*Apéndice C*

```
{  
    while ( !inp( STATUS_R ) & 0x04 );  
    while ( !inp( STATUS_R ) & 0x04 );  
}
```

WaitOdd()

```
{  
    while ( !inp( STATUS_R ) & 0x04 );  
    while ( !inp( STATUS_R ) & 0x04 );  
}
```

## Programa 2

```
;/* Programa para el manejo de la tarjeta PIB */
;
; /* imgio.asm */
;
;This program contains a couple of routines which can be called through
;programs generated by Microsoft C compiler V 3.00 or V 4.00 LARGE MODEL.
;
;LANGUAGE
;    Microsoft Macro Assembly

;History
;when      who      version      what
;04/07/87  w.l.h      V 1.00      First version
;09/15/87  w.l.h      V 1.01      Change blanking and sync parameters
;04/14/88  c.y.c      Add new Genlock source switching
;          automate by redirecting timer.
;          Insync -- test whether is int or ext
;          genlock source
;05/26/88  c.y.c      Combine NTSC and PAL into single
;          module
;
;12/23/88  c.y.c      V 1.53      Stablized VCR input for PAL Standard
;01/19/89  c.y.c      V 1.54      Stablized NTSC PIB output for VCR

TRUE      equ      1
FALSE     equ      0

;PIB registers, offset from IO base address ( io_base )
IO_MODE_R EQU 0
EXTRA_MODE_R EQU 1
INDEX_R EQU 2
DATA_R EQU 3
MEM_S_R EQU 4
STATUS_R EQU 7

;PIB internal registers index
VSC_CONT1_L EQU 2cH ;16H
VSC_CONT1_H EQU 2eH ;17H
VSC_CONT2_L EQU 30H ;18H
VSC_CONT2_H EQU 32H ;19H
```

Apéndice C

```
DSP_START_L EQU 24H
DSP_START_H EQU 26H
M_TO_S EQU 2000H
S_TO_M EQU 1000H

REF_VOL_R EQU 84H ;42H
MASK_R_L EQU 86H ;46H
MASK_R_H EQU 87H ;47H
HUE_R EQU 88H ;48H
CONTRAST_R EQU 89H ;49H
SAT_R EQU 8Ah ;4AH

H_END_BLANK_L EQU 04H
H_END_BLANK_H EQU 06H
H_START_BLANK_L EQU 08H
H_START_BLANK_H EQU 0AH
```

```
BASE equ 2
```

```
;-----
; Data area
;-----
```

```
DGROUP GROUP _DATA
```

```
extrn _io_base:WORD
extrn _seg_addr:WORD
extrn _XMAX:WORD
extrn _YMAX:WORD
```

```
extrn _hor_end :WORD
extrn _hor_start:WORD
extrn _ver_end :WORD
extrn _ver_start:WORD
```

```
_DATA segment word public 'DATA'
```

```
; Initial value for PIB internal registers
; the first byte is value, second byte is mask byte
; NTSC Initialization Sequence
public _ntsc_para
imgioidn db "ImgioldNTSC"
_ntsc_para label byte
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
init_seq1    db    23, 0ffh    ; horizontal end syn
             db    0, 0ffh
             db    50, 0ffh    ; horizontal end blank
             db    0, 0ffh
             db    50, 0ffh    ; horizontal start blank
             db    1, 0ffh
             db    62, 0ffh    ; horizontal total
             db    1, 0ffh
             db    2, 0ffh      ; vertical end sync
             db    0, 0ffh
             db    17, 0ffh     ; vertical end blank
             db    0, 0ffh
             db    2, 0ffh      ; vertical start blank
             db    1, 0ffh
             db    5, 0ffh      ; vertical total
             db    1, 0ffh
             db    4, 0ffh      ; display address update
             db    0, 0
             db    0, 0ffh      ; display start
             db    0, 0ffh
             db    0, 0ffh      ; vertical interrupt
             db    0, 0ffh
             db    0, 0ffh      ; control register 1
             db    50h, 7fh
             db    0, 0ffh      ; control register 2
             db    24h, 7fh
init_seq_count1 equ ($-init_seq1)/2

ext_seq1     db          192      ; reference voltage
             db    60          ; brightness
             db    0ffh, 0ffh    ; mask register word
             db    128          ; hue
             db    128          ; contrast
             db    144          ; saturation
ext_seq_count1 equ ($ - ext_seq1)
;
; PAL Initialization Sequence
public      _pal_para
imgioidp   db    "imgloldPAL"
_pal_para  label byte
init_seq2  db    23, 0ffh      ; horizontal end syn
             db    0, 0ffh
```

Apéndice C

```
db 61, 0ffh ; horizontal end blank
db 0, 0ffh
db 61, 0ffh ; horizontal start blank
db 1, 0ffh
db 65, 0ffh ; horizontal total
db 1, 0ffh
db 2, 0ffh ; vertical end sync
db 0, 0ffh
db 55, 0ffh ; vertical end blank
db 0, 0ffh
db 52, 0ffh ; vertical start blank
db 1, 0ffh
db 55, 0ffh ; vertical total
db 1, 0ffh
db 4, 0ffh ; display address update
db 0, 0
db 0, 0ffh ; display start
db 0, 0ffh
db 0, 0ffh ; vertical interrupt
db 0, 0ffh
db 0, 0ffh ; control register 1
db 50h, 7fh
db 0, 0ffh ; control register 2
db 24h, 7fh
init_seq_count2 equ ($-init_seq2)/2

ext_seq2 db 192 ; reference voltage
db 180 ; brightness
db 0ffh, 0ffh ; mask register word
db 150 ; hue
db 145 ; contrast
db 165 ; saturation
ext_seq_count2 equ ($ - ext_seq2)
;
; data variables
_DATA ends

;-----
; Codes start from here
;-----
_TEXT segment byte public 'CODE'
```

assume cs:\_TEXT, ds: DGROUP

```
-----  
; /*  
; InitPIB (source)  
; Description  
; InitPIB initials the PIB to the internal display mode.  
;  
; Parameter  source 0,1 -- NTSC  
;             source 2,3 -- PAL  
; return value  
;     ax  0      OK  
;         others      address of the register that has red/write error  
; */  
-----
```

public \_InitPIB

```
_InitPIB    proc far  
    push    bp  
    mov     bp, sp  
    push    ds  
    push    si  
    push    di  
    cld
```

```
    mov     dx, _io_base      ; I/O Mode Register  
    mov     al, 30h          ; internal, direct access  
    out     dx, al  
    nop  
    nop  
    in     al, dx  
    cmp     al, 30h  
    jz     i1111  
    jmp     i2a
```

i1111:

```
    inc     dx                ; Extra Mode Register  
    mov     al, 00000101B     ; enable VSC , video output is from PIB  
    out     dx, al  
    nop  
    nop  
    in     al, dx
```

Apéndice C

```

    and al, 00000111B          ; mask out
    cmp al, 00000101B
    jz i111
    jmp i2a
i111:
    add dx, 3                  ; memory shift enable register
    xor al, al                 ; disable
    out dx, al

    mov dx, _io_base
    add dx, INDEX_R

    push ax
    mov ax, [bp+BASE+4]        ; NTSC / PAL
    cmp ax, 3
    jg i11
    cmp ax, 2
    jl i11

    pop ax
    mov cx, init_seq_count2
    mov si, offset DGROUP:init_seq2 ; PAL
    sub ah, ah
    jmp short i1
i11:
    pop ax
    mov cx, init_seq_count1
    mov si, offset DGROUP:init_seq1 ; NTSC
    sub ah, ah

i1:
    mov al, ah
    out dx, al
    inc ah
    inc ah
    inc dx
    lodsb                      ; load parameters
    mov bl, al
    out dx, al
    nop
    nop
    in al, dx
    xor bl, al

```

```
lodsb
and bl, al
jnz i2b

dec dx
loop i1

push ax
mov ax, [bp+BASE+4] ; NTSC / PAL
cmp ax, 3
jg i22
cmp ax, 2
jl i22

pop ax
mov cx, ext_seq_count2
mov si, offset DGROUP:ext_seq2 ; PAL
mov ah, REF_VOL_R
jmp short i2

i22:
pop ax
mov cx, ext_seq_count1
mov si, offset DGROUP:ext_seq1 ; NTSC
mov ah, REF_VOL_R

i2:
mov al, ah
out dx, al
inc ah
inc dx
lodsb ; load external parameters
out dx, al
dec dx
loop i2

sub ax, ax ; no error
pop di
pop si
pop ds
mov sp, bp
pop bp
ret
```

Apéndice C

```
i2b:  sub  ah, 2
      mov  al, 80h
      xchg al, ah
      mov  dx, ax
```

```
i2a:  mov  ax, dx
      pop  di
      pop  si
      pop  ds
      mov  sp, bp
      pop  bp
      ret
```

```
_InitPIB  endp
```

```
-----
; TestRow ( row )
; int row;
;
; TestRow will write two pattern ffffh and 0 to the row specified
; , then read back the data if the data is not correct it
; return a error
;
;
; return value
;   ax   1   OK
;       0   error
-----
```

```
public _TestRow
```

```
_TestRow  proc  far
  push  bp
  mov   bp, sp
  push  si
  push  di
  push  es
  mov   es, _seg_addr

  mov   ax, [bp+BASE+4]      ; get row
  call  set_bank
  mov   bx, 512
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
sal    bx, 1
mul    bx                ; ax = ( row mod 64 ) * 2*512

mov    bx, 0ffffh
call   RWrow
jc     t0
;mov   bx, 05555h
;call  RWrow
;jc    t0
mov    bx, 0
call   RWrow
jc     t0
mov    ax, 1

t1:
pop    es
pop    di
pop    si
mov    sp, bp
pop    bp
ret

t0:
xor    ax, ax
jmp    short t1

_TestRow    endp
```

```
-----
; RWrow
;
;   write and verify a row of data, a row has 512 pixels
; input
;   ax   starting address
;   bx   pattern to read / write
; return value
;   carry set   RW error
;   carry clear   OK
; destroy
;   cx, di
;-----
RWrow    proc    near
        cld
```

Apéndice C

```
    push ax                ; save ax = row * 512
    mov di, ax
    mov ax, bx
    mov cx, 512
    rep stosw
    pop ax
    mov di, ax
    mov cx, 512
t2:  cmp bx, es:[di]
     jnz t3
     inc di
     inc di
     loop t2
     clc
     ret
t3:  stc
     ret
RWrow endp
```

```
-----
; _SaveData ( destseg, destoff, value)
;
; This routine moves the value to the destination address.
;
-----
```

```
public _SaveData
_SaveData proc far
    push bp
    mov bp, sp
    push es
    push di

    mov ax, [bp+BASE+8]    ; ax has the value
    mov di, [bp+BASE+6]
    mov es, [bp+BASE+4]
    stosb

    pop di
    pop es
```

```
mov sp, bp
pop bp
ret
_SaveData endp
```

```
-----
; /*
; _WaitVsync ()
;
; This routine first waits VSYNC to be high, and then wait VSYNC to
; be low.
;
; Result
; 0 Time out
; 1 OK
; */
-----
```

```
public _WaitVsync
_WaitVsync proc far
push cx
push dx
mov dx, _io_base
add dx, STATUS_R
mov cx, 16000
```

```
m1:
in al, dx
test al, 01H ;wait for vertical sync high
jnz m2
loop m1
jmp short no_external_source

m2:
mov cx, 16000

m3:
in al, dx
test al, 01H ;wait for vertical sync low
jz m4
loop m3
jmp short no_external_source

m4:
```

## Apéndice C

```
mov ax, 1
pop dx
pop cx
ret
```

```
no_external_source :
```

```
xor ax, ax
pop dx
pop cx
ret
```

```
_WaitVsync endp
```

```
-----
; /*
; void FillPibRow ( color, x1, x2, y1, mode )
; unsigned color;
; int x1, x2, y2;
;
; Description
;
; FillPibRow Fills the row specified by ( x1, y1 ) and ( x2, y1 ) with color.
;
; Parameters
; color : color to fill,
; x1, y1 : starting address
; x2, y1 : ending address
; NOTE :
; 1. x2 > x1 and x2 < 512 and x1 >= 0
; 2. 0 <= y1 < YMAX
; Input
; mode = 0 set the pixel to the value specified.
; mode = 1 exclusive the pixel with value specified
; mode = 2 and the pixel with tvalue specified
;
; */
-----
```

```
public _FillPibRow
_FillPibRow proc far
push bp
mov bp, sp
push es
```

```
push si
push di
mov es, _seg_addr
mov ax, [bp+BASE+10] ; y1
call set_bank ; ax = y1 mod 64
mov bx, 512
sal bx, 1
mul bx
mov di, ax
mov ax, [bp+BASE+6] ; x1
mov cx, [bp+BASE+8] ; x2
sub cx, ax
inc cx ; cx = x2 - x1 + 1
sal ax, 1
add di, ax ; DI = destination offset = 2*x + (y mod 64)*2*512
mov ax, [bp+BASE+4] ; value
mov bx, [bp+BASE+12]
or bx, bx
jnz f1
rep stosw
jmp short f3

f1:
dec bx
jnz f2

f1a:
or ax, 8000h
xor word ptr es:[di], ax
inc di
inc di
loop f1a
jmp short f3

f2:
dec bx
jnz f3

f2a:
and word ptr es:[di], ax
inc di
inc di
loop f2a

f3:
pop di
```

## Apéndice C

```
pop si
pop es
mov sp, bp
pop bp
ret
```

```
_FillPibRow endp
```

```
-----
; /*
; FGetPibRow ( buf, x1, x2, y1 )
;
; Copys a row from PIB to the buffer, This routine does not check the
; boundary. Also it does't change the display mode. but it changes
; to direct access mode and memory bank.
```

```
; Parameters
```

```
; char far *buf : destination buffer
; int x1, y1 : starting address
; int x2, y1 : ending address
```

```
; NOTE :
```

```
; 1.  $x2 > x1$  and  $x2 < 512$  and  $x1 \geq 0$ 
; 2.  $0 \leq y1 < YMAX$ 
```

```
; */
-----
```

```
public _FGetPibRow
```

```
_FGetPibRow proc far
```

```
push bp
mov bp, sp
push ds
push es
push si
push di
mov ax, [bp+BASE+12] ; ax = y1
call set_bank ; ax = y1 mod 64
mov bx, 512
sal bx, 1
mul bx ; ax = (y1 mod 64) * 2*512
mov si, ax
mov ax, [bp+BASE+8] ; x1
mov cx, [bp+BASE+10] ; x2
sub cx, ax
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
inc cx ; CX = length = x2-x1 + 1
sal ax, 1
add si, ax ; SI = offset of the first pixel in the row
mov ds, _seg_addr

mov di, [bp+BASE+4] ; DI = destination
mov es, [bp+BASE+6]

rep movsw

pop di
pop si
pop es
pop ds
mov sp, bp
pop bp
ret
_FGetPibRow endp
```

```
-----
; /*
; FPutPibRow ( buf, x1, x2, y1 )
;
; moves a row of data from buf to PIB memory. This routine does not check the
; boundary. Also it doesn't change the display mode. but it changes
; to direct access mode and memory bank.
;
; Parameters
; char far *buf : source buffer
; int x1, y1 : starting address
; int x2, y1 : ending address
; NOTE :
; 1. x2 > x1 and x2 < 512 and x1 >= 0
; 2. 0 <= y1 < YMAX
; 3. Try to load more data then the size of buffer is a disaster.
; */
-----
public _FPutPibRow
_FPutPibRow proc far
push bp
```

Apéndice C

```
mov bp, sp
push ds
push es
push si
push di
mov es, _seg_addr
mov ax, [bp+BASE+12] ; y1
call set_bank ; ax = y1 mod 64
mov bx, 512
sal bx, 1
mul bx
mov di, ax
mov ax, [bp+BASE+8] ; x1
mov cx, [bp+BASE+10] ; x2
sub cx, ax
inc cx ; CX = length = x2-x1+1
sal ax, 1
add di, ax ; DI = destination offset = x1 + y1*2*512

mov si, [bp+BASE+4] ; SI = point to the buffer
mov ds, [bp+BASE+6]
rep movsw

pop di
pop si
pop es
pop ds
mov sp, bp
pop bp
ret
```

\_FPutPibRow endp

```
-----
; /*
; GetPibRow ( buf, x1, x2, y1 )
;
; Copys a row from PIB to the buffer, This routine does not check the
; boundary. Also it doesn't change the display mode. but it changes
; to direct access mode and memory bank.
;
; Parameters
```

```
; buf : destination buffer
; x1, y1 : starting address
; x2, y1 : ending address
; NOTE :
; 1.  $x2 > x1$  and  $x2 < 512$  and  $x1 \geq 0$ 
; 2.  $0 \leq y1 < YMAX$ 
;*/
```

```
-----
public _GetPibRow
_GetPibRow proc far
    push bp
    mov bp, sp
    push ds
    push es
    push si
    push di
    mov ax, [bp+BASE+12] ; ax = y1
    call set_bank ; ax = y1 mod 64
    mov bx, 512
    sal bx, 1
    mul bx ; ax = (y1 mod 64) * 2*512
    mov si, ax
    mov ax, [bp+BASE+8] ; x1
    mov cx, [bp+BASE+10] ; x2
    sub cx, ax
    inc cx ; CX = length = x2-x1+1
    sal ax, 1
    add si, ax ; SI = offset of the first pixel in the row
    mov ds, _seg_addr

    mov di, [bp+BASE+4] ; DI = destination
    mov es, [bp+BASE+6]

    rep movsw

    pop di
    pop si
    pop es
    pop ds
    mov sp, bp
    pop bp
    ret
```

\_GetPibRow endp

```

;-----
; /*
; PutPibRow ( buf, x1, x2, y1 )
;
; moves a row of data from buf to PIB memory. This routine does not check the
; boundary. Also it doesn't change the display mode. but it changes
; to direct access mode and memory bank.
;
; Parameters
;   buf : source buffer
;   x1, y1 : starting address
;   x2 : ending address
; NOTE :
;   1. x2 > x1 and x2 < 512 and x1 >= 0
;   2. 0 <= y1 < YMAX
;   3. Try to load more data then the size of buffer is a disaster.
; */
;-----

```

```

public _PutPibRow
_PutPibRow proc far
    push bp
    mov bp, sp
    push ds
    push es
    push si
    push di
    mov es, _seg_addr
    mov ax, [bp+BASE+12] ; y1
    call set_bank ; ax = y1 mod 64
    mov bx, 512
    sal bx, 1
    mul bx
    mov di, ax
    mov ax, [bp+BASE+8] ; x1
    mov cx, [bp+BASE+10] ; x2
    sub cx, ax
    inc cx ; CX = length = x2-x1+1
    sal ax, 1
    add di, ax ; DI = destination offset = x1 + y1*2*512

```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
mov si, [bp+BASE+4] ; SI = point to the buffer
mov ds, [bp+BASE+6]
rep movsw
```

```
pop di
pop si
pop es
pop ds
mov sp, bp
pop bp
ret
```

**\_PutPibRow** endp

**public \_PutLiveRow**

**\_PutLiveRow** proc far

```
push bp
mov bp, sp
push ds
push es
push si
push di
mov es, _seg_addr
mov ax, [bp+BASE+12] ; y1
call set_bank ; ax = y1 mod 64
mov bx, 512
sal bx, 1
mul bx
mov di, ax
mov ax, [bp+BASE+8] ; x1
mov cx, [bp+BASE+10] ; x2
sub cx, ax
inc cx ; CX = length = x2-x1+1
sal ax, 1
add di, ax ; DI = destination offset = x1 + y1*2*512
```

```
mov si, [bp+BASE+4] ; SI = point to the buffer
mov ds, [bp+BASE+6]
```

**\_PutLiveRow1:**

```
lodsw
test ax, 8000h
jz _PutLiveRow2
inc di
```

Apéndice C

```
    inc    di
    jmp    short _PutLiveRow3
_PutLiveRow2:
    stosw
_PutLiveRow3:
    loop  _PutLiveRow1

    pop    di
    pop    si
    pop    es
    pop    ds
    mov    sp, bp
    pop    bp
    ret

_PutLiveRow    endp
```

```
-----
; /*
; unsigned GetPixel ( color, x, y )
; unsigned *color;
; int x,y;    position to get the pixel
;
; Description
;   GetPixel returns the pixel value in the PIB display memory specified
;   by the location ( x, Y ).
;
; */
;-----
```

```
public _GetPixel
_GetPixel proc far
    push bp
    mov  bp, sp
    push si
    push di
    push es
    push ds

    mov ax, [bp+BASE+10]    ; ax = y
    call set_bank    ; ax = y mod 64
    mov bx, 512
    sal  bx, 1
```

```
mul  bx          ; ax = (y mod 64) * 2* 512
mov  si, ax
mov  ax, [bp+BASE+8] ; x
sal  ax, 1
add  si, ax      ; SI = offset of the pixel in the memory
mov  ds, _seg_addr
mov  ax, ds:[si] ; get pixel value
pop  ds
mov  si, [bp+BASE+4]

mov  es, [bp+BASE+6]
mov  word ptr es:[si], ax
```

```
pop  ds
pop  es
pop  di
pop  si
mov  sp, bp
pop  bp
ret
```

**\_GetPixel**    **endp**

```
-----
; /*
; void PutPixel ( value, x, y , mode)
; unsigned *value;
; int x, y;
; int mode;
; Description
;     PutPixel sets or exclusives the pixel specified by ( x, y )
;     to the value specified.
;
; Input
;     mode = 0    set the pixel to the value specified.
;     mode = 1    exclusive the pixel with value specified
;     mode = 2    and the pixel with tvalue specified
; */
-----
```

```
public _PutPixel
_PutPixel    proc    far
```

Apéndice C

```

push bp
mov bp, sp
push ds
push es
push si
push di
mov es, _seg_addr
mov ax, [bp+BASE+10] ; y
call set_bank ; ax = y mod 64
mov bx, 512
sal bx, 1
mul bx
mov di, ax
mov ax, [bp+BASE+8] ; x
sal ax, 1
add di, ax ; DI = destination offset = 2*x + ( y mod 64 ) * 2 * 512
mov si, [bp+BASE+4] ; point to value
push ds
mov ds, [bp+BASE+6]
mov ax, ds:[si]
pop ds
mov bx, [bp+BASE+12] ; mode
or bx, bx
jnz p1
mov word ptr es:[di], ax
jmp short p3

p1:
dec bx
jnz p2
or ax, 8000h
xor word ptr es:[di], ax
jmp short p3

p2:
and word ptr es:[di], ax

p3:
pop di
pop si
pop es
pop ds
mov sp, bp
pop bp
ret

```

`_PutPixel    endp`

-----  
;  
; `set_bank`

; `Description`

; `set_bank` sets the memory bank that corresponding to the row value.

; `input`

;    `ax`    `row`

; `output`

;    `ax`    `row mod 64`

; `destroy`

;    `ax, cx, dx`

-----  
`set_bank`    `proc near`

`push ax`

`mov cl, 5`

`sar ax, cl`           ; set the bank select

`and al, 00001110B`

`mov ah, al`           ; save in ah

`mov dx, _io_base`

`in al, dx`           ; get current mode register setting

`test al, 1`           ; is it 512 x 512 mode

`jz s1`

`and al, 11110001B`   ; yes, it is

`s1:`

`and al, 11111001B`   ; don't change mode

`or al, ah`

`out dx, al`

`pop ax`

`and ax, 00111111B`   ; mod 64

`ret`

`set_bank`    `endp`

`public _Isinsync`

`_Isinsync`   `proc far`

Apéndice C

```
    push bp
    mov  bp,sp
    push dx
    mov  dx, ds:_io_base
    in   al,dx
    and  al,01100000B
    cmp  al,00100000B
    je   _Isinsync10
    xor  ax,ax
_Isinsync10:
    pop  dx
    mov  sp, bp
    pop  bp
    ret
_Isinsync  endp

; /* ..... */
; /* local data definition */
; /* ..... */

; following data must be within CS segment
    tax  dw  ?

_TEXT  ends

end
```

### Programa 3

```
/* ***** */
/* Programa que usa las rutinas de 'C' y ensamblador para el */
/* manejo de la tarjeta PIB. Estos programas estan incluidos */
/* en el software de la tarjeta */
/*
/* El codigo que se presenta a continuacion hace uso de las */
/* rutinas del los programa imgio.asm y pibtool.c */
/*
/* Realizo: Diana Anabel Hernandez Alonso */
/* Gabriela de Jesus Magallanes Gonzalez */
/*
/* julio/93 */
/*
/* ***** */
```

```
#include <stdio.h>
#include <dos.h>
#include "c:pibtool.c"
```

```
main()
```

```
/* ----- inicializa la tarjeta PIB ----- */
```

```
InitPIB(0);
```

```
SetSource(0);
```

```
SetScreen(0);
```

```
SetPIBorEGA(1);
```

```
SetInDispMode();
```

```
SetUnderScan(0,0);
```

```
/* ----- hacer el recuadro en la T.V. ----- */
```

*Apéndice C*

```
FillPibRect(BLUE,0,0,150,250);  
FillPibRect(BLUE,350,0,550,250);  
FillPibRect(BLUE,0,0,550,50);  
FillPibRect(BLUE,0,200,550,250);
```

```
/* ----- inicializa la camara de video ----- */
```

```
SetMixMode();
```

```
SetBlank(1);
```

```
/* ----- espera a que este listo el usuario --- */
```

```
printf("Cuando estes listo presiona ENTER \n");  
getch();
```

```
SetCamera(0);
```

```
CaptureFrame();
```

```
SetExtDispMode();
```

```
/* ----- almacena la imagen en un archivo ----- */
```

```
system("\\PIB\\pibfileT=4F=PCXPCX30=SX1=150Y1=50X2=350Y2=200");
```

```
printf("Eso es todo,\n presiona cualquier tecla para continuar ....");  
getch();  
}
```

# **Apéndice**

## **D**

### **Programas Temporizadores**



## Apéndice D

```
jnz salida          ;diferente de cero? salida
mov cs:[contador],TIME ;reset al contador

push ax             ;salvamos registros
push bx
push cx
push dx
push ds
push es
push ss
push si
push di
push sp
push bp
pushf

sti                 ;enable interrupts
mov ah,2Ch
int DOS             ;obtenemos la hora del sistema
cmp cl,MINUTOS
jne fuera           ;si el minuto != MINUTOS no se hace nada

mov ax,VIDEO
mov ds,ax
xor si,si           ;DS:SI -> B800:0
push cs
pop es
mov ax,offset buffer
mov di,ax           ;ES:DI -> CS:[Buffer]
mov cx,80
cid
rep movsw           ;salvamos la primera linea de la pantalla

mov ax,VIDEO        ;inicio del Video
mov es,ax
xor bx,bx
xor di,di
mov cx,len_mensaje ;desplegaremos el mensaje
put_char: mov al,cs:[mensaje + bx]
           mov es:[di],al ;caracter por caracter
           mov byte ptr es:[di + 1],COLOR
           inc bx
```

*Diseño y construcción de un sistema de captura de imágenes, y control y asignación de equipo.*

```
    add di,2
    loop put_char

    mov cx;90          ;se hace una pausa
otra_vez:  push cx
           mov cx,OFFFh
retardo:  loop retardo
           pop cx
           loop otra_vez

           push cs
           pop ds
           mov ax,offset buffer
           mov si,ax          ;DS:SI -> CS:[Buffer]
           mov ax,VIDEO
           mov es,ax
           xor di,di         ;ES:DI -> B800:0
           mov cx,80
           cld
           rep movsw        ;restauramos la pantalla

fuera:    popf              ;restauramos registros
           pop bp
           pop sp
           pop di
           pop si
           pop ss
           pop es
           pop ds
           pop dx
           pop cx
           pop bx
           pop ax

salida:   iret              ;...y salimos

inicializa: mov ax,351Ch    ;obtenemos el vector original
            int DOS
            mov cs:[orig_intr],bx
            mov cs:[orig_intr+2],es

            mov ax,251Ch    ;cambiamos el vector de la interrupcion 1C
```



# **Apéndice**

## **E**

### **Hojas Técnicas**

## APENDICE E

# MOTOROLA SEMICONDUCTOR TECHNICAL DATA

## 6-Pin DIP Optoisolators Triac Driver Output

These devices consist of gallium-arsenide infrared emitting diodes, optically coupled to silicon bilateral switch and are designed for applications requiring isolated triac triggering, low-current isolated ac switching, high electrical isolation (to 7500 V peak), high detector standoff voltage, small size, and low cost.

- UL Recognized File Number 54915
- VDE approved per standard 0883/6.80 (Certificate number 41853), with additional approval to DIN IEC380/VDE0806, IEC435/VDE0805, IEC65 VDE0860, VDE 110b, covering all other standards with equal or less stringent requirements, including IEC204/VDE0113, VDE0160, VDE0832, VDE0833, etc.
- Special lead form available (add suffix "T" to part number) which satisfies VDE0883/6.80 requirement for 8 mm minimum creepage distance between input and output solder pads.
- Various lead form options available. Consult "Optoisolator Lead Form Options" data sheet for details.

**MOC3009  
MOC3010  
MOC3011  
MOC3012**

**6-PIN DIP  
OPTOISOLATORS  
TRIAC DRIVER OUTPUT  
250 VOLTS**



CASE 730A-02  
PLASTIC

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

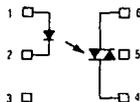
Rating	Symbol	Value	Unit
<b>INFRARED EMITTING DIODE</b>			
Reverse Voltage	$V_R$	3	Volts
Forward Current — Continuous	$I_F$	60	mA
Total Power Dissipation $\ast T_A = 25^\circ\text{C}$ Negligible Power in Transistor Derate above $25^\circ\text{C}$	$P_D$	100	mW
		1.33	mW/°C
<b>OUTPUT DRIVER</b>			
Off-State Output Terminal Voltage	$V_{DRM}$	250	Volts
Peak Repetitive Surge Current (PW 1 ms, 120 pps)	$I_{TSM}$	1	A
Total Power Dissipation $\ast T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	300	mW
		4	mW/°C

### TOTAL DEVICE

Isolation Surge Voltage (1) (Peak ac Voltage, 60 Hz, 5 Second Duration)	$V_{ISO}$	7500	$V_{ac}$
Total Power Dissipation $\ast T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	330	mW
		4.4	mW/°C
Junction Temperature Range	$T_J$	40 to 100	°C
Ambient Operating Temperature Range	$T_A$	40 to 85	°C
Storage Temperature Range	$T_{stg}$	-40 to +150	°C
Soldering Temperature (10 s)		260	°C

(1) Isolation surge voltage,  $V_{ISO}$ , is an effective surge detector breakdown voltage.

### COUPLER SCHEMATIC

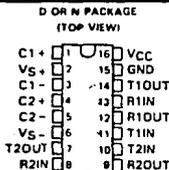


- 1 ANODE
- 2 CATHODE
- 3 NC
- 4 MAIN TERMINAL
- 5 SUBSTRATE  
DO NOT CONNECT
- 6 MAIN TERMINAL

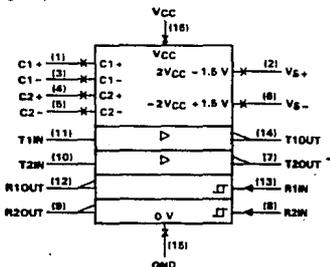
## MAX232 DUAL EIA-232 DRIVER/RECEIVER

DS120 FEBRUARY 1989—REVISED JUNE 1989

- Operates with Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typ
- Meets ANSI/EIA-232-D-1986 Specifications (Revision of EIA Standard RS-232-C)
- Designed to be Interchangeable with Maxim MAX232
- Applications
  - EIA-232 Interface
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers



logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

description

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.6 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Input supply voltage, VCC (see Note 1)	−0.3 V to 6 V
Positive output supply voltage, VS+	VCC − 0.3 V to 15 V
Negative output supply voltage, VS−	0.3 V to −16 V
Input voltage range: Driver	−0.3 V to VCC + 0.3 V
Receiver	±30 V
Output voltage range: T1OUT, T2OUT	VS− − 0.3 V to VS+ + 0.3 V
R1OUT, R2OUT	−0.3 V to VCC + 0.3 V
Short-circuit duration: VS+	30 s
VS−	30 s
T1OUT, T2OUT	unlimited
Operating free-air temperature range	0°C to 70°C
Storage temperature range	−65°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to network ground terminal.

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.

PHOTODUPLICATION DATA documents contain information derived on all publications data. Products undergo no modification for the purpose of Texas Instruments' standard terminology. Reproduction proceeding does not necessarily include testing of all parameters.



POST OFFICE BOX 664303 • DALLAS, TEXAS 75265

Copyright © 1989, Texas Instruments Incorporated

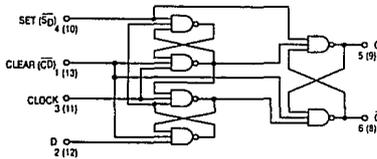


## DUAL D-TYPE POSITIVE EDGE-TRIGGERED FLIP-FLOP

The SN54/74LS74A dual edge-triggered flip-flop utilizes Schottky TTL circuitry to produce high speed D-type flip-flops. Each flip-flop has individual clear and set inputs, and also complementary Q and  $\bar{Q}$  outputs.

Information at input D is transferred to the Q output on the positive-going edge of the clock pulse. Clock triggering occurs at a voltage level of the clock pulse and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the HIGH or the LOW level, the D input signal has no effect.

LOGIC DIAGRAM (Each Flip-Flop)



MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	$\bar{S}_D$	$\bar{C}_D$	D	Q	$\bar{Q}$
Set	L	H	X	H	L
Reset (Clear)	H	L	X	L	H
Undetermined	L	L	X	H	H
Load "1" (Set)	H	H	h	H	L
Load "0" (Reset)	H	H	l	L	H

\* Both outputs will be HIGH while both  $\bar{S}_D$  and  $\bar{C}_D$  are LOW, but the output states are unpredictable if  $\bar{S}_D$  and  $\bar{C}_D$  go HIGH simultaneously. If the levels at the set and clear are near  $V_{IH}$ , maximum then we cannot guarantee to meet the minimum level for  $V_{OH}$ .

H, h = HIGH Voltage Level

L, l = LOW Voltage Level

X = Don't Care

i, h (Q) = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the HIGH to LOW clock transition.

**SN54/74LS74A**

**DUAL D-TYPE POSITIVE  
EDGE-TRIGGERED FLIP-FLOP**  
LOW POWER SCHOTTKY



**J SUFFIX  
CERAMIC  
CASE 632-08**



**N SUFFIX  
PLASTIC  
CASE 646-06**

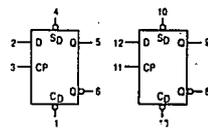


**D SUFFIX  
SOIC  
CASE 751A-02**

**ORDERING INFORMATION**

SN54LSXXJ Ceramic  
SN74LSXXN Plastic  
SN74LSXXD SOIC

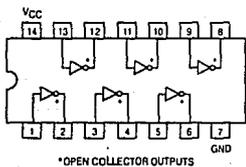
**LOGIC SYMBOL**



VCC = PIN 14  
GND = PIN 7



# HEX INVERTER



\*OPEN COLLECTOR OUTPUTS

**SN54/74LS05**

**HEX INVERTER  
LOW POWER SCHOTTKY**



J SUFFIX  
CERAMIC  
CASE 632-08



N SUFFIX  
PLASTIC  
CASE 646-08



D SUFFIX  
SOIC  
CASE 751A-02

**ORDERING INFORMATION**

SN54LSXXJ Ceramic  
SN74LSXXN Plastic  
SN74LSXXD SOIC

**GUARANTEED OPERATING RANGES**

Symbol	Parameter		Min	Typ	Max	Unit
VCC	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
TA	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
V <sub>OH</sub>	Output Voltage — High	54, 74			5.5	V
I <sub>OL</sub>	Output Current — Low	54			4.0	mA
		74			8.0	



**MOTOROLA**

**Product Preview**

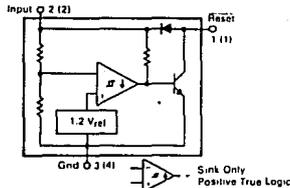
**UNDervoltage SENSING CIRCUIT**

The MC34064 is an undervoltage sensing circuit specifically designed for use as a reset controller in microprocessor-based systems. It offers the designer an economical solution for low-voltage detection with a single external resistor. The MC34064 features a trimmed-in-package bandgap reference, and a comparator with precise thresholds and built-in hysteresis to prevent erratic reset operation. The open collector reset output is capable of sinking in excess of 10 mA, and operation is guaranteed down to 1.0 volt input with low standby current. These devices are packaged in 3-pin TO-226AA and 8-pin surface mount packages.

Applications include direct monitoring of the 5.0 volt MPU logic power supply used in appliance, automotive, consumer and industrial equipment.

- Trimmed-in-Package Temperature Compensated Reference
- Precise Comparator Thresholds Guaranteed Over Temperature
- Comparator Hysteresis Prevents Erratic Reset
- Reset Output Capable of Sinking in Excess of 10 mA
- Internal Clamp Diode for Discharging Delay Capacitor
- Guaranteed Reset Operation with 1.0 Volt Input
- Low Standby Current
- Economical TO-226AA and Surface Mount Packages

**REPRESENTATIVE BLOCK DIAGRAM**



Pin numbers adjacent to terminals are for the 3-pin TO-226AA package  
Pin numbers in parentheses are for the D suffix SO-8 package.

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

**MC34064  
MC33064**

**UNDervoltage SENSING CIRCUIT  
SILICON MONOLITHIC INTEGRATED CIRCUIT**

P SUFFIX  
PLASTIC PACKAGE  
CASE 23-04



PIN 1. RESET  
2. INPUT  
3. GROUND

D SUFFIX  
PLASTIC PACKAGE  
CASE 751-02  
SO-8



PIN 1. RESET      5. N.C.  
2. INPUT          6. N.C.  
3. N.C.            7. N.C.  
4. GROUND        8. N.C.

**ORDERING INFORMATION**

Device	Temperature Range	Package
MC34064D-5		Plastic SO-8
MC34064P-5	0°C to +70°C	Plastic TO-226AA
MC33064D-5		Plastic SO-8
MC33064P-5	-40°C to +85°C	Plastic TO-226AA

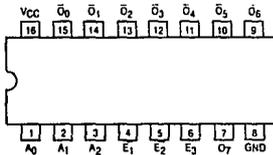


## 1-OF-8 DECODER/ DEMULTIPLEXER

The LSTTL/MSI SN54/74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Demultiplexing Capability
- Multiple Input Enable for Easy Expansion
- Typical Power Dissipation of 32 mW
- Active Low Mutually Exclusive Outputs
- Input Clamp Diodes Limit High Speed Termination Effects

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE  
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package

PIN NAMES

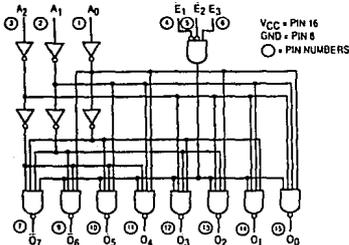
- A<sub>0</sub>-A<sub>2</sub> Address Inputs
- E<sub>1</sub>, E<sub>2</sub> Enable (Active LOW) Inputs
- E<sub>3</sub> Enable (Active HIGH) Input
- O<sub>0</sub>-O<sub>7</sub> Active LOW Outputs (Note b)

LOADING (Note a)

HIGH	LOW
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.

- NOTES:  
a) 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.  
b) The Output LOW drive factor is 2.5 U.L. for Metlay (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC DIAGRAM



## SN54/74LS138

### 1-OF-8 DECODER/ DEMULTIPLEXER LOW POWER SCHOTTKY



J SUFFIX  
CERAMIC  
CASE 620-09



N SUFFIX  
PLASTIC  
CASE 648-08

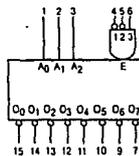


D SUFFIX  
SOIC  
CASE 751B-03

ORDERING INFORMATION

- SN54LSXXXJ Ceramic
- SN74LSXXXN Plastic
- SN74LSXXXD SOIC

LOGIC SYMBOL



VCC = PIN 15  
GND = PIN 8

## **BIBLIOGRAFIA**

- DIGITAL IMAGE PROCESING** 1985  
Catlemman-Kenneth  
Prentice -Hall  
Cap. 1,2 y3
- COMPUTER & GRAPHICS** 1989  
Vol. 13  
AT&T's image capture board (ICB)
- IEEE SPECTRUM** march 1992  
Digital Video  
Pags. 24-30
- SISTEMAS DE COMUNICACIONES** 1989  
Strembley  
Alfa-Omega  
Cap. 1,2 y 5
- MICRO MONITOR** abril 1992  
El monitor de la computadora  
Pags. 21-24
- BYTE** february 1990  
Firts Looks  
Pags. 33-35
- BYTE** december 1989  
Images  
Pags. 248-256

<b><u>BYTE</u></b> Video Computing Pags. 64-80	September 1993
<b><u>AV Video</u></b> Computer Graphics Pags. 54-62	may 1993
<b><u>BYTE</u></b> Smile for the computer Pags. 139-144	november 1992
<b><u>HC11</u></b> Reference manual	1989
<b><u>EGC</u></b> Manual de refencia	1990
<b><u>Paradox</u></b> Manual de usuario	1993
<b><u>CODIGO DE BARRAS</u></b> Mc Graw Hill	1990
<b><u>ASSEMBLY LENGUAJE</u></b> <b><u>80386/80286</u></b> Murray Prntice Hall	1990