ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

DISEÑO Y CONSTRUCCIÓN DE UN GENERADOR DE CARACTERES A PARTIR DE UNA FILA DE LEDS GIRATORIA

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO INGENIERO MENCIÓN ELECTRÓNICA Y TELECOMUNICACIONES

JORGE PAÚL GARCÍA NAVAS

DIRECTOR: Ing. FERNANDO VÁSQUEZ

Quito, Octubre 2003

DECLARACIÓN

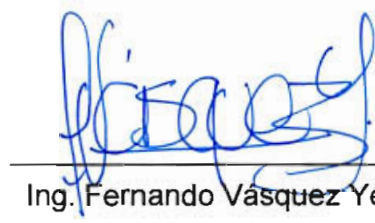
Yo, Jorge Paúl García Navas, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido en la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad Institucional vigente.

Jorge Paúl García Navas

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Jorge Paúl García Navas, bajo mi supervisión.



Ing. Fernando Vásquez Yépez

Director de proyecto

AGRADECIMIENTO

Gracias a Dios por permitirme cumplir esta meta, a mis Padres por que sin ellos no hubiese sido posible realizar mi carrera y a Pamela por todo el amor y el apoyo que me ha dado.

Al Ing. Fernando Vásquez por su valiosa guía e incondicional amistad.

A todas las personas que de alguna manera han contribuido para la realización de este trabajo.

DEDICATORIA

A mis padres y a Pamela quienes han estado siempre a mi lado apoyándome incondicionalmente y han sido parte fundamental para la realización de este objetivo.

CONTENIDO:

Página

CAPITULO 2

CAPITULO 3

INTRODUCCIÓN

En la actualidad los dispositivos visuales son muy utilizados para transmitir ideas o mensajes, debido a que se puede captar la atención de las personas con mayor facilidad utilizando este tipo de medios. Por esta razón, los tableros electrónicos han tomado auge y son objeto de constante cambio y desarrollo para obtener dispositivos cada vez más sofisticados a menor costo.

En el presente trabajo se ha desarrollado un generador de caracteres construido con diodos emisores de luz que permite visualizar mensajes de texto de una manera muy novedosa, utilizando una sola fila de LEDs giratoria.

El sistema de control y generación de caracteres del sistema implementado se ha realizado utilizando un microcontrolador cuya arquitectura es totalmente compatible con la de la familia MCS51 pero en un encapsulado de 20 pines.

Al no existir ninguna referencia sobre un dispositivo similar al propuesto en este proyecto de titulación, en este trabajo se describe detalladamente el proceso de construcción y las consideraciones de hardware y software necesarias para implementar un sistema de generación de mensajes. Adicionalmente se ha desarrollado un software para implementar un programa que desde un computador personal permite editar y transmitir los mensajes que se requieran mostrar en el generador de caracteres.

RESUMEN

El presente proyecto de titulación describe el diseño y construcción de un dispositivo que permite generar caracteres utilizando una sola fila de LEDs giratoria. Para lograrlo, se ha implementado un sistema microprocesado basado en el microcontrolador AT89C2051 de ATMEL cuya arquitectura es compatible a la de los microcontroladores de la familia MCS51 de INTEL.

En el Capítulo 1 se resumen los conceptos generales y los términos utilizados para la realización de este trabajo, así como el estado actual de la tecnología para la visualización de mensajes utilizando tableros electrónicos con LEDs.

En el Capítulo 2 se detalla el diseño del generador de caracteres describiendo las alternativas que se utilizaron y las consideraciones de hardware y software necesarias para la implementación del generador.

El Capítulo 3 describe minuciosamente la implementación del hardware y software descrito en el capitulo anterior, así como también un programa que se ejecuta en un computador personal el cual permite editar y transmitir los mensajes al generador. Además se explica paso a paso los aspectos concernientes a la construcción de cada una de las partes que conforman el generador de caracteres.

En el capítulo 4 se explican las pruebas realizadas, los resultados obtenidos y las posibles aplicaciones del generador de caracteres.

Finalmente en el capítulo 5 se mencionan las conclusiones y recomendaciones que surgieron luego de implementar este proyecto de titulación.

CAPITULO 1

CONCEPTOS GENERALES

En este capítulo se resumen los conceptos y términos que se van a utilizar en el presente proyecto de titulación, así como también se describe el estado actual de la tecnología utilizada en la construcción de tableros electrónicos con diodos emisores de luz (LEDs).

1.1 IMAGEN

Las imágenes son el resultado de las variaciones de la intensidad luminosa sobre una superficie, sin embargo, la luz no es el único parámetro del cual se puede obtener imágenes. Por ejemplo, una imagen puede formarse con la información de señales tales como las variaciones de temperatura de un circuito integrado, la velocidad de la sangre en la arteria de una persona, la transmisión de rayos x de una galaxia lejana o el movimiento de la tierra durante un terremoto. Estas imágenes pueden ser evaluadas por el ojo humano al traducirlas a imágenes convencionales variando la intensidad luminosa.

1.1.1 *BRILLO*

El brillo se refiere a cuán clara u oscura es una imagen. Cuando el brillo es muy alto, las partes más claras de la imagen están saturadas destruyendo los detalles de la imagen en estos lugares, en cambio, si el brillo es demasiado bajo, las partes más negras de la imagen son las que se saturan.

1.1.2 *CONTRASTE*

Es la diferencia del brillo entre objetos o regiones. Esto influye significativamente al apreciar una imagen puesto que al incrementar el contraste las partes claras se tornan más claras y las partes oscuras se tornan más oscuras

1.1.3 *PÍXEL*

La palabra píxel es una contracción de la frase *picture element*, es una muestra de un punto de luz que constituye una imagen y conforma un arreglo de dos dimensiones estructurado en filas y columnas.

El valor de cada píxel en una imagen digital representa una pequeña región de esa imagen siendo digitalizada. Por ejemplo, si se toman una muestra del planeta Venus cada 10 metros de su superficie, entonces se esta definiendo un espacio cuadrado de muestreo donde cada píxel representa un área de 10 x 10 metros como se muestra en la figura 1.1.

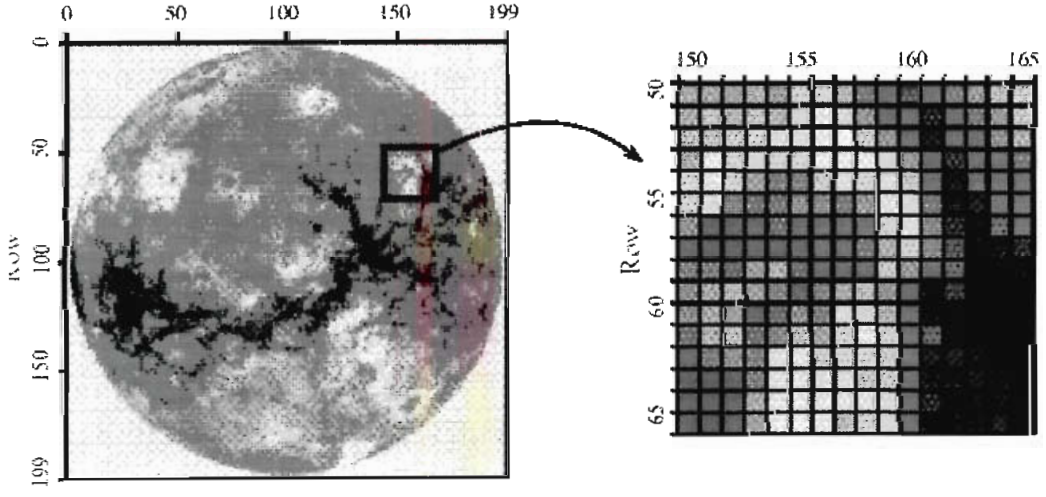


Figura 1.1. Imagen del planeta Venus en donde cada píxel representa 10m² de superficie[1]

[1] Ejemplo Tomado de The Scientist and Engineer's Guide to Digital Signal Processing, SMITH, Steven, Capitulo 23: Image Formation & Display.

1.2 ESTADO ACTUAL DE LA TECNOLOGÍA PARA LA VISUALIZACIÓN DE MENSAJES.

1.2.1 *CARACTERISTICAS DE LOS TABLEROS ELECTRONICOS*

Los Tableros Electrónicos han tenido gran aceptación para visualización tanto de texto como de animaciones, gráficos y video. Las características de cada tablero dependen de las necesidades del usuario. A continuación se presenta las características que deben tomarse en cuenta para utilizar un tablero electrónico.

- Las condiciones externas del medio ambiente en que va a funcionar el tablero. Si el tablero va a ser utilizado en exteriores deberá tener las protecciones necesarias para soportar los cambios de clima y las características de brillo deberán ser mayores para que pueda verse a la luz del día.

- La resolución en pixels del tablero, esto es, el número de pixels horizontales por el número de pixels verticales. Por ejemplo, una resolución de 16 x 64 pixels significa que el tablero tiene 16 pixels horizontalmente por 64 pixels verticalmente. Una imagen digital típica está compuesta de alrededor de 500 filas por 500 columnas. Esta es la calidad de imagen que se encuentra en televisión y en aplicaciones de computadores personales. Las imágenes compuestas por un número menor de pixels, por ejemplo 250 x 250, se dice que tienen una resolución baja, ya que a menudo pueden distinguirse los pixels disminuyendo la calidad de la imagen, o el espacio en la pantalla es demasiado corto y no pueden mostrarse imágenes completas. Por otro lado, las imágenes con una resolución mayor a 1000 x 1000 pixels se consideran excepcionalmente buenas. Mientras crece el número de pixels, crece el número de datos que deben manejarse, obligando a utilizar equipos capaces de manejar estos datos a una velocidad mayor.

- El tipo de información que se va a visualizar, la cual puede ser solo texto, gráficos, animaciones o video. Las aplicaciones de solo texto pueden ser

desarrolladas con solamente una fila de caracteres de por lo menos 7 pixels de altura y el número de pixels de ancho depende del numero de caracteres que se quiera mostrar en una pantalla. Las aplicaciones de gráficos, animaciones o video requieren de un número mayor de filas, es decir, se requiere un mayor número de pixels a lo ancho de la pantalla para poder mostrar gráficos.

- Si la aplicación es de texto, una característica importante es el número de pixels que forman un caracter, generalmente un carácter se compone de una matriz de 5 x 7 pixels. Se pueden utilizar otras configuraciones, pero para obtener caracteres bien definidos por lo menos se debe tener una resolución de 5x7.

- El color del tablero, el cual puede ser monocromático o de varios colores. De cada color, pueden obtenerse varios niveles de cuantización que dan lugar a diferentes tonalidades del mismo. En los tableros de LED's, estas tonalidades se obtienen al hacer circular varios niveles de corriente por cada LED. Generalmente, se utilizan hasta 256 niveles de color o tonalidades por pixel. Existen varias razones para ello, primero, un byte es conveniente para el manejo de los datos debido a que usualmente los computadores utilizan esta unidad para almacenar su información, segundo, el alto número de pixels que conforman una imagen justifica el que deba limitarse el número de pasos de cuantización del color y tercero, y mas importante, un paso de 1/256 es mayor de lo que el ojo humano puede percibir, es decir, para el ojo humano una imagen no mejorará si se usa mas de 256 niveles de cuantización del color. Si se combinan en un pixel las distintas tonalidades de dos o los tres colores básicos (rojo, verde y azul) puede obtenerse una combinación de hasta 16.7 millones de colores.

- El método de comunicación con los tableros, lo cual servirá para actualizar la información que presentan los mismos. Generalmente se utiliza cable directo con RS232 o RS485, pero también se puede utilizar MODEM, receptores inalámbricos, fibra óptica o satélite.

- El ángulo de visibilidad del tablero es también un factor muy importante. Esto depende del ángulo de intensidad media de los LEDs que se utilicen. Los valores de ángulo de intensidad media de los LEDs brillantes, super brillantes y ultra brillantes, que son los que se utilizan en la construcción de tableros electrónicos, están entre los 15° y los 120°

A continuación se describen algunos términos relacionados con la construcción de tableros electrónicos:

1.2.2 *ESTRUCTURA DE UN SOLO CUERPO.*

En este tipo de estructuras, el tablero se construye en un solo gabinete. Generalmente las dimensiones de este tipo de estructuras son máximo de 6" por 13" (aproximadamente 15 x 33 cm).

Figura 1.2. Tablero construido de un solo cuerpo

1.2.3 *ESTRUCTURA MODULAR.*

En este tipo de estructuras, el producto final lo conforman un determinado número de módulos dispuestos horizontal y/o verticalmente para completar el tablero. Cada módulo es una matriz de LEDs y debido a que los circuitos digitales existentes manejan la información orientada al byte estas matrices pueden ser de 8x8, 16x16, 32x32, 8x16, 16x32, etc pixels.

Este tipo de tableros no tiene limitaciones en cuanto a dimensiones, los límites se establecen de acuerdo al tamaño que se necesite para el tablero y al controlador que actualiza los datos para el mismo.

a) b)

Figura 1.3. *a) Módulo de 16x16 pixels construido con clusters. b) Estructura de un tablero construido en forma modular.*

1.2.4 *PITCH*

Es el espacio entre pixels, definido como la distancia física entre los centros de dos pixels adyacentes.

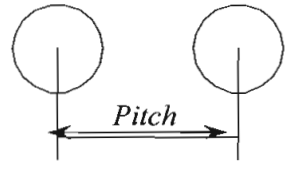


Figura 1.4. *Definición de Pitch*

Los valores de *pitch* utilizados en la construcción de tableros electrónicos varían de acuerdo al fabricante y a la aplicación, así, en tableros que van a utilizarse en interiores, se utilizan valores de pitch de 6,8,12.5,15,16, 20 y 22 mm. En cambio en tableros que van a usarse en exteriores se utilizan valores de pitch de 25,39,40,52 mm.

1.2.5 *MATRIZ DE CARACTERES.*

Es un conjunto de pixels dispuestos en filas y columnas en los cuales pueden mostrarse caracteres alfanuméricos.

1.2.6 *CARACTERES POR LÍNEA.*

Se refiere al máximo número de caracteres que pueden pintarse en una sola línea. Por lo general un carácter se forma en una matriz de 5x7 pixels, por lo tanto, una fila tendrá tiene 7 pixels de ancho.

Figura 1.5. Tablero de una línea de caracteres

1.2.7 *ALTURA DE CARACTER.*

Se refiere a la altura del carácter producido en una matriz. De la altura del carácter depende la distancia máxima a la que puede verse el mismo. Los fabricantes de tableros electrónicos mencionan que la distancia a la que puede ser visto un carácter aumenta 50 pies (1 pie = 30,48 centímetros) por cada pulgada (1 pulgada = 2,54 centimetros) de altura de un carácter.

1.2.8 *TABLEROS ELECTRONICOS CON LEDS EXISTENTES EN EL MERCADO*

Las principales aplicaciones de los tableros electrónicos son la publicidad y el deporte.

En la figura 1.6 se muestran tableros electrónicos con LEDs de un solo color. Los tableros de las figuras 1.16 a) y b) son tableros alfanuméricos utilizados para el deporte y locales comerciales respectivamente. La figura 1.16 c) muestra un tablero gráfico de dos colores (rojo y negro) y en la figura 1.16 d) se tiene un tablero gráfico con control de intensidad del color.

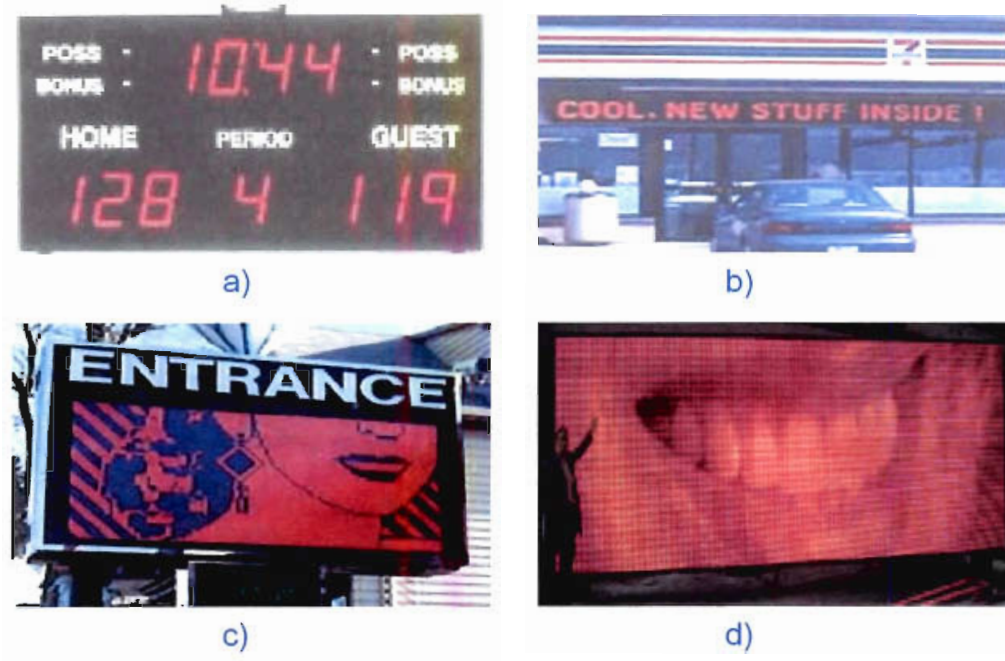


Figura 1.6. *Tableros con LEDs de un solo color, a) tablero numérico para deportes, b) tablero alfanumérico para locales comerciales, c) tablero gráfico sin control de intensidad del color, d) tablero gráfico con control de intensidad del color.*

Los tableros electrónicos con dos colores de LEDs se ilustran el la figura 1.7. La figura 1.17 a) muestra un tablero alfanumérico de cuatro colores (rojo, verde, amarillo y negro) formados a partir las combinaciones de los colores rojo y verde. En la figura 1.17 b) se observa un tablero gráfico de cuatro colores, y en la figura 1.17 c) se tiene un tablero electrónico utilizado para aplicaciones de video y cuyos colores se forman a partir de la combinación de las tonalidades de los LEDs de color rojo y verde.

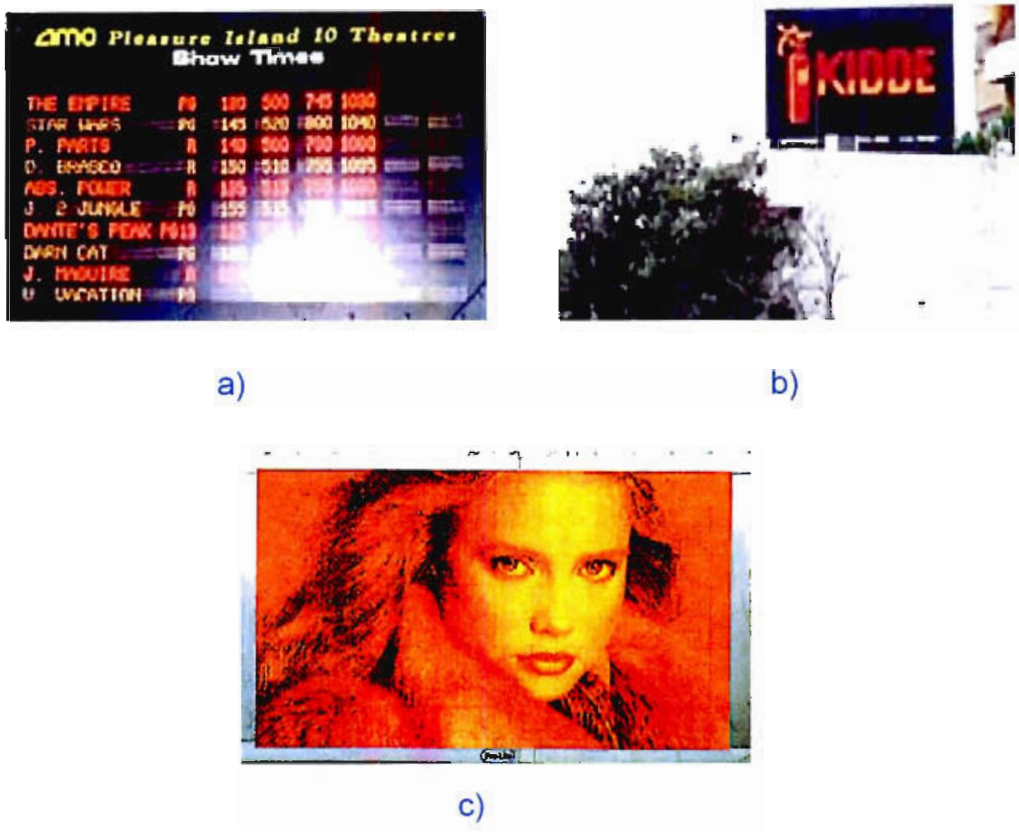
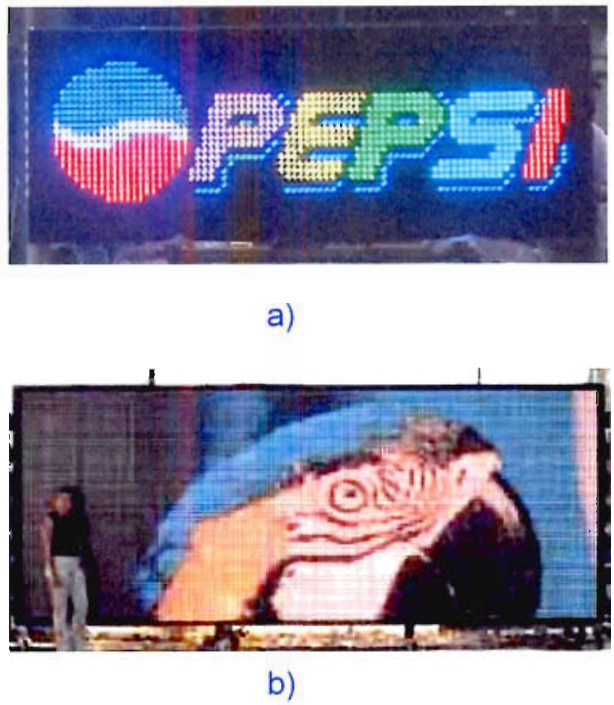


Figura 1.7. *Tableros con dos colores de LEDs y sus combinaciones, a) tablero alfanumérico, b) tablero gráfico sin control de intensidad del color, c) tablero gráfico con control de intensidad del color.*

En la figura 1.8 se pueden observar tableros electrónicos con tres colores de LEDs (rojo, verde y azul), que son los colores básicos a partir de los cuales se puede obtener todos los colores restantes. En la figura 1.8 a) se observa un tablero electrónico de 8 colores, resultado de la combinación de los tres colores básicos sin control de las tonalidades. Al controlar las tonalidades de los tres colores básicos se pueden obtener tableros como el de la figura 1.8 b), el cual se utiliza para video a todo color.



a)

b)

Figura 1.8. *Tableros con tres colores de LEDs y sus combinaciones (Full Color), a) tablero gráfico de 8 colores, b) tablero para video a todo color.*

1.2.9 *OTRAS TECNOLOGIAS*

Además de los LED´s se utilizan otras tecnologías en la construcción de tableros electrónicos, como son las tecnologías de lámpara halógena, tubo fluorescente, tubo de rayos catódicos y la tecnología electromagnética. La ventaja de los LED´s sobre estas tecnologías es su bajo costo de operación y mantenimiento, ya que permiten tener estructuras de bajo peso, mayor tiempo de vida útil (alrededor de 50000 horas vs. 10000 horas de los tableros de tubos de rayos catódicos y lámparas fluorescentes) y bajo consumo de potencia. Además permiten mayores resoluciones utilizando menos espacio y su costo es mucho menor. Uno de los problemas es el brillo, pero con los avances de la tecnología se tiene ahora LED´s brillantes, super brillantes y ultra brillantes, los cuales posibilitan superar este problema.

A continuación se presentan figuras de tableros hechos con estas tecnologías alternativas:

a)

b)

c)

Figura 1.9. *Tecnologías alternativas para la construcción de tableros electrónicos, a) Lámpara Halógena, b) Tubo de Descarga de Gas, c) Tecnología Electromagnética*

1.3 MAGNITUDES Y UNIDADES DE MEDIDA UTILIZADAS EN LA EVALUACION DE FUENTES DE LUZ

La luz, al igual que las ondas de radio, los rayos X o los gamma es una forma de energía. En el Sistema Internacional la energía se mide en joules (J), pero debido a que no toda la luz emitida por una fuente llega al ojo y produce sensación luminosa, ni toda la energía que consume se convierte en luz es que se utilizan unidades diferentes, Las magnitudes que se utilizan para medir los distintos parámetros de la luz son: flujo luminoso, intensidad luminosa, iluminancia, luminancia y rendimiento o eficiencia luminosa.

1.3.1 *FLUJO LUMINOSO*

Se define el flujo luminoso como la potencia (W) emitida en forma de radiación luminosa a la que el ojo humano es sensible. Su símbolo es Φ y su unidad es el lumen (lm). A la relación entre watios y lúmenes se le llama equivalente luminoso de la energía y equivale a:

1 watio-luz a 555 nm = 683 lm

Por ejemplo, si se consideran dos luces incandescentes, una de 25 W y otra de 60 W, se podrá obtener una luz más intensa con la de 60 W. Estos valores se refieren a la potencia consumida por la bombilla de la cual solo una parte se convierte en luz visible, la cual se denomina flujo luminoso. Se podría medirlo en watts (W), pero es más sencillo definir una nueva unidad, el lumen, que tome como referencia la radiación visible. Empíricamente se demuestra que a una radiación de 555 nm de 1 W de potencia emitida por un cuerpo negro[1] le corresponden 683 lumen.

[1] Cuerpo Negro.- Sistema ideal que absorbe toda la radiación que incide sobre él

1.3.2 *INTENSIDAD LUMINOSA*

Se conoce como intensidad luminosa (I) al flujo luminoso emitido (Φ) por unidad de ángulo sólido (ω) y tiene por objeto caracterizar la emisión de luz en una dirección concreta. Se define por la ecuación:

$$I = \frac{\Phi}{\omega} \qquad \textit{(Ec. 1.1)}$$

Su unidad es la candela (cd), que se define como la intensidad luminosa en una dirección dada, de una fuente que emite una radiación monocromática de frecuencia $540x10^{12}$ hertz, equivalente a 555 nm en el vacío, y de la cual la intensidad radiada en esa dirección es (1/683) watt por estereorradián.

En la figura 1.10. se pueden observar los parámetros que intervienen en la ecuación 1.1, el flujo luminoso de la fuente de luz incide en una esfera en la que se miden las unidades de ángulo sólido en estereo radianes.

Figura 1.10. Definición de Intensidad Luminosa

La intensidad luminosa es uno de los parámetros utilizados para evaluar el nivel de brillo de un LED, mientras se pueda obtener un mayor nivel de intensidad luminosa con la misma corriente pasando por el LED , éste será de mejor calidad. Esta medida se realiza en el eje de 0°.

La diferencia entre flujo luminoso e intensidad luminosa se puede explicar utilizando la figura 1.11. El flujo luminoso nos da una idea de la cantidad de luz

que una fuente luminosa emite en todas las direcciones del espacio. Pero si pensamos en un proyector es fácil ver que sólo ilumina en una dirección. Por lo tanto, también se debe conocer cómo se distribuye el flujo en cada dirección del espacio y para eso se define la intensidad luminosa.

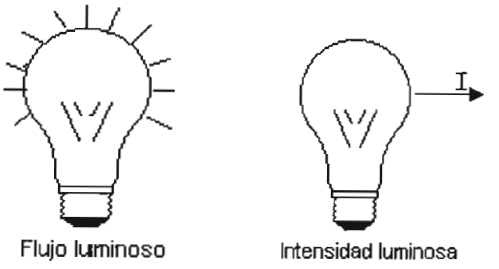


Figura 1.11. Diferencia entre flujo e intensidad luminosa.

1.3.3 *ILUMINANCIA*

Se define iluminancia como el flujo luminoso (Φ) recibido por una superficie (S). Esta relación se indica en la ecuación 1.2, su símbolo es E y su unidad el lux que es un lm/m2.

$$E = \frac{\Phi}{S} \qquad \textit{(Ec. 1.2)}$$

Existe también otra unidad, el *foot-candle* (fc), utilizada en países de habla inglesa cuya relación con el lux es:

$$1 \text{ fc} \simeq 10 \text{ lux}$$
$$1 \text{ lux} \simeq 0.1 \text{ fc}$$

Mediante el ejemplo indicado en la figura 1.12 se puede explicar el concepto de luminancia. Al iluminar con una linterna una pared a diferentes distancias, se puede observar que si se ubica la linterna cerca de la pared ésta se

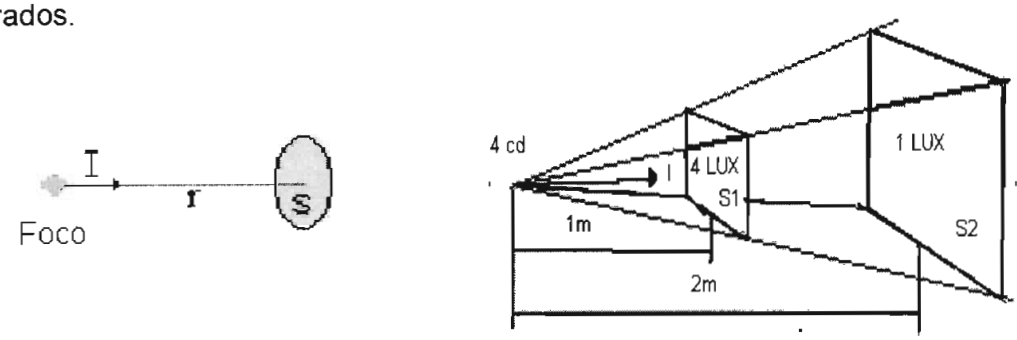
encuentra fuertemente iluminada por un círculo pequeño, en cambio, mientras se aleja la linterna, se puede notar que el círculo se hace más grande y la luz débil.

Figura 1.12. Definición de Iluminancia[1]

Por lo tanto iluminancia depende de la distancia del foco al objeto iluminado y esta relación de dependencia viene dada por la ley inversa de los cuadrados que relaciona la intensidad luminosa (I) y la distancia a la fuente (r). De esta forma, el cálculo de la iluminancia se puede realizar también utilizando la relación:

$$E = \frac{I}{r^2} \qquad \textit{(Ec 1.3)}$$

Esta ley solo es válida si la dirección del rayo de luz incidente es perpendicular a la superficie. En la figura 1.13 se ilustra la ley inversa de los cuadrados.



a) b)

Figura 1.13. Ley inversa de los Cuadrados[2]*. a) Variables que intervienen, b) ejemplo numérico*

[1] Tomado de www.Edison.upc.es
[2] Tomado de: Tratado Práctico de Luminotecnia, FERRER, Ricardo, Primera Edición, 1934

1.3.4 *LUMINANCIA*

Se llama luminancia a la relación entre la intensidad luminosa (I) y la superficie aparente (S') proyectada sobre un plano perpendicular a la dirección vista por el ojo. Por lo tanto, la luminancia es la sensación luminosa, que por efecto de la luz, se produce en la retina del ojo humano y determina la impresión de mayor o menor claridad producida por una superficie. Su símbolo es L y su unidad es la cd/m^2. También es posible encontrar otras unidades como el stilb (1 sb = 1 cd/m^2) o el nit (1 nt = 1 cd/cm^2).

En la ecuación 1.4 se define el concepto de luminancia y las variables que intervienen se ilustran en la figura 1.14.

$$L = \frac{I}{S'} = \frac{I}{S.\cos\alpha} \qquad \textit{(Ec. 1.4)}$$

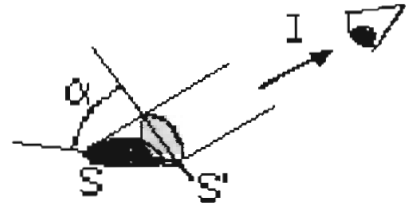


Figura 1.14. Concepto de Luminancia

1.3.5 *RENDIMIENTO LUMINOSO O EFICIENCIA LUMINOSA*

Como se mencionó anteriormente, no toda la energía eléctrica consumida por una lámpara (bombilla, fluorescente, etc.) se transforma en luz visible. Parte se pierde por calor y parte en forma de radiación no visible (infrarrojo o ultravioleta). En la figura 1.15 se muestran las principales pérdidas de la potencia consumida por una lámpara incandescente.

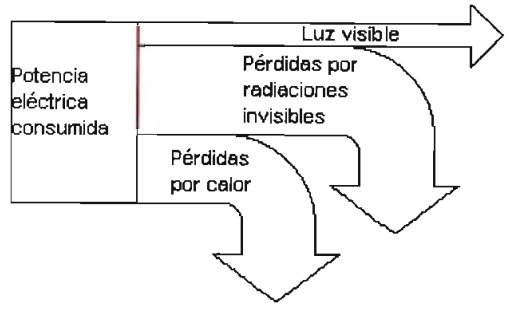


Figura 1.15. Pérdidas de potencia consumida por una lámpara incandescente

Se define el rendimiento luminoso como el cociente entre el flujo luminoso producido (Φ) y la potencia eléctrica consumida (W) indicado en la ecuación 1.5, y que viene con las características de las lámparas (25 W, 60 W). Mientras mayor sea el rendimiento mejor será la lámpara y menos potencia consumirá. La unidad es el lumen por watt (lm/W) y su símbolo es η .

$$\eta = \frac{\Phi}{W} \qquad \textit{(Ec. 1.5)}$$

1.4 DIODOS EMISORES DE LUZ (LED)

Un diodo emisor de luz (LED, siglas de *light emmiting diode*) es un dispositivo semiconductor que emite luz cuando una corriente eléctrica circula a través de él. La luz tiene una longitud de onda y por lo tanto un color, el cual puede ir desde el rojo (aproximadamente 700 nanómetros) hasta el azul-violeta (alrededor de 400 nanómetros). Algunos LEDs emiten luz infrarroja (830 nanómetros o más) y se conocen también con el nombre de diodos emisores de infrarrojo (IRED).

Un LED consta de dos elementos de material procesado llamado semiconductor tipo P y semiconductor tipo N. Estos dos elementos están en contacto directo, formando una región llamada juntura P-N. En este aspecto, el

LED no difiere de cualquier otro tipo de diodo, pero las diferencias radican en que los LEDs tienen una cubierta transparente que permite pasar la luz visible o infrarroja. Además, los LEDs poseen un área de juntura P-N mayor que los diodos normales cuya forma se adapta a su uso. La figura 1.16 muestra las partes básicas de que está compuesto un LED.

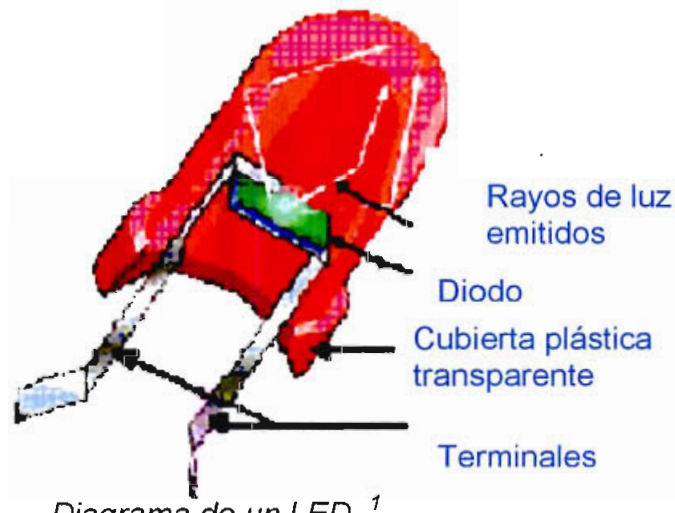


Figura 1.16. Diagrama de un LED [1]

Los LEDS tienen varias ventajas sobre las lámparas incandescentes y fluorescentes convencionales entre las cuales se puede mencionar:

- Elevada resistencia mecánica: Los LEDs son elementos 100% sólidos, resisten golpes y vibraciones mucho mejor que las lámparas incandescentes, las cuales están construidas con un filamento que con el tiempo puede destruirse.

- Larga vida útil: Hasta 100000 horas de vida útil comparado con 8000 horas de vida útil de una buena lámpara incandescente
- Reducido consumo de energía: Esta es la principal ventaja de los LEDs comparados con las lámparas incandescentes. En el caso de los LEDs, gran parte de la energía eléctrica es transformada en energía lumínica debido a que generan muy poco calor al producir

luz. Una lámpara fabricada con LEDs consume del 5 al 10% de la energía que consume una lámpara incandescente[2].

- Bajo tiempo de respuesta: Usualmente en el orden de los 0.1 microsegundos en comparación con mas de 20 milisegundos para las lámparas incandescentes. Esto permite utilizar los LEDs en forma multiplexada como en los displays alfanuméricos o en aplicaciones de telecomunicaciones por aire o por fibra óptica.

El usar LEDs para iluminar un ambiente resulta costoso, pero su alto costo se compensa con su durabilidad y bajo consumo de energía. Esta característica ha dado como resultado el que en los últimos años se ha comenzado a considerar a los LEDs como una muy buena opción para este tipo de aplicaciones.

En la tabla 1.1 se resumen las ventajas y desventajas de los LEDs respecto a las lámparas incandescentes.

	Tiempo de Vida Útil (Horas)	**Consumo de Energía**	**Costo**	**Resistencia Mecánica**	**Tiempo de Respuesta**
LEDs	100000	Bajo	Alto	Alta	0.1 µs
LAMPARAS INCANDESCENTES	8000	Alto	Bajo	Baja	20 ms

Tabla 1.1. Comparación entre LEDs y Lámparas incandescentes

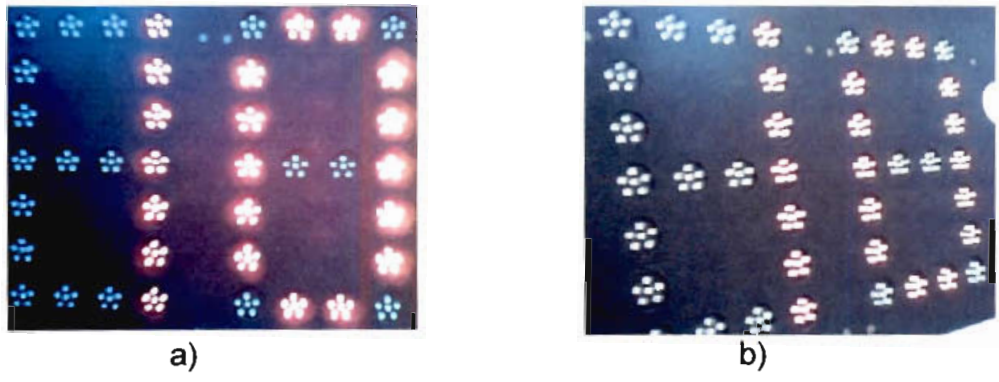
Los LEDs son muy utilizados para luces indicadoras, gráficos de barras o caracteres alfanuméricos, transmisión de datos en fibra óptica, control remoto, opto acopladores, etc. A continuación se describen algunos de los parámetros utilizados para evaluar LEDs.

---

[1] Tomado de www.lightemittingdiodes.com
[2] Datos tomados de Advanced Lighting Inc. (www.aliww.com)

1.4.1 *ANGULO DE INTENSIDAD MEDIA*

Al observar un LED o un conjunto de LEDs encendidos, se puede notar que si la posición del observador es perpendicular a los LEDs, la intensidad de su brillo es mayor que al ubicarse en un cierto ángulo respecto al LED. El ángulo de intensidad media es especificado por el fabricante y se refiere al valor del ángulo en el cual la intensidad de brillo del LED es la mitad de la intensidad medida a 0°. En la figura 1.17 se puede observar los dígitos de un tablero construído con LEDs visto perpendicularmente (fig.1.17 a) y desde un ángulo respecto al tablero (fig.1.17 b).



a) b)

Figura 1.17. Determinación del ángulo de intensidad media, a) Vista perpendicular, b) Vista desde un ángulo respecto al tablero

1.4.2 *MÁXIMO ÁNGULO DE VISTA*

Se refiere al mayor ángulo en el cual se puede distinguir si el LED está encendido o no. Este valor varía de acuerdo a las condiciones de luz del ambiente, por lo cual no se utiliza mucho para evaluar la calidad de un LED y se utiliza más el ángulo de intensidad media.

1.4.3 *CIRCUITOS PARA MANEJAR LEDs*

Para encender un LED se necesita hacer circular una corriente por el mismo, para lo cual se utilizan circuitos para manejar LEDs denominados

controladores o *drivers*. La figura 1.18 muestra 3 circuitos básicos para el manejo de LEDs. La intensidad del brillo de los LEDs depende de la corriente If que pasa por los mismos y depende de la resistencia R. El voltaje de polarización directa del LED depende del valor de If pero se aproxima a un valor constante una vez dimensionada la resistencia R.

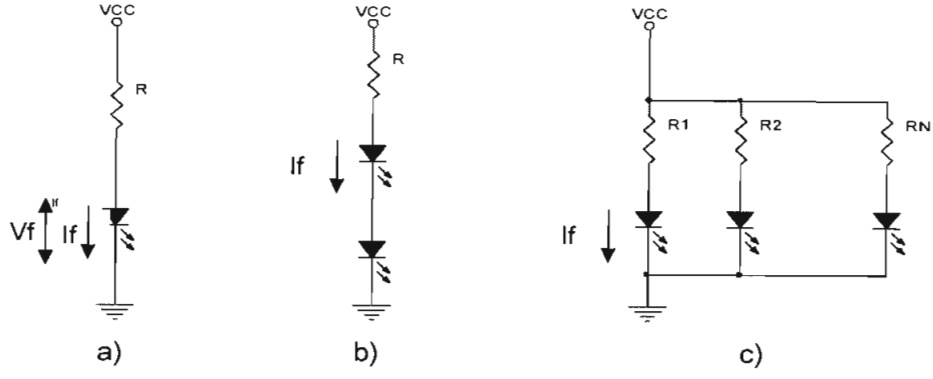


Figura 1.18. Circuitos para manejar LEDs, a) manejo de un solo LED, b) LEDs en serie, c) LEDs en paralelo.

Para estabilizar la intensidad del brillo del LED, se puede mantener la corriente If en un valor estable utilizando un circuito de corriente constante como el que se muestra en la figura 1.19

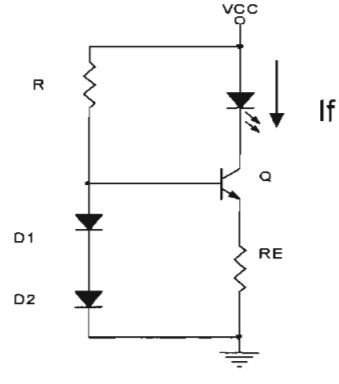


Figura 1.19. Circuito de corriente constante

Con el avance de la tecnología, se ha llegado a construir circuitos integrados capaces de manejar varios LEDs. La figura 1.20 muestra un ejemplo

de este tipo de controladores, compuesto de un registro de desplazamiento de 32 a 35 bits, un *latch* interno, y un circuito de corriente constante para manejar los LEDs cuya corriente de salida es controlada por una única resistencia (R).

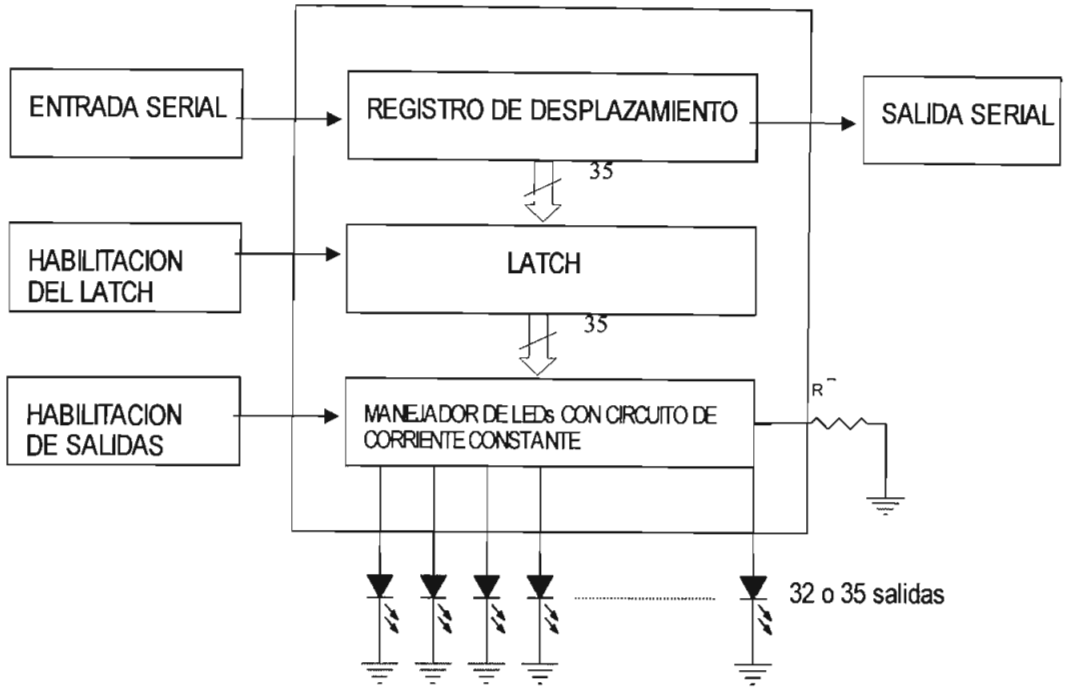


Figura 1.20. Manejador de LEDs con circuito de corriente constante

1.5 MATRIZ DE LEDs

Se refiere al número de puntos o pixels de largo por el número de puntos o pixels de ancho de un *display*. Si estos puntos están conformados por LED's, entonces podemos hablar de una matriz de LED's. En la figura 1.21 se muestra una matriz de LEDs de 8x8.

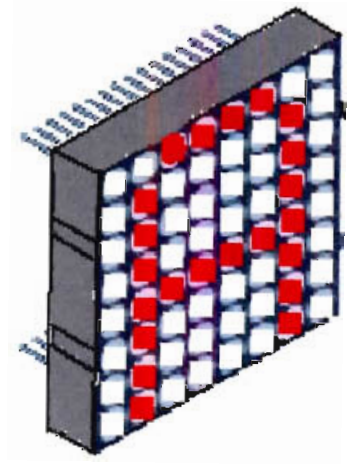


Figura 1.21. Matriz de LEDs 8x8

Los tableros electrónicos se construyen formando matrices y cada *píxel* que compone la matriz generalmente se direcciona mediante el dato de fila y columna en la que se encuentra ubicado.

1.5.1 *TIPOS DE CONSTRUCCIÓN DE MATRICES DE LED'S*

Se refiere a la manera de formar los pixels de una matriz de LED's. Un *píxel* puede estar formado por uno o varios LEDs dispuestos en forma cuadrada, circular u ovalada y puede tener uno o varios colores, los cuales se forman a partir de la combinación de las tonalidades de rojo, verde y azul que son los colores básicos.

Existen dos tipos de construcción de matrices de LED's:

- Construcción Discreta: El término discreto se refiere al diseño tradicional de pixels con LEDS. En construcción discreta, varios LED´s son montados en un circuito impreso para producir muchos pixels en una misma placa de circuito impreso.

- Construcción en *Cluster*: El *cluster* es un diseño mejorado de *pixels* con LEDS. En este método se encapsulan varios LED´s en un tubo y el conjunto forma un solo *píxel*. En la figura 1.22 a) se muestra un *cluster*

de LEDs de un solo color dispuestos en forma circular, En la figura 1.22 b) se muestra un *cluster* cuadrado de dos colores. Para incrementar el brillo de un pixel y la distancia máxima a la cual puede distinguirse se aumenta el número de LEDs por *cluster.*

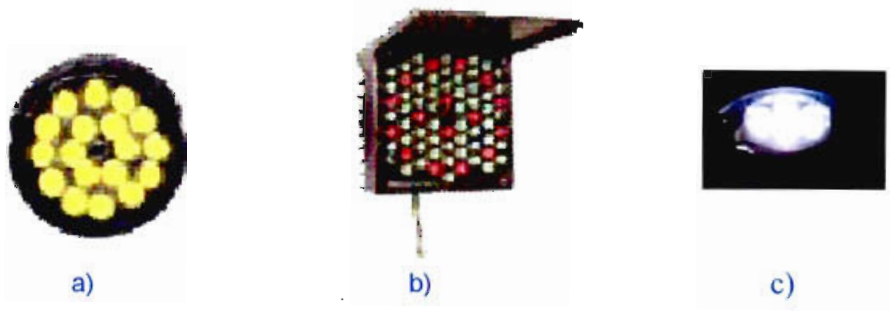


Figura 1.22. Clusters de LEDs, a) redondo, b) cuadrado, c) ovalado

1.5.2 *METODOS PARA MANEJAR DATOS EN UNA MATRIZ DE LEDs*

Para pintar datos en una matriz de LEDs, existen diferentes métodos, entre los cuales se pueden destacar:

a) Manejo Directo Simple: Denominado también Controlador Estático, consiste en que exista un circuito controlador (*driver*) para encender o apagar un LED individualmente. Con este método, el controlador necesita actualizar los datos únicamente cuando éstos cambian pero se requiere de más controladores y componentes y, por tanto, más espacio en la tarjeta lo cual resulta una desventaja cuando el número de LEDs que conforma la matriz es alto. En la figura 1.23 se muestra un esquema de este tipo de manejo de LEDs en una matriz de 5x7.

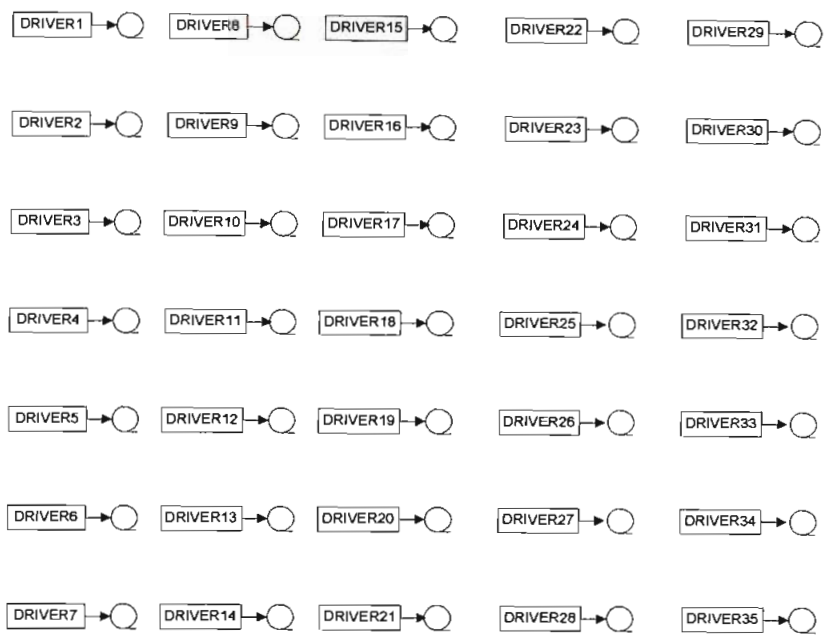


Figura 1.23. Manejo directo simple de una matriz de 5x7

b) Refresco por multiplexación en el tiempo: Llamado también Controlador Dinámico, consiste en que cada fila (o columna) de una matriz de LEDs sea habilitada secuencialmente para actualizar sus datos y realizar este proceso en forma continua, aún cuando los datos no cambien. En la figura 1.24 se muestra un esquema del método de multiplexación en el tiempo para una matriz de 5x7. Al utilizar este método se reduce considerablemente el número de controladores pero se necesita memoria para almacenar los datos que se muestran en la matriz de LEDs.

La tasa de refresco se define como la frecuencia con la cual se habilita a cada fila (o columna). Las formas de los caracteres podrán observarse debido al efecto de la persistencia del ojo humano, esto consiste en que las imágenes luminosas formadas en la retina no desaparecen instantáneamente, razón por la cual el ojo humano no es capaz de distinguir si una luz esta constantemente cambiando de estado de encendido a apagado y viceversa si esto ocurre a una frecuencia mayor a 50 Hz[1]. Típicamente, los fabricantes recomiendan frecuencias

[1]Fuente: National Semiconductor, Application Note 371, Autor: David Stewart

de 100 a 200 Hz. Con una frecuencia de 100 Hz, se tendría un tiempo de refresco de 10 ms para cada fila (o columna), por ejemplo, para una multiplexación de 16 filas se necesitaría un tiempo de 10ms/16 = 625 µs para habilitar y actualizar el dato de cada fila (o columna).

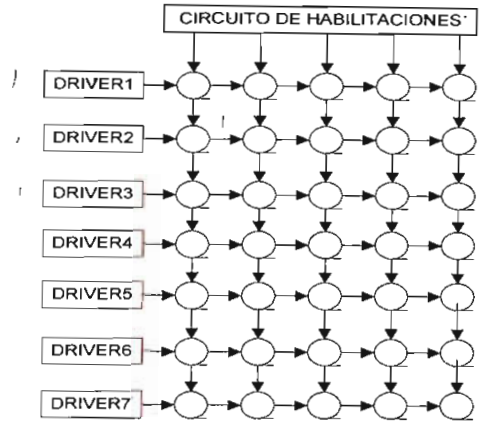


Figura 1.24. Refresco por multiplexación en el tiempo

Cuando se maneja una matriz de LEDs mediante multiplexación en el tiempo, cada LED es habilitado durante un cierto tiempo dentro del periodo de actualización de la matriz lo cual se denomina ciclo de trabajo (*duty cycle*). Esta relación se denota con la letra "d" y se calcula de la siguiente manera:

$$d = \frac{t_e}{T} \qquad \text{(Ec. 1.6)}$$

siendo:

T = Tiempo para actualizar toda la matriz

te = Tiempo en que una fila o columna se habilita para actualizar sus datos durante el periodo T .

El hecho de habilitar a los LEDs durante una fracción del tiempo que se utiliza para actualizar los datos de una matriz hace que la corriente efectiva que pasa por los LEDs disminuya y con esta, su brillo.

El valor efectivo de una señal periódica se calcula de la siguiente manera:

$$I_{efectiva} = \sqrt{\frac{1}{T}\int_0^T (I(t))^2\, dt} \qquad \text{(Ec. 1.7)}$$

Por lo que se puede demostrar que el valor efectivo de la señal de corriente que circula por un LED si se utiliza multiplexación es:

$$I_{efectiva} = \sqrt{d}\,.I_{máxima} \qquad \textit{(Ec. 1.8)}$$

En donde $I_{máxima}$ es el valor pico de corriente aplicado al LED.

Esto reduce el brillo de las figuras que se muestran dificultando el poder distinguir los mensajes. Para solucionar este inconveniente, además de aumentar al máximo posible el ciclo de trabajo y aplicar a los LEDs picos de corriente más altos, se utilizan los LEDs brillantes, súper brillantes y ultra brillantes, cuya característica es proporcionar más mili candelas de intensidad luminosa al pasar la misma cantidad de corriente. En la Tabla 1.2 se muestran los rangos de intensidad luminosa que abarcan los LEDs normales, brillantes, superbrillantes y ultra brillantes.

LEDs	Intensidad Luminosa (mcd)*
Normales (Estándar)	3-100
Brillantes	100-1000
Súper Brillantes	1000-4000
Ultra Brillantes	4000-15000

* Datos medidos cuando circula por el LED una corriente de 20 mA

Tabla 1.2. Rangos de Intensidad Luminosa de los LEDs normales, brillantes, Súper brillantes y Ultra brillantes[1]

[1] Estos valores varían de acuerdo al fabricante

En el presente proyecto de titulación, se pretende crear una matriz de LEDs a partir del movimiento de una única fila de LEDs giratoria, en la cual se pintarán los datos mediante refresco por multiplexación reemplazando el circuito de habilitaciones por la determinación de la posición de la fila de LEDs a medida que ésta se gira.

1.6 SISTEMAS MICROPROCESADOS

Al diseñar sistemas digitales existen distintos grados de complejidad, los cuales permiten clasificarlos de acuerdo a la escala de integración que poseen los elementos utilizados en la solución de un problema en particular.

De acuerdo a esta clasificación se tiene:

- Circuitos de baja escala de integración (SSI), los cuales utilizan las compuertas básicas (AND, OR, NOT, NAND, NOR, Flip Flops) y las soluciones se plantean aplicando el Álgebra de Boole, Mapas de Karnaugh, Tablas de estados, etc.

- Circuitos de Mediana escala de Integración (MSI), mediante los cuales la solución se plantea por bloques o etapas. Utiliza Decoders, Multiplexers, contadores, registros, etc.

- Circuitos de alta escala de integración (LSI), los cuales reemplazan los circuitos lógicos por información almacenada en dispositivos de memoria (RAM, ROM), en donde la información se maneja utilizando circuitos digitales externos.

- Circuitos de muy alta escala de integración (VLSI), mediante los cuales se plantea la solución mediante el desarrollo de un algoritmo o un conjunto de instrucciones que son ejecutadas por un microprocesador o microcontrolador.

Un microprocesador es la construcción en un solo circuito integrado de una unidad central de proceso (CPU), la cual se compone de una unidad aritmética y lógica (ALU) encargada de realizar operaciones aritméticas y lógicas, y una unidad de control la cual se encarga de interpretar las instrucciones del programa que se ejecuta. Un microcontrolador es una CPU, una unidad de memoria donde se almacenan las instrucciones del programa, datos y resultados que se procesan junto con un sistema de entrada y salida de datos en un solo circuito integrado.

Al utilizar circuitos integrados VLSI se puede reducir en gran parte el costo y el tiempo utilizado en el desarrollo de la solución. Además aumenta la confiabilidad del diseño y la posibilidad de realizar de una manera más sencilla cambios o incremento de funciones del sistema diseñado.

Como desventaja se puede mencionar la necesidad de tener herramientas computacionales y hardware que permita almacenar en un dispositivo de memoria las instrucciones del programa desarrollado.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado. La razón de esta tendencia es que los microcontroladores de 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más costosos. Los microcontroladores de 32 bits son utilizados para el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

Entre las características más importantes de los microcontroladores de 8 bits, se tienen:

- Capacidad de procesamiento de palabras de 8 bits.
- Circuito de reloj incorporado.

- Múltiples puertos de entrada/salida programables.
- Bajo consumo de energía en las versiones CHMOS.
- Alta inmunidad al ruido eléctrico.
- Set de instrucciones optimizadas para la adquisición y tratamiento de datos, tablas, multiplicación, división, etc.
- Instrucciones lógicas y de bifurcación orientadas al proceso de señales bit a bit. (procesador booleano)
- Memoria de programa y de datos (ROM y RAM).
- Protección de la memoria de programa (encriptación).
- Perro guardián (*watchdog*), que vigila el funcionamiento óptimo de la CPU.
- Posibilidad de comunicación Full Duplex con otros sistemas.
- Conversores A/D integrados en el propio chip.
- Salida de Modulación de Ancho de Pulso (PWM) para la conversión D/A.
- 2 o 3 temporizadores de 16 bits.
- 5 o 6 interrupciones programables con niveles de prioridad.

En la actualidad los microcontroladores y microprocesadores son muy utilizados para automatizar procesos o mediciones en la industria, equipos electrónicos, artículos utilizados en el hogar, dispositivos para entretenimiento, etc.

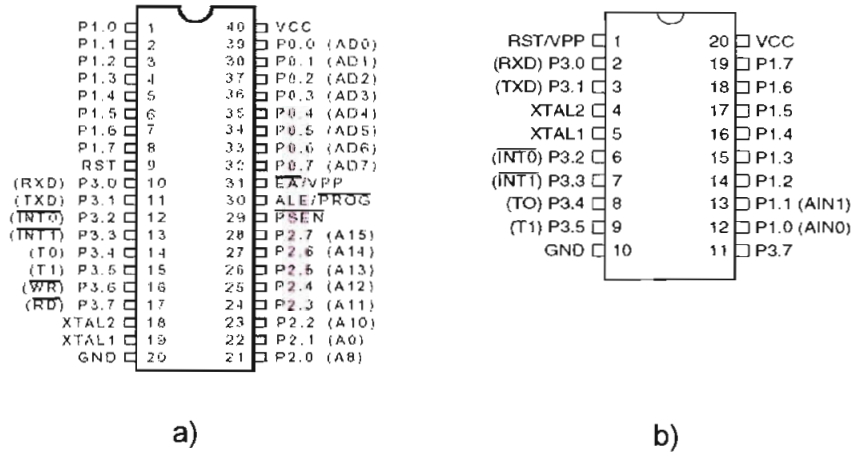
1.6.1 *MICROCONTROLADORES DE ATMEL*

ATMEL es un fabricante de circuitos integrados que ha desarrollado una familia de microcontroladores, entre los cuales se destacan en el presente proyecto de titulación, los microcontroladores AT89C51 y AT89C2051, cuya arquitectura básica presenta las siguientes características:

- Fabricados con tecnología CMOS
- Son totalmente compatibles con los microcontroladores de la familia MCS-51 de INTEL

- Poseen memoria flash reprogramable hasta 1000 veces como memoria de programa con una capacidad de 4Kbytes para el microcontrolador AT89C51 y 2KBytes para el AT89C2051.
- Manejan el mismo set de instrucciones que los microcontroladores de la familia MCS-51
- Permiten frecuencias de cristal de 0 a 24 MHz
- Tres niveles de seguridad de memoria de programa para el AT89C51 y dos niveles de seguridad de memoria de programa para el AT89C2051.
- Memoria RAM interna de 128x8 bits.
- Dos temporizadores de 16 bits
- Manejan 6 tipos de interrupciones.
- Puerto serial programable.
- Comparador analógico incorporado (solamente para el microcontrolador AT89C2051).
- 32 líneas de entrada/salida para el AT89C51 y 15 líneas entrada/salida para el AT89C2051.
- Pueden operar en modo de bajo consumo de potencia.
- Empaquetamiento de 40 pines para el AT89C51 y de 20 pines para el AT89C2051

En la figura 1.25 se muestran estos microcontroladores. La figura 1.25 a) muestra el microcontrolador AT89C51, se puede observar su empaquetamiento de 40 pines y sus 32 líneas de entrada/salida distribuidas en 4 puertos. En la figura 1.25 b) se muestra el microcontrolador AT89C2051 con su empaquetamiento de 20 pines y sus 15 líneas de entrada y salida distribuidas en 2 puertos.



a) b)

Figura 1.25. a) Microcontrolador AT89C51, b) Microcontrolador AT89C2051.

Estos microcontroladores ofrecen grandes ventajas para el desarrollo de proyectos por su alta confiabilidad, facilidad de programación y bajo costo, razón por la cual serán utilizados en el desarrollo del presente proyecto de titulación

CAPITULO 2

DISEÑO DEL GENERADOR DE CARACTERES

En el presente capítulo se describen detalladamente las consideraciones de hardware tomadas en cuenta en el diseño de cada una de las partes que conforman el generador de caracteres.

2.1. DESCRIPCIÓN GENERAL DEL SISTEMA.

El generador de caracteres a diseñarse deberá permitir al usuario mostrar un conjunto de caracteres ASCII, utilizando una fila de LEDs giratoria.

Este generador deberá tener las características de brillo y contraste que permitan leer claramente los mensajes que se muestren en un ambiente de interiores.

El sistema a diseñarse estará compuesto de un computador PC en el cual se instalará un software que funcione bajo Windows con el que se brinde una interfaz amigable entre el equipo y el usuario para que este pueda editar y transmitir los mensajes que van a mostrarse en el generador de caracteres. El computador se comunicará serialmente con un sistema microprocesado encargado de pintar los caracteres en la fila de LEDs mientras va girando. En la figura 2.1. se pueden distinguir cada uno de estos componentes.

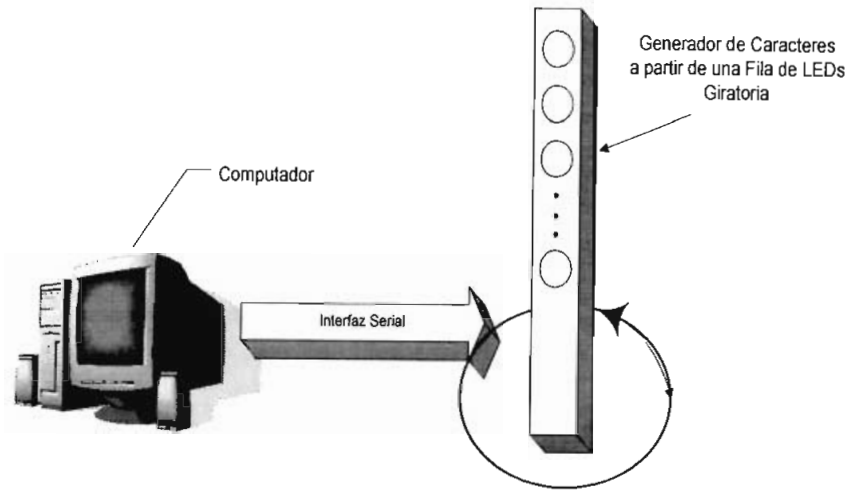


Figura 2.1. Gráfico en bloques del sistema a diseñarse

A continuación se describirá el diseño de cada uno de los componentes del sistema:

2.2. GENERADOR DE CARACTERES.

Es la parte encargada de pintar o dibujar los caracteres en la fila de LEDs giratoria.

2.2.1. PRINCIPIO DE FUNCIONAMIENTO.

En el presente trabajo, se busca realizar una multiplexación de los datos en el tiempo haciendo girar una única fila de LEDs de acuerdo a la siguiente hipótesis: al girar, la fila de LEDs ocupa posiciones diferentes en tiempos diferentes. Si se envían pulsos de corriente a los LEDs durante pequeños intervalos de tiempo en ciertas posiciones, se tendrá como resultado el poder observar figuras, en particular, si nos referimos a una tabla de datos predeterminada, podremos observar las formas de los caracteres. Por ejemplo, para formar los caracteres que se muestran en la figura 2.2, los pulsos de corriente que deben aplicarse en el tiempo a cada uno de los LEDs se indican en la figura 2.3.

Figura 2.2. *Grafico de la información mostrada en el generador del caracteres*

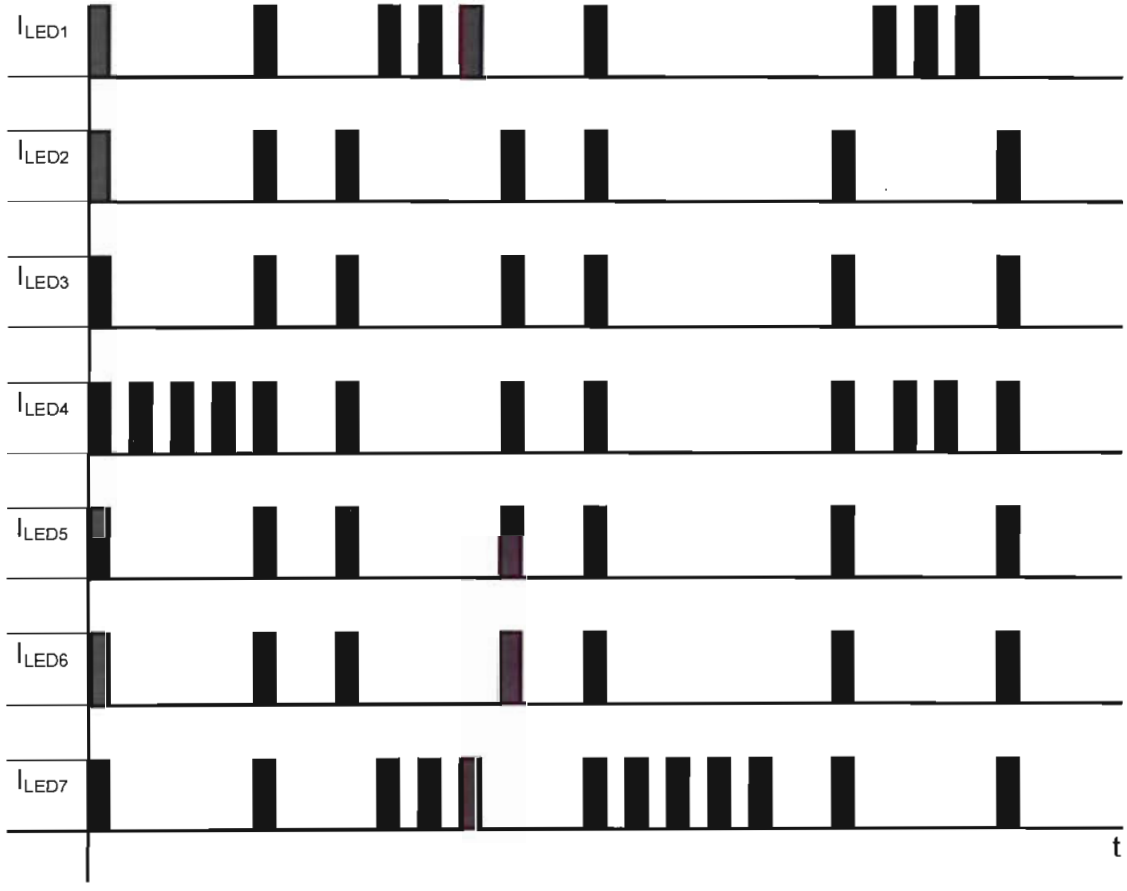


Figura 2.3. *Grafico de la corriente en cada uno de los LEDs vs el tiempo para formar el mensaje de la figura 2.2*

Para lograr esto, se necesitan los siguientes elementos:

- Un dispositivo que permita girar a la fila de LEDs.
- Una fuente de voltaje que permita alimentar a los circuitos que estarán en permanente movimiento.
- Un circuito que permita determinar en qué posición se encuentran los LEDs en un determinado instante de tiempo.
- Un interfaz de comunicaciones entre el computador y el generador de caracteres, el cual es un dispositivo en constante movimiento.
- Un circuito que permita manejar los LEDs de tal forma que se pueda controlar la corriente (y por lo tanto el brillo) que circule por cada uno de ellos en un instante determinado.
- Un sistema microprocesado que controle cada uno de los elementos antes mencionados.

En la figura 2.4 se muestra un diagrama de bloques de los elementos que componen el generador de caracteres.

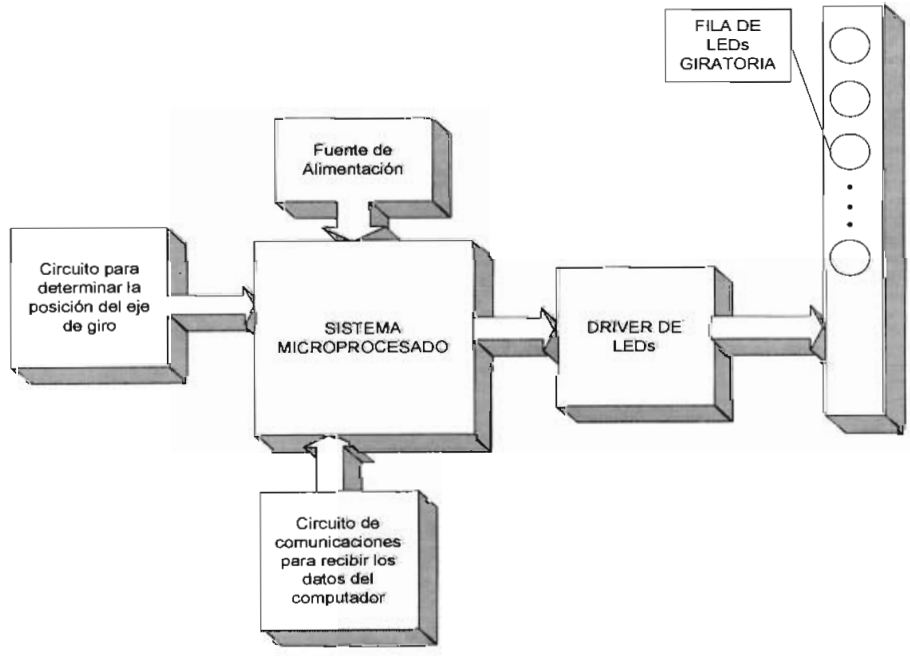


Figura 2.4. Diagrama de Bloques del Generador de Caracteres

A continuación se describe el diseño de cada una de las partes del generador de caracteres:

2.2.2. DISPOSITIVO DE GIRO.

Para hacer girar la fila de LEDs se necesita un motor. Los LEDs y el circuito de control del tablero deberán colocarse de manera que no alteren significativamente el equilibrio del peso para el motor y por lo tanto su funcionamiento.

Además deberá comprobarse si la velocidad de giro del motor es la adecuada para poder observar caracteres en este dispositivo sin ningún efecto molesto para el ojo humano tal como el que las letras que se muestren titilen. Si fuese este el caso, será necesario que el motor gire a mayor velocidad. En teoría, el ojo humano no distingue si una luz esta constantemente cambiando de estado de encendido a apagado y viceversa si esto ocurre a una frecuencia mayor a 50 Hz. Debido a esta razón, para no tener este efecto de titilado, la fila de LEDs debería girar a una velocidad de 3000 rpm y de esta manera pintar el dato correspondiente a una posición determinada 50 veces por segundo. Si se elige un motor que posea un control de velocidad se podrá encontrar una velocidad adecuada para que los caracteres puedan ser vistos sin ningún inconveniente y se podrá comprobar en la práctica si la hipótesis planteada es o no correcta.

2.2.3. FUENTE DE VOLTAJE PARA UN DISPOSITIVO EN MOVIMIENTO.

Debido a que los LEDs, el microcontrolador y el *driver* están en movimiento, se necesita proporcionar una fuente de voltaje para que puedan encenderse. Para resolver este problema existen las siguientes posibilidades:

- Utilizar una batería ubicada en el eje rotatorio, así se tendrá una fuente de voltaje giratoria. Tal como se esquematiza en la figura 2.5

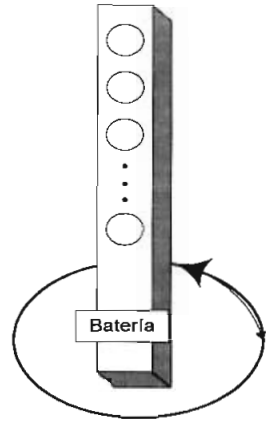


Figura 2.5. Fuente de voltaje con batería

- Construir un sistema parecido al de las escobillas de un motor, es decir, que consista en un par de conductores girando de tal manera que siempre estén en contacto con una fuente de voltaje fija, y así, proporcionar una fuente al dispositivo. El esquema de esta opción se muestra en la figura 2.6

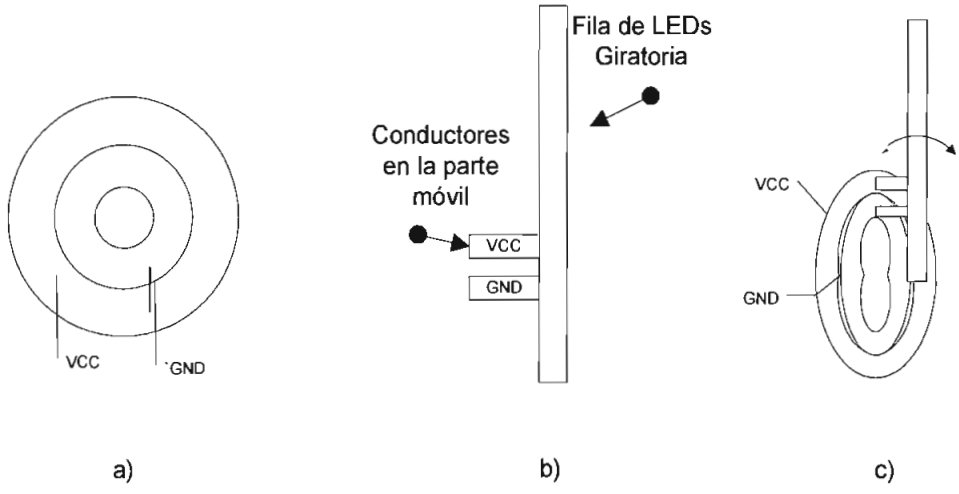


Figura 2.6. a) Discos conductores concéntricos (Parte Fija), b) Conductores acoplados a la fila de LEDs Giratoria (Parte Móvil), c) Parte Fija y Móvil Acopladas

- Diseñar algo similar a un generador de voltaje DC, es decir, aprovechar la fuerza mecánica generada por el giro del motor para hacer girar una bobina y crear un campo magnético que permita generar un voltaje DC en la parte móvil del generador de caracteres.

En los dos últimos casos, La fuente proporcionará un voltaje DC no regulado, por lo tanto deberá añadirse un C.I. regulador de voltaje en la parte móvil del sistema, es decir, en el circuito de control del generador de caracteres para eliminar el rizado que pueda producirse en la fuente y asegurar el voltaje necesario para el correcto funcionamiento del sistema.

2.2.4. INTERFAZ PARA DETERMINAR LA POSICIÓN DEL EJE DE GIRO.

Para poder pintar los datos conforme los LEDs van girando se necesita saber en qué posición se encuentra la fila de LEDs. Para ello es necesario determinar la velocidad del motor y con esta calcular el tiempo que se requiere para desplazarse un ángulo determinado. Este tiempo será el periodo en el cual cambiarán los datos pintados en la fila de LEDs.

Para encontrar este ángulo se han graficado los datos a distintos ángulos para poder elegir el más adecuado para representar los caracteres. A continuación se muestran las diferentes pruebas realizadas gráficamente con una fila de 8 LEDs.

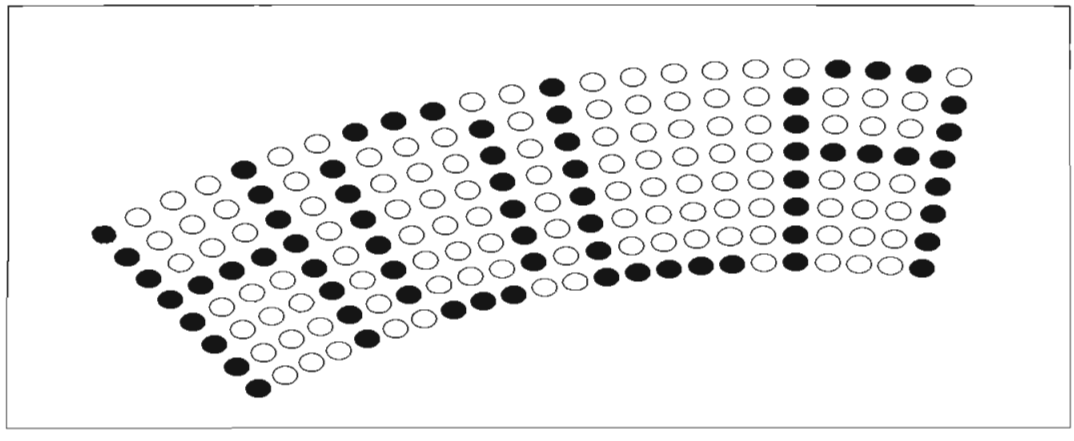


Figura 2.7. Gráfico con 2 grados de separación

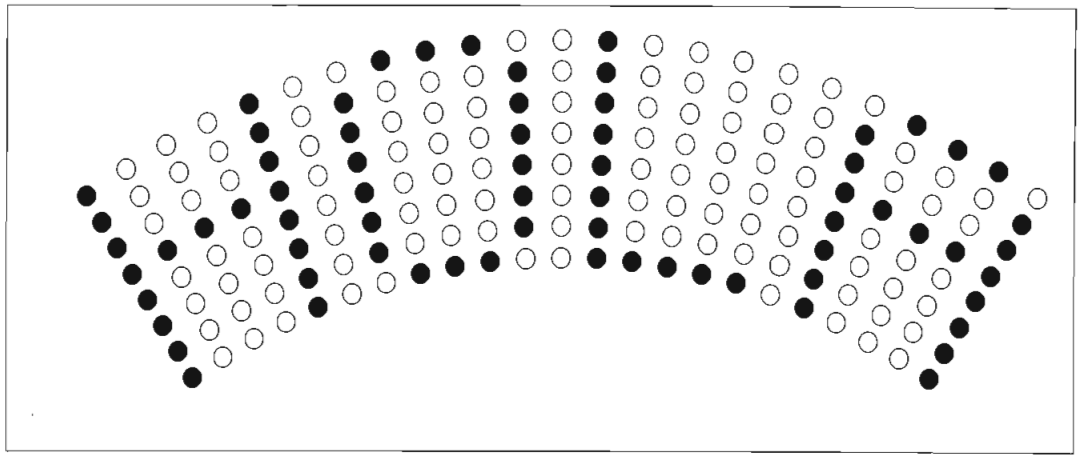


Figura 2.8. Gráfico con 3 grados de separación

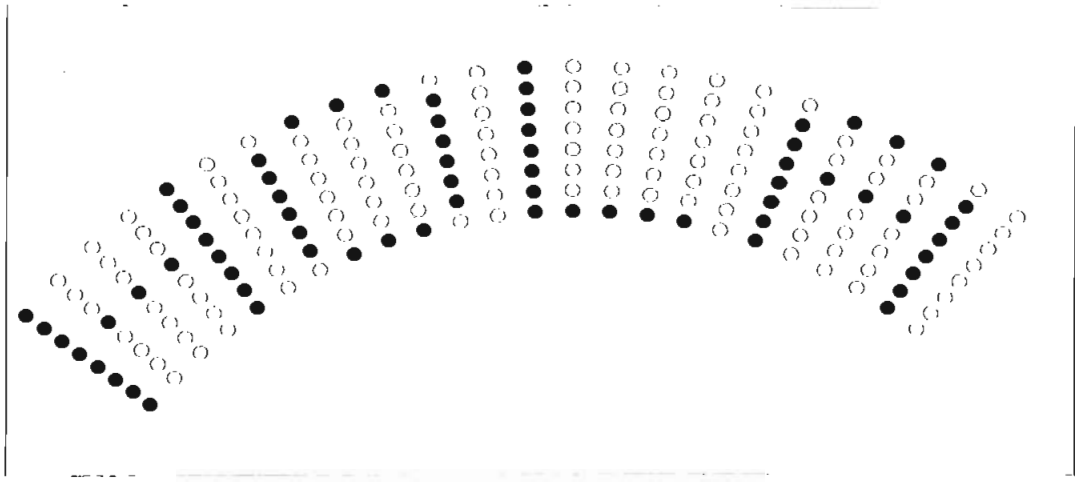


Figura 2.9. Gráfico con 4 grados de separación

Mediante la comparación de estos gráficos se decide pintar los datos en los LEDs cada 3 grados puesto que se considera que con este ángulo se puede obtener un *pitch* adecuado para el generador de caracteres. Con este ángulo de separación y tomando en cuenta que la fila de LEDs recorre un ángulo de 360° por revolución, se puede generar una matriz de 120 columnas de 8 pixels.

Este valor de 3° servirá de referencia para un cálculo preliminar, esto puede cambiar al realizar las pruebas del tablero una vez construido.

Debido a que se tomarán intervalos de tiempo para determinar la posición del eje de giro, se necesita establecer un origen a partir del cual se comience a contar los intervalos de tiempo y obtener las diferentes posiciones en donde se pintarán los datos en los LEDs. Una forma de proporcionar un origen para el generador de caracteres es el circuito de la figura 2.10 a) formado por un fotodiodo y un fototransistor. Un fotodiodo es un LED emisor de infrarrojo (IRED), el cual al estar polarizado directamente emite una señal de luz infrarroja, en cambio, el fototransistor recibe esa señal de luz, y la transforma en corriente eléctrica. El fotodiodo se ubicará en una parte fija del generador de caracteres y el fototransistor se ubicará en la fila de LEDs giratoria, es decir en la parte móvil del generador de caracteres, tal como se indica en la figura 2.10 b) y el circuito funcionará de la siguiente manera: cuando el fototransistor funcionando en corte y saturación pase frente al fotodiodo en polarización directa, recibirá la señal de luz infrarroja que saturará al fototransistor y la señal de salida pasará de alto a bajo, de esta forma se podrá obtener una señal que indique que la fila de LEDs ha pasado por el origen.

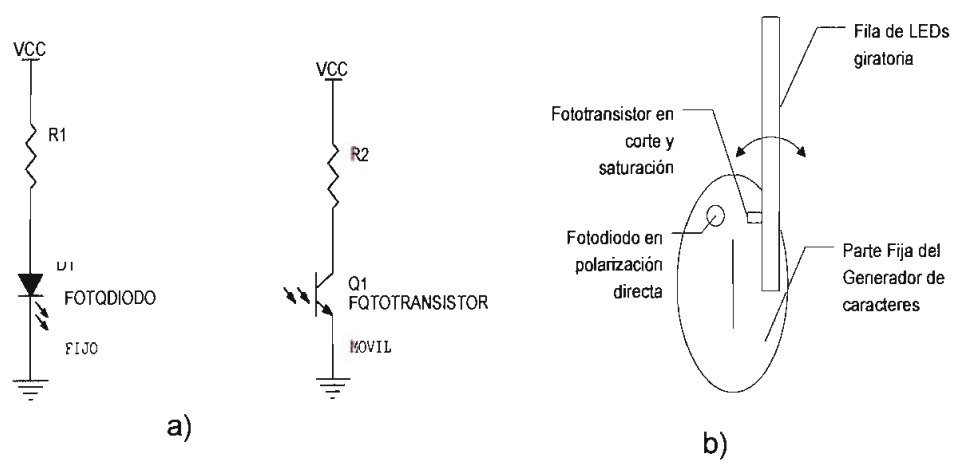


Figura 2.10. a) Circuito de acoplamiento óptico para determinar el origen del sistema, b) Esquema de la ubicación de los elementos del circuito

Debido a que se va a trabajar con señales TTL, la salida deberá tomar valores de 0 o +5V.

Al recibir una señal de luz infrarroja, el fototransistor comienza a conducir de acuerdo al nivel de luz que recibe, por esta razón la señal de salida que se obtenga podrá tener valores dentro de rango de 0 a +5V a medida que el dispositivo vaya acercándose o alejándose del origen. Debido a que se necesita una señal de 2 niveles lógicos se adicionará un circuito doble inversor con disparador de Schmitt. El disparador de Schmitt es un circuito comparador con retroalimentación positiva, lo cual permite acelerar el ciclo de conmutación. Esto aumenta la ganancia y por tanto agudiza la transición entre los dos niveles de salida. La retroalimentación positiva mantiene al comparador en uno de los dos estados de saturación a menos que se aplique una entrada lo suficiente mente grande para sobrepasar la retroalimentación. Esta característica permite minimizar las señales que no corresponden a un estado lógico de 0L o 1L.

Con estas consideraciones, el circuito para determinar el origen en el generador de caracteres se muestra en la figura 2.11:

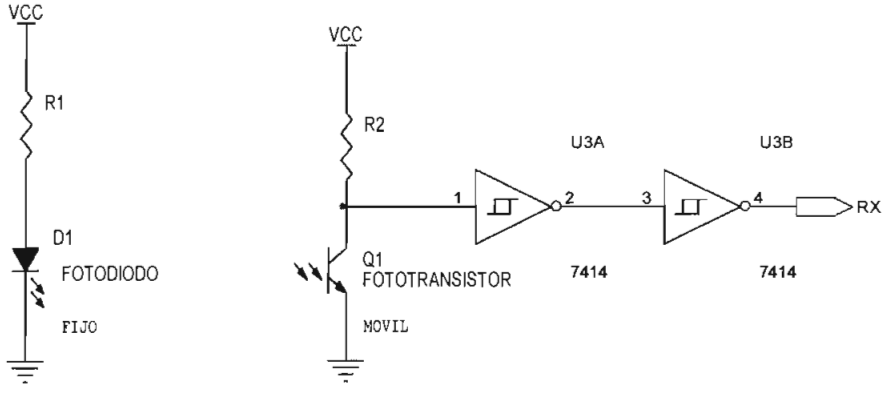


Figura 2.11. Circuito para determinar el origen del sistema

Se utilizará el fotodiodo ECG 3017, el cual soporta corriente máxima de 150 mA en polarización directa. El fototransistor a utilizarse es el ECG3037, cuya corriente de colector máxima es de 50 mA. Estas consideraciones serán tomadas en cuenta para calcular los valores de las resistencias R1 y R2 indicadas en la figura 2.11

2.2.5. COMUNICACIÓN CON EL GENERADOR DE CARACTERES.

La capacidad de transmitir datos al generador de caracteres es una característica muy importante, ya que permite que el usuario pueda modificar los mensajes que se muestren a su conveniencia. Por lo tanto, es necesaria una comunicación de tipo simplex para poder actualizar los datos del tablero.

Entonces es necesario encontrar la manera de comunicarse con el generador de caracteres que es un dispositivo en movimiento, por lo que se plantean dos soluciones:

a) Utilizar un control remoto infrarrojo. Para ello se necesita un *hardware* que permita transformar la señal eléctrica a señal de luz en el espectro infrarrojo, las características de este *hardware* son semejantes al circuito utilizado para determinar el origen del sistema. Existen en el mercado dispositivos mucho más sensibles a la luz infrarroja, los cuales podrían ser utilizados para recibir los datos en el generador de caracteres. En la figura 2.12 se muestra el circuito diseñado para implementar un control remoto.

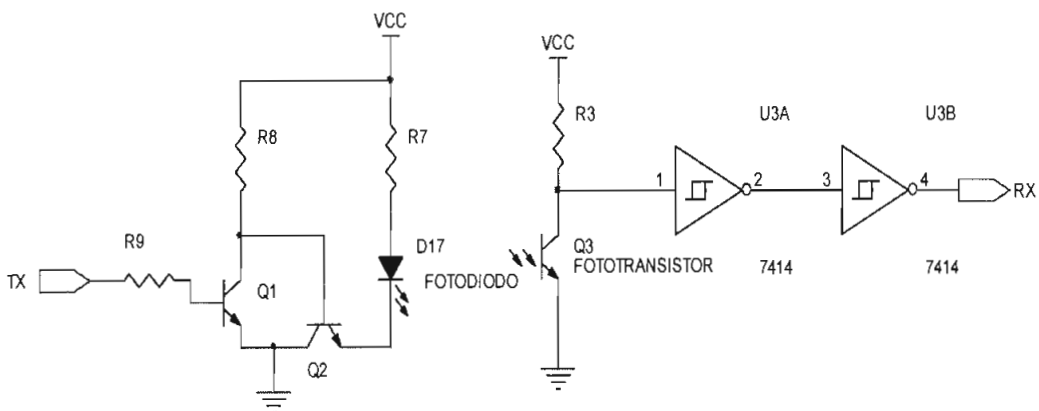


Figura 2.12. Circuito de Control Remoto Infrarrojo

Cuando se transmita un 1L el fotodiodo no conduce y por lo tanto el fototransistor no detecta ninguna señal y se tiene un 1L en la señal de recepción. Al transmitir un 0L, el fotodiodo entra en

conducción y el fototransistor detecta la señal infrarroja saturándose y llevando a la salida un 0L

b) Implementar el segundo mecanismo propuesto para proporcionar voltaje DC al generador de caracteres, es decir, un conductor ubicado de tal manera que al girar el motor, siempre esté en contacto con una línea de datos fija tal como se ilustra en la figura 2.13

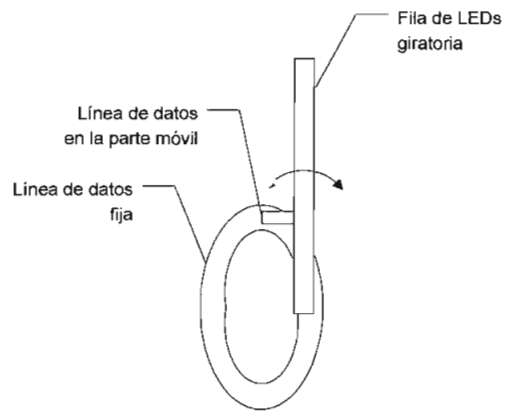


Figura 2.13. Esquema de la segunda opción propuesta para recepción de datos en el generador de caracteres

2.2.6. MANEJO DE LOS LEDS.

Una característica importante en los tableros electrónicos es la intensidad luminosa de los LEDs, la cual se obtiene al pasar corriente a través de ellos. El fabricante proporciona el dato de intensidad luminosa en mili candelas (mCd) y los niveles de corriente recomendables para obtener este nivel de brillo sin reducir la vida útil del LED. En este caso se utilizarán LEDs brillantes, cuya intensidad luminosa es de 600 mCd cuando circula por ellos una corriente de 20 mA, para poder obtener un mayor brillo en los LEDs utilizando la misma cantidad de corriente y, por lo tanto, los caracteres generados podrán verse mucho mejor.

También influye mucho el tener el mismo nivel de brillo en todos los LEDs, con lo cual mejorará la calidad de la imagen que se muestre, por lo tanto

se necesita que circule la misma corriente por cada uno de ellos, y además que esta corriente sea constante, debido a esto, se deberá implementar una fuente de corriente constante para cada LED.

Existen circuitos integrados diseñados (C.I.) exclusivamente para el manejo de LEDs, los cuales en sus salidas presentan circuitos equivalentes a fuentes de corriente constante la cual puede controlarse utilizando una sola resistencia conectada al integrado.

Entre los circuitos integrados diseñados para el manejo de LEDs están el A6275 y el A6276, los cuales permiten manejar 8 y 16 LEDs respectivamente. Poseen un registro de desplazamiento que permite actualizar los datos serialmente a una velocidad superior a 20MHz y *latches* para mantener estos datos en las salidas del integrado. Además, sus salidas son fuentes de corriente constante cuya corriente es determinada por el usuario a través de una única resistencia conectada al integrado, y con una corriente máxima de 90mA en cada una de ellas.

Entre las señales de control de estos C.I. manejadores de LEDs se pueden distinguir:

- *Serial Data In:* Utilizado para ingresar los datos a pintarse en los LEDs al registro de desplazamiento.
- *Serial Data Out:* Permite obtener los datos seriales que han sobrepasado el número de bits del registro de desplazamiento, se utiliza para conectar varios *drivers* en cascada.
- *Clock (CLK):* Es la entrada de la señal de reloj que sincroniza los datos ingresados al registro de desplazamiento.
- *Latch Enable*: Habilitación del *latch* interno del integrado que almacena el dato ingresado serialmente.
- *Output Enable:* Sirve para habilitar las salidas o apagarlas todas con tan solo poner esta entrada en alto. Esto es muy útil para eliminar el denominado "*efecto fantasma*", el cual se produce al multiplexar los

datos en varias filas de LEDs con un solo *driver*. Al cambiar la fila habilitada y pintar los datos correspondientes a esa fila, puede suceder que los datos anteriores permanezcan aún en las salidas del *driver* y dar la impresión de tener una "sombra" del dato anterior. Este problema se evita apagando todas las salidas antes de cambiar el dato, a esto se le denomina "*blanqueo interdígitos*". En el caso del generador de caracteres, esta habilitación permitirá encender los LEDs solamente en el instante en que pasen por una posición determinada y el resto del tiempo mantenerlos apagados.

En el diagrama de bloques de la figura 2.14 se puede observar cada una de las partes que componen el *driver*. Los datos a pintarse en los LEDs deberán ser enviados serialmente al C.I. manejador de LEDs por la entrada serial (*Serial Data In*) y sincronizados a través de la señal de reloj (*CLK*). Una vez enviado el dato deberá activarse la señal de habilitación del *latch* interno (*Latch Enable*) para guardar el dato y por último, la señal de habilitación de las salidas (*Output Enable*) permitirá mostrar o no el dato en los LEDs.

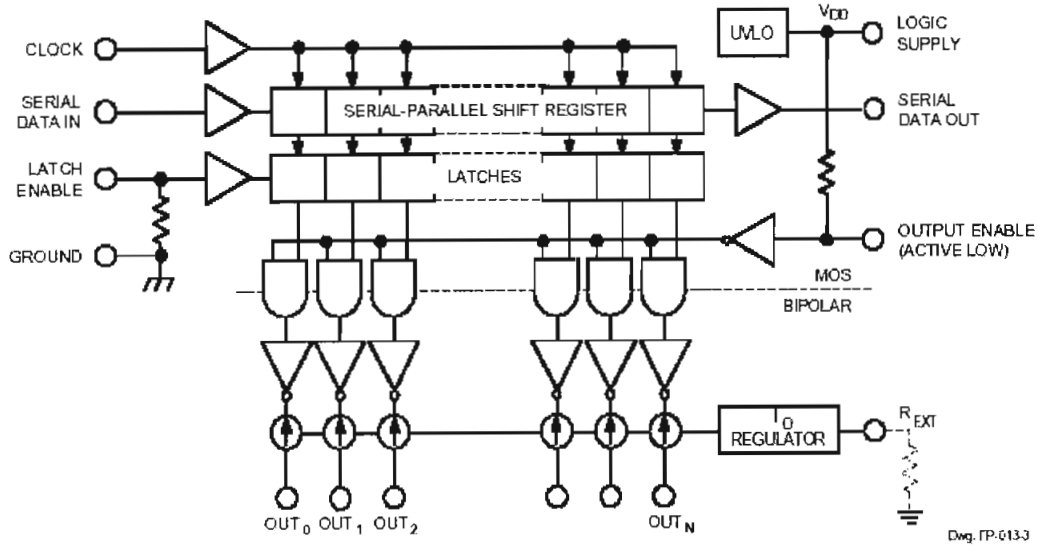


Figura 2.14. Diagrama de bloques del CI A6275[1]

[1] Fuente: Hoja de especificaciones del CI A6275

Estos circuitos integrados están diseñados para operar con voltajes de 0.4V a 0.7V en cada una de sus salidas y con caídas de voltaje de 1.2V a 4V en los LEDs polarizados directamente. Si se tienen mayores niveles de voltaje, la disipación de potencia del integrado crecerá significativamente. Para minimizarla, se recomienda usar una fuente de voltaje del menor valor posible en los LEDs, lo cual se obtiene añadiendo un diodo zener, o varios diodos en serie al circuito, o en su defecto, colocando resistencias limitadoras de corriente en serie con cada uno de los LEDs, tal como se indica en la figura 2.15.

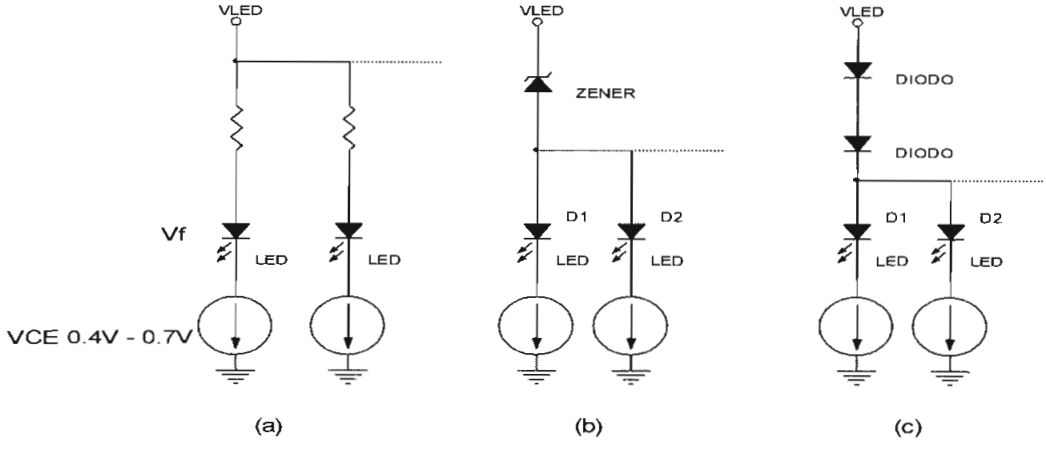


Figura 2.15. Circuitos para minimizar la disipación de potencia en los C.I. A6275 y A6276, a) Resistencias limitadoras de corriente para cada LED, b) Diodo zener en serie a la fuente de voltaje, c)Varios diodos en serie a la fuente de voltaje

Las Hojas de especificaciones de los CI A6275 y A6276 se muestran en el anexo A del presente proyecto de titulación.

Para el generador de caracteres, se utilizará el C.I A6275 para manejar una fila de 8 LEDs, de esta manera, al hacer girar los LEDs, se podrán pintar datos como si se tratara de matrices de 5x7. Para minimizar la disipación de potencia del integrado se colocarán diodos en serie a la fuente de voltaje.

2.2.7. DISTRIBUCIÓN DE LOS RECURSOS DEL MICROCONTROLADOR

Para implementar el generador de caracteres se utilizará el microcontrolador AT89C2051 de ATMEL. Se ha elegido este microcontrolador debido a su empaquetado de 20 pines que ayuda a reducir el espacio que ocupa la tarjeta de control del generador de caracteres puesto que el tamaño y peso de la tarjeta influyen mucho para minimizar el desequilibrio producido al acoplar la tarjeta al motor.

Las Hojas de especificaciones del microcontrolador AT89C2051 se muestran en el anexo A.

Para el control del manejador de LEDs se utilizarán cuatro líneas del microcontrolador las cuales tendrán las siguientes funciones:

- Dos líneas se utilizarán para implementar por software un registro de desplazamiento de 8 bits con una señal de reloj, lo cual nos servirá para transmitir los datos al *driver* A6275.
- Una línea para habilitar la retención (*latch*) del nuevo dato a pintarse.
- Una línea para habilitar todas las salidas del *driver* y mostrar en los LEDs el dato retenido o apagar todos los LEDs para tener blanqueo interdígitos.

La señal de salida del circuito que indica que el eje de giro ha pasado por el origen se conectará a uno de los pines de interrupción externa del microcontrolador. Por tanto, cada vez que se produzca esta interrupción, se comenzará a pintar los datos correspondientes al primer caracter.

Para recibir los datos a pintarse en el generador de caracteres, como se mencionó anteriormente, se utilizará una comunicación de tipo serial RS232. El microcontrolador posee un puerto de comunicaciones de tipo serial con cuatro modos de funcionamiento, los cuales se indican en la tabla 2.1.

MODO	DESCRIPCIÓN
Modo	Registro de desplazamiento (velocidad = fosc/12)
Modo	UART de 8 bits (velocidad variable)
Modo	UART de 9 bits (velocidad = fosc/32 o fosc/64)
Modo	UART de 9 bits (velocidad variable)

Tabla 2.1. Modos de funcionamiento del puerto serial del microcontrolador AT89C2051

Para el presente trabajo se utilizará el puerto serial del microcontrolador en modo 1 para establecer una comunicación serial RS232 de 8 bits con el computador. Se utilizará un cristal de 11.0592 MHz para poder establecer una comunicación serial en cualquiera de las velocidades estándar de RS232.

Tomando en cuenta las consideraciones descritas anteriormente, en la figura 2.16 se muestra el circuito de control del generador de caracteres:

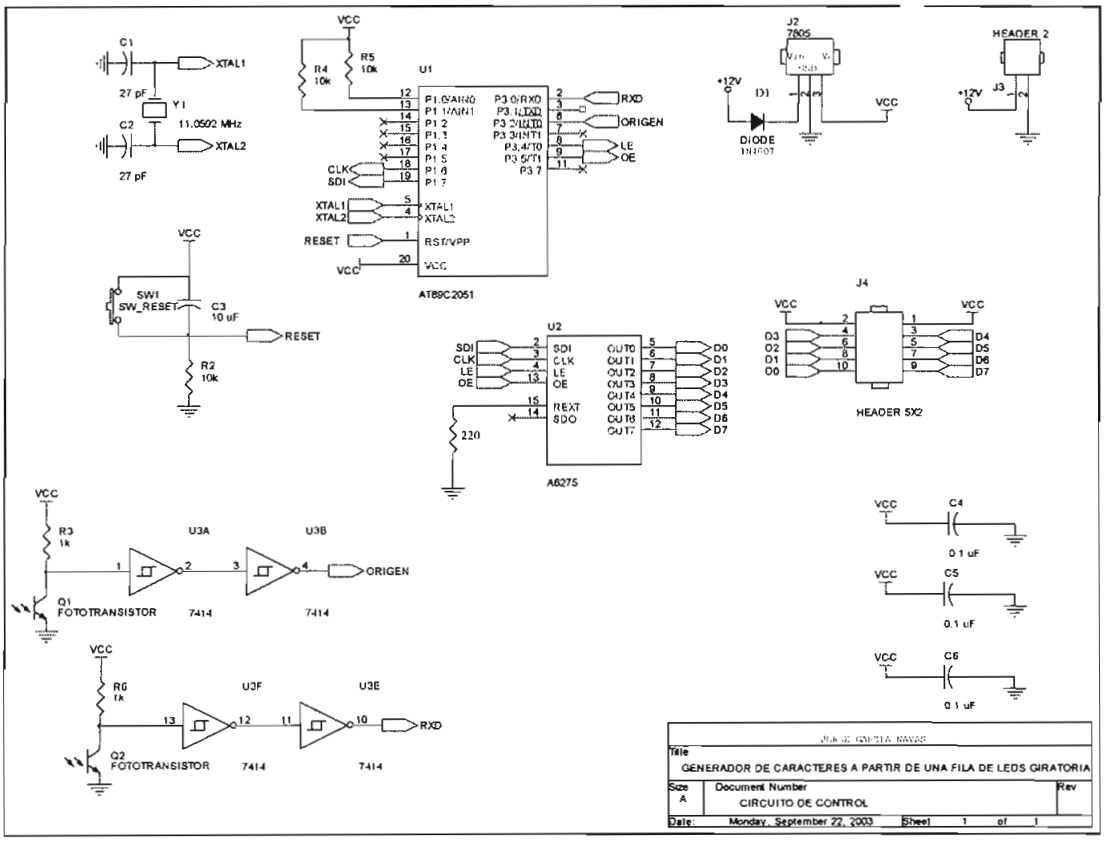


Figura 2.16. Circuito de control del generador de caracteres

corriente que se presenta en las hojas de especificaciones del mismo, pudiendo elegirse un valor de 80 mA. El valor de resistencia para obtener esta corriente se encuentra utilizando la curva de la figura 2.17 que muestra la corriente de salida por bit del CI A6275 vs el valor de la resistencia externa conectada al integrado. Según esta curva, para obtener una corriente de 80 mA en cada una de las salidas, se necesita una resistencia de 220Ω. Estos valores de corriente se reducen cuando se manejan los LEDs mediante refresco por multiplexación en el tiempo de acuerdo al ciclo de trabajo. El valor de R1=220Ω servirá de referencia y podrá reducirse en caso de ser necesario hasta obtener la corriente (y por lo tanto el brillo) necesaria.

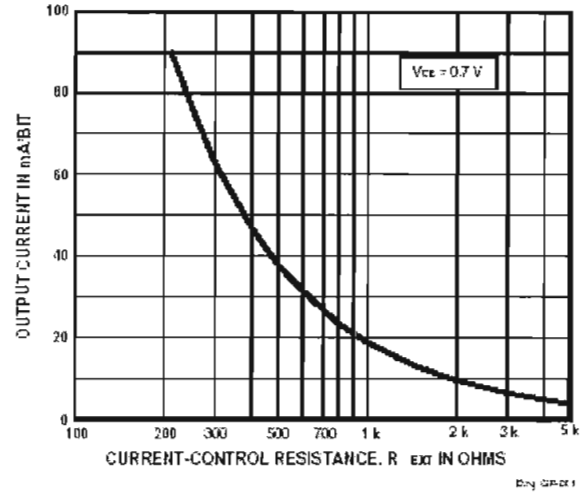


Figura 2.17. Corriente de salida por bit del CI A6275 vs el valor de la resistencia externa conectada al integrado.[1]

2.2.8. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL GENERADOR DE CARACTERES

Cuando la fila de LEDs pase por el origen del sistema, comenzará a medirse el tiempo para que ésta gire un ángulo determinado y cambiar los datos a pintarse en los LEDs cada vez que recorran este ángulo accediendo a localidades de memoria en las que se almacenen una tabla de datos correspondientes a los caracteres ASCII. Cada vez que

[1] Fuente: Hoja de especificaciones del CI A6275

los LEDs pasen por el origen, se comenzara a pintar desde el primer dato.

Los mensajes que se muestren podrán ser actualizados en cualquier momento por el usuario mediante la interrupción que genera la recepción de un dato por el puerto serial del microcontrolador.

2.3. INTERFAZ ENTRE EL COMPUTADOR Y EL GENERADOR DE CARACTERES

Para que el usuario pueda actualizar los datos en el generador de caracteres, se necesita un dispositivo que le permita configurar fácilmente los mensajes que van a mostrarse. Se utilizará un computador o PC debido a que se puede implementar de manera sencilla un software que controle el puerto serial del computador y transmitir de esta manera los datos del usuario al generador de caracteres.

El programa deberá tener las siguientes características:

- Proveer una interfaz amigable y fácil de utilizar entre el usuario y el generador de caracteres.
- Permitir simular o tener una vista previa del mensaje que se va a visualizar en el generador de caracteres antes de enviar estos datos.
- Permitir generar un archivo de texto que contenga un conjunto de mensajes para enviar estos datos al generador de caracteres
- Comunicarse serialmente mediante RS232 con el generador de caracteres para enviar los datos a pintarse.
- Simular, guardar en un archivo y enviar al generador de caracteres secuencias de mensajes que permitan observar animaciones en el generador de caracteres.

CAPITULO 3

CONSTRUCCIÓN DEL GENERADOR DE CARACTERES

En este capitulo se detallan los aspectos concernientes a la implementación del *hardware* y software del generador de caracteres, así como el software para el PC y las consideraciones que se tomaron en cuenta para el acoplamiento de todo el sistema.

3.1. IMPLEMENTACIÓN DEL HARDWARE Y SOFTWARE DEL GENERADOR DE CARACTERES

A continuación se describe detalladamente la construcción del generador de caracteres a partir de las consideraciones de diseño que se mencionan en el capítulo anterior. La construcción se inicia con la elección del dispositivo de giro y el diseño de la tarjeta de circuito impreso del circuito a implementarse, luego se implementa la fuente de alimentación y el interfaz para determinar la posición de la fila de LEDs. Con este *hardware* se podrá implementar el software del microcontrolador para finalmente, construir el interfaz de comunicación serial que permitirá editar los mensajes que se muestren en el generador de caracteres.

3.1.1. DISPOSITIVO DE GIRO Y DISEÑO DE CIRCUITOS IMPRESOS

Como se menciona en el capítulo anterior, se requiere un motor para hacer girar los LEDs. Existen en el mercado diferentes tipos de motores entre los cuales, como primera opción, se escogieron los motores utilizados como

ventiladores en los computadores debido a que son fáciles de conseguir y de bajo costo. En particular, se tiene un ventilador que funciona con 120 V de voltaje AC a 60 Hz, el cual se muestra en la figura 3.1.

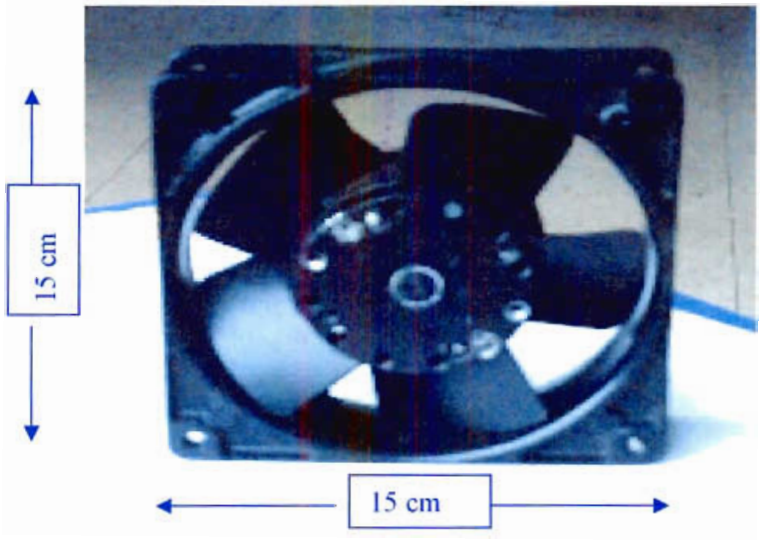


Figura 3.1 *Ventilador escogido como primera opción para el generador de caracteres*

Se implementó la tarjeta de control basada en el circuito de la figura 2.12. Se utilizó el C.I. A6276 para manejar 16 LEDs, con el objeto de estudiar la posibilidad de realizar el generador de caracteres con una fila de 16 LEDs.

Para la fuente de voltaje se implementó la primera opción mencionada en el capítulo anterior, es decir, se colocó una batería de 9 voltios DC en la tarjeta del circuito de control del generador de caracteres.

En vista de que el peso de la tarjeta debe tener la menor influencia posible en el motor, puesto que el peso de la misma provoca una disminución de la velocidad de giro del motor, se colocó la batería, los circuitos integrados y demás elementos lo más cerca posible del centro del motor para reducir el torque que se necesita para hacer girar a la tarjeta.

Con estas consideraciones, el circuito de control implementado para la primera opción del generador de caracteres se muestra en la figura 3.2.

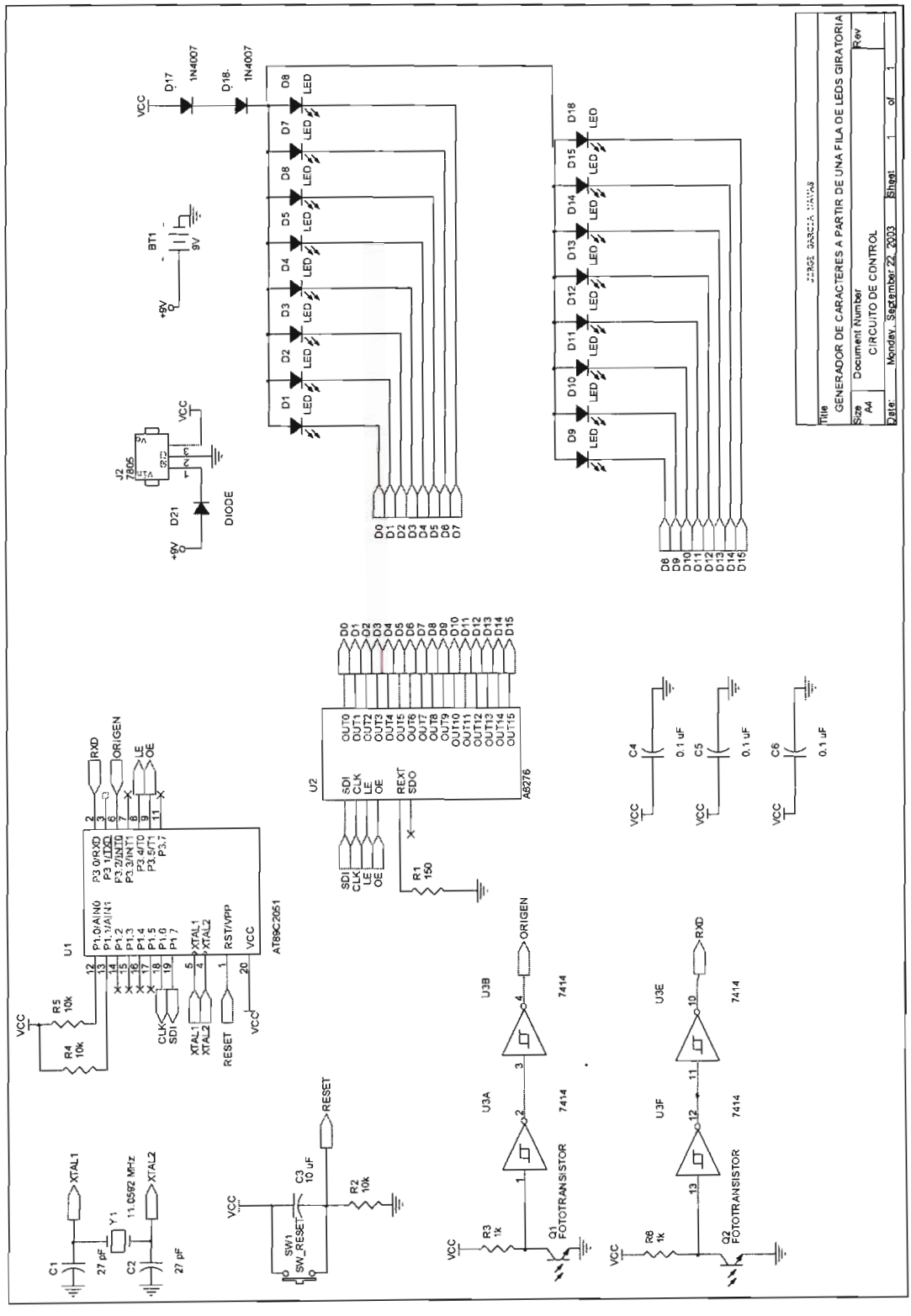


Figura 3.2 *Circuito de la tarjeta de control del generador de caracteres*

A partir de este esquemático, se realizó el diseño del circuito impreso. La figura 3.3 muestra la distribución de los elementos en el circuito impreso diseñado para la tarjeta de control del generador de caracteres.

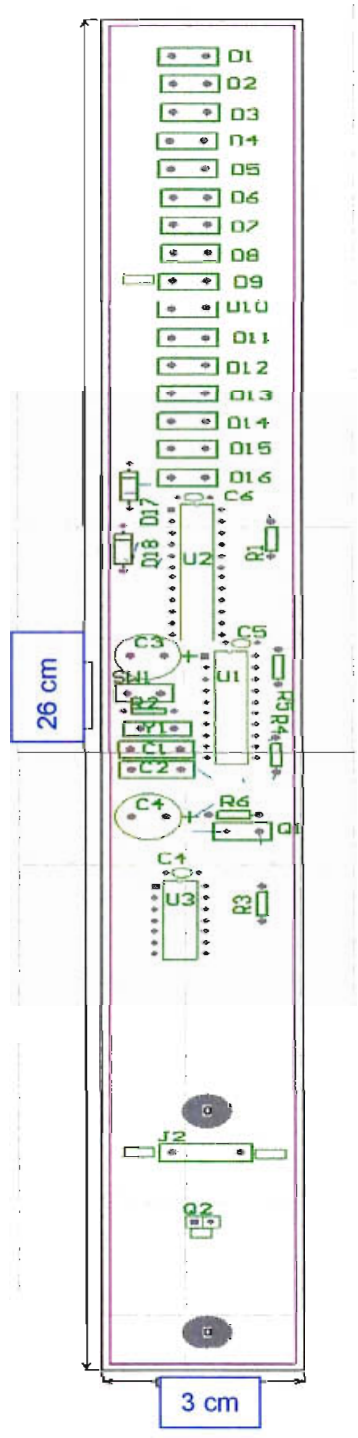


Figura 3.3 *Circuito impreso de la tarjeta de control del generador de caracteres.*

Al acoplar el circuito impreso al motor y hacerlo girar se pudo observar lo siguiente:

- La longitud de la tarjeta y el peso de los componentes es demasiado grande y hace que esta se incline, por lo tanto, se hace necesario distribuir los elementos de forma diferente.

- El desequilibrio que produce el peso de la tarjeta acoplada al motor hace que la velocidad de giro se reduzca en gran proporción, por lo tanto, se necesitará también un motor de mayor torque para hacer girar a la fila de LEDs. Este desequilibrio produce también una vibración muy fuerte en el motor, por lo que se necesita una estructura mucho más robusta y también hallar la forma de equilibrar el peso en el motor.

De acuerdo a estas consideraciones, se hizo necesario buscar otro motor que tenga un mayor torque y que proporcione una estructura más robusta para el generador de caracteres.

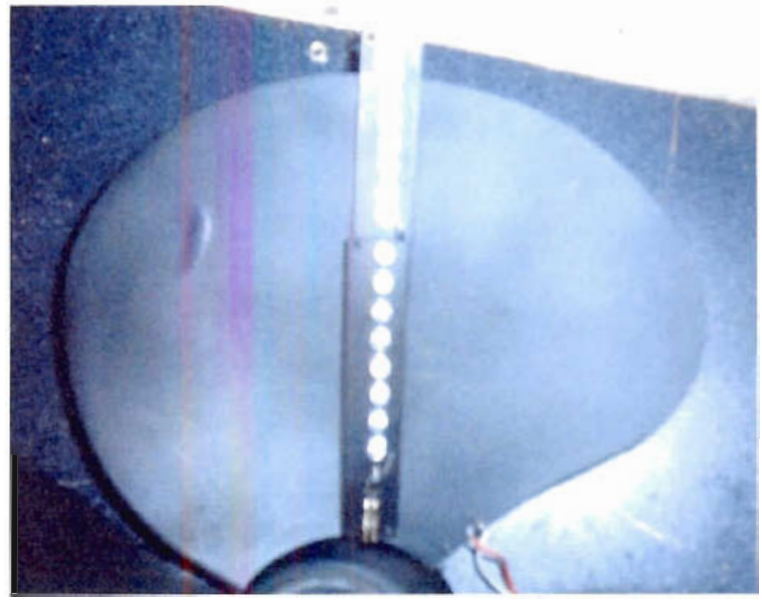
Como segunda opción, se escogió un ventilador de mesa utilizado para acondicionar el aire en las casas como el que se muestra en la figura 3.4. Estos ventiladores funcionan con voltaje de 120V AC, 60Hz. Este dispositivo brinda la facilidad de tener lista la estructura del generador de caracteres, es decir, tiene un soporte para toda la estructura y un lugar en donde colocar los LEDs, que puede ser en una de las aspas del ventilador, además de ser fácil de conseguir y de bajo costo.

Además de facilitar mecánicamente la implementación del generador de caracteres, el ventilador posee un control que le permite girar a tres velocidades diferentes, con las cuales se puede comprobar en la práctica, la necesidad de una velocidad de giro de 3000 rpm planteada en la sección 2.2.2 del capítulo anterior.

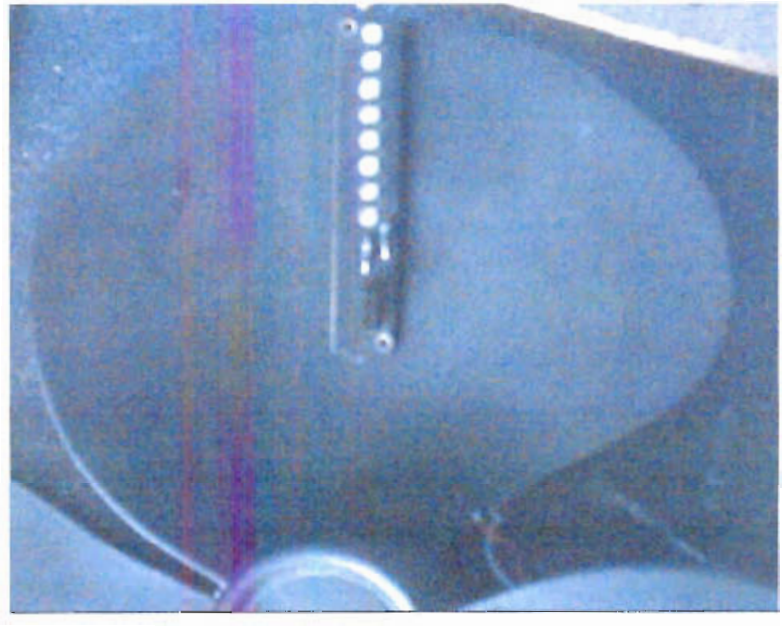
Figura 3.4 *Ventilador escogido como segunda opción para el generador de caracteres*

De la experiencia obtenida de la primera opción implementada, se decidió separar el circuito de control en dos partes para no alterar significativamente el equilibrio del peso en el motor del ventilador, la primera parte contiene todos los elementos y circuitos integrados que constituyen el sistema microprocesado, el circuito para determinar el origen, el circuito de recepción de infrarrojo para la comunicación con el generador de caracteres y el *driver* de los LEDs. La otra placa esta constituida únicamente por los LEDs. Al separar el circuito en dos partes, se puede ubicar los LEDs en una de las aspas del ventilador y la placa de control en el centro de giro del ventilador. De esta manera, se concentra la mayor parte del peso extra en la parte central y así se minimiza el desequilibrio que produce el acoplar estos componentes al ventilador.

Debido a las dimensiones del ventilador, no fue posible ubicar 16 LEDs en una de las aspas. En la figura 3.5 se puede apreciar que la fila de 16 LEDs sobrepasa el tamaño del aspa del ventilador, de modo que se trabajó con una fila de 8 LEDs y con el CI A6275.



a)



b)

Figura 3.5 *a) Fila de 16 LEDs acoplada al aspa del ventilador, b) Fila de 8 LEDs acoplada al aspa del ventilador.*

El circuito de control del generador de caracteres que incluye los cambios mencionados anteriormente se indica en la figura 3.6 y el circuito de la placa de LEDs se muestra en la figura 3.7. En este circuito se incluyen dos

diodos en serie a la fuente de voltaje para minimizar la disipación de potencia del CI A6275

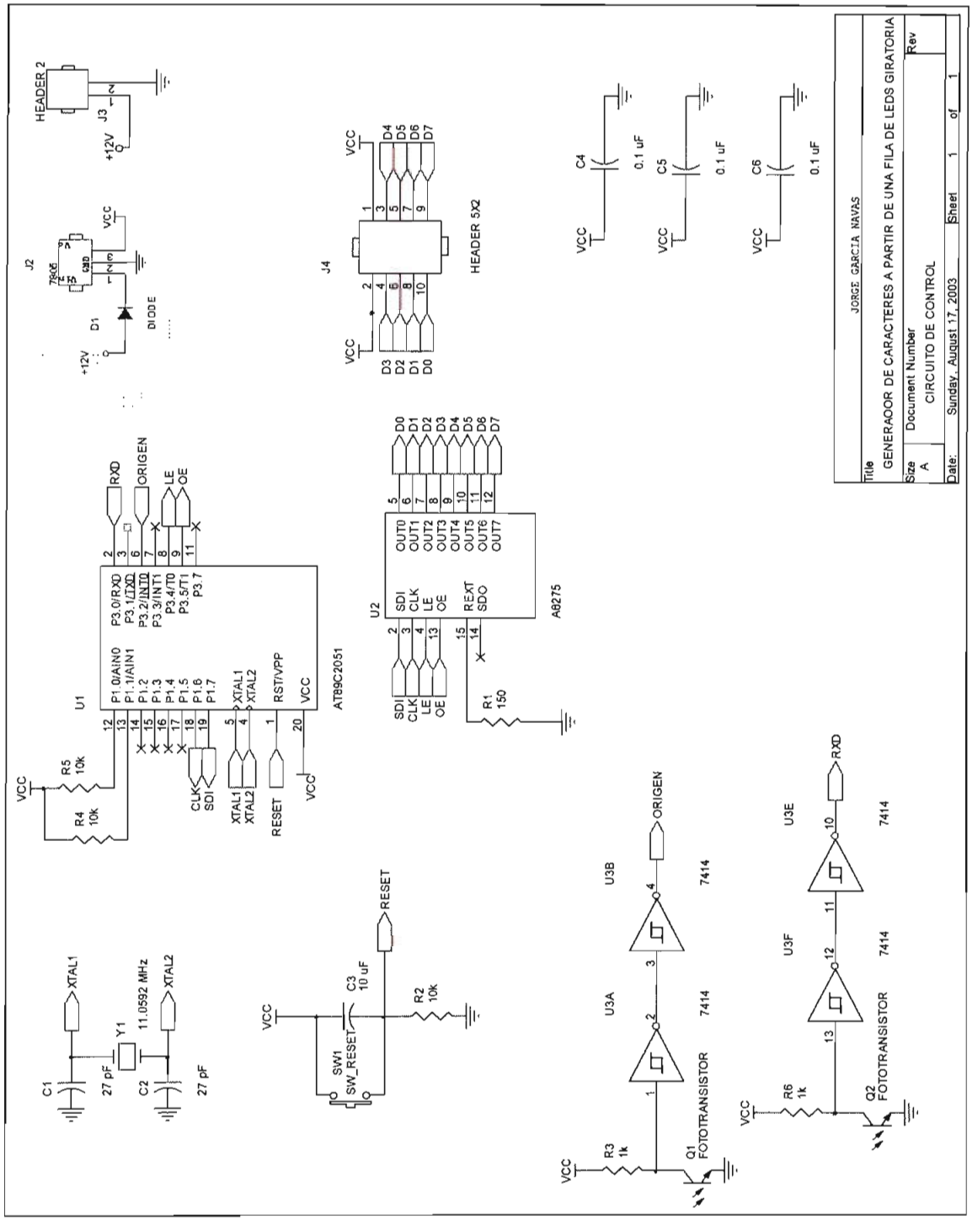


Figura 3.6 Circuito de control del generador de caracteres

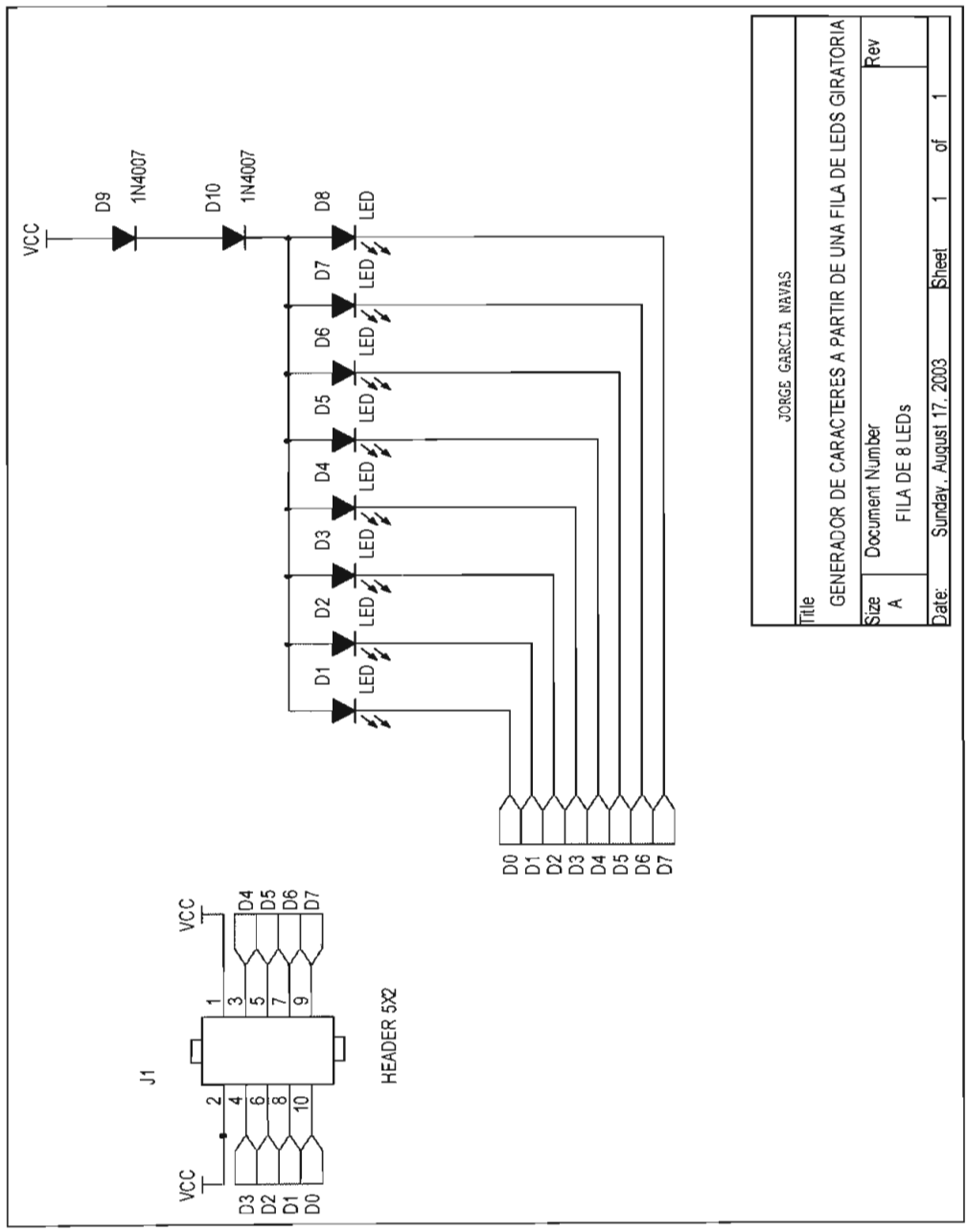
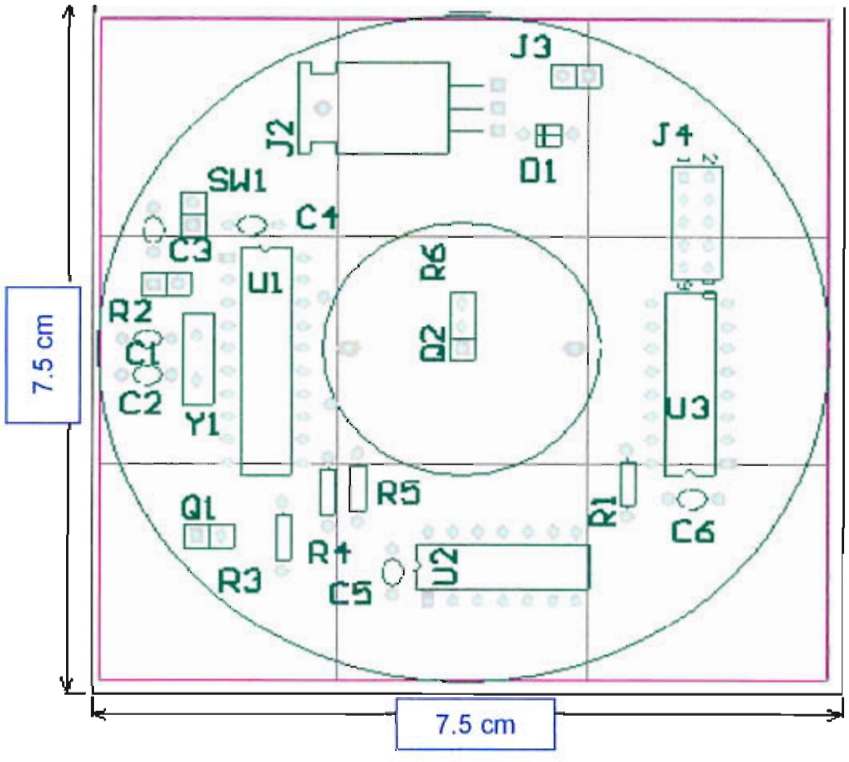
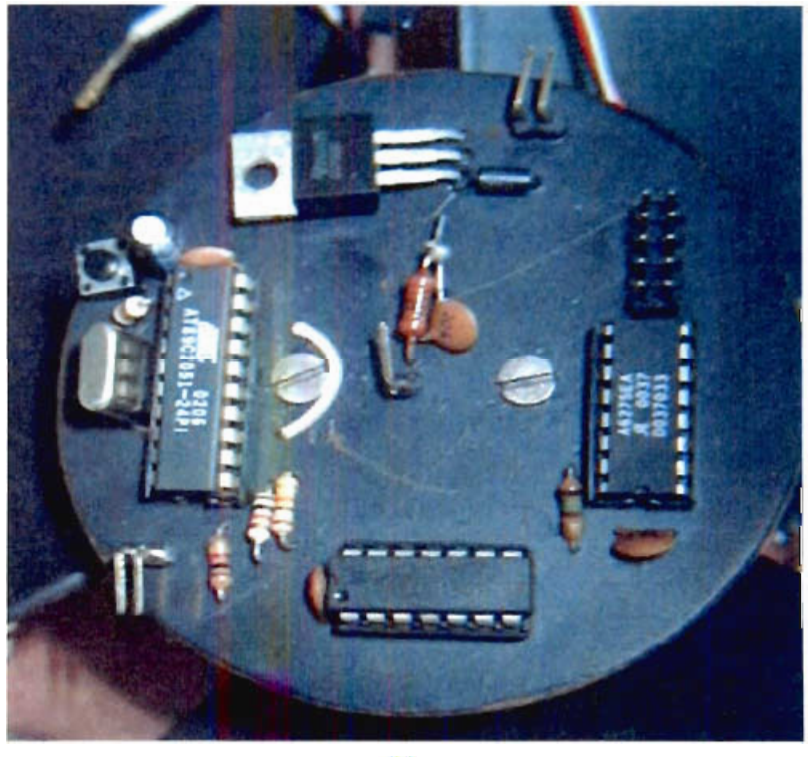


Figura 3.7 Fila de LEDs

En base a los circuitos de las figuras 3.6 y 3.7 se presenta los diseños de las nuevas tarjetas de circuito impreso a continuación:



a)



b)

Figura 3.8 *a) Circuito impreso de la tarjeta de control del generador de caracteres, b) tarjeta de control del generador de caracteres*

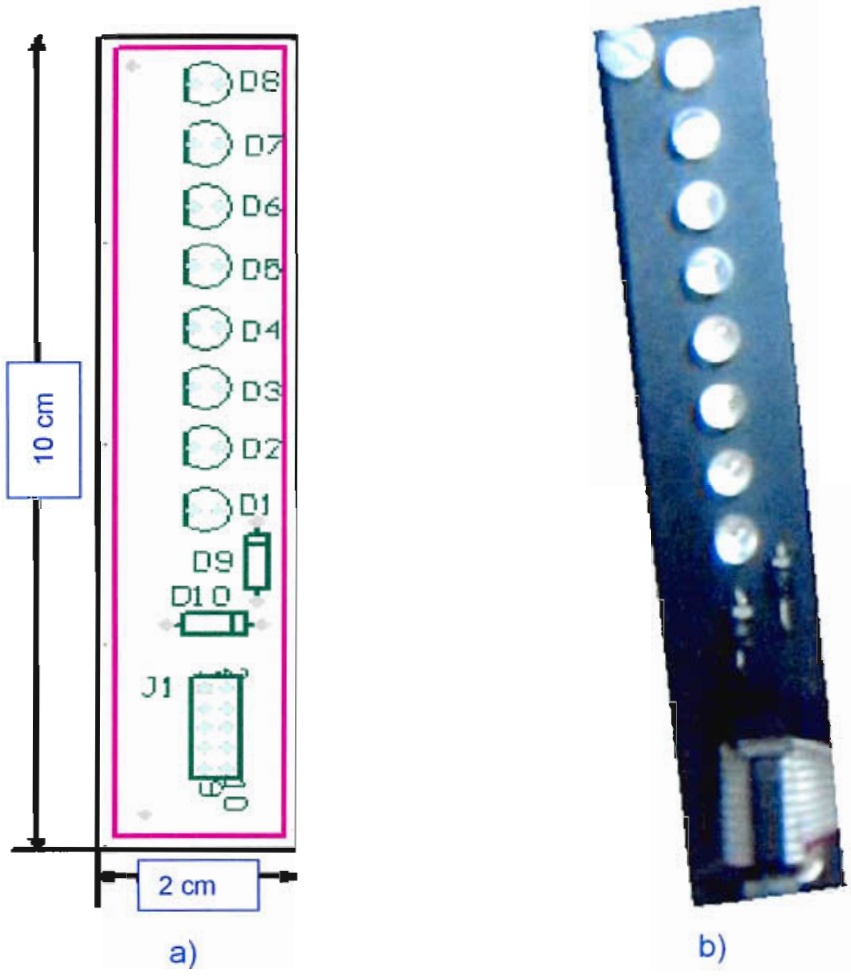
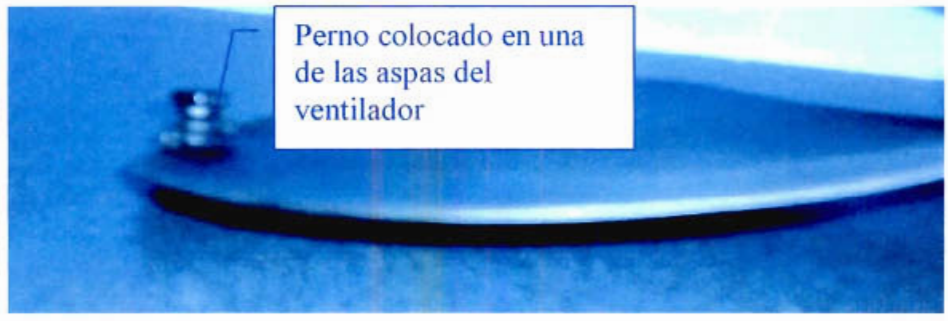
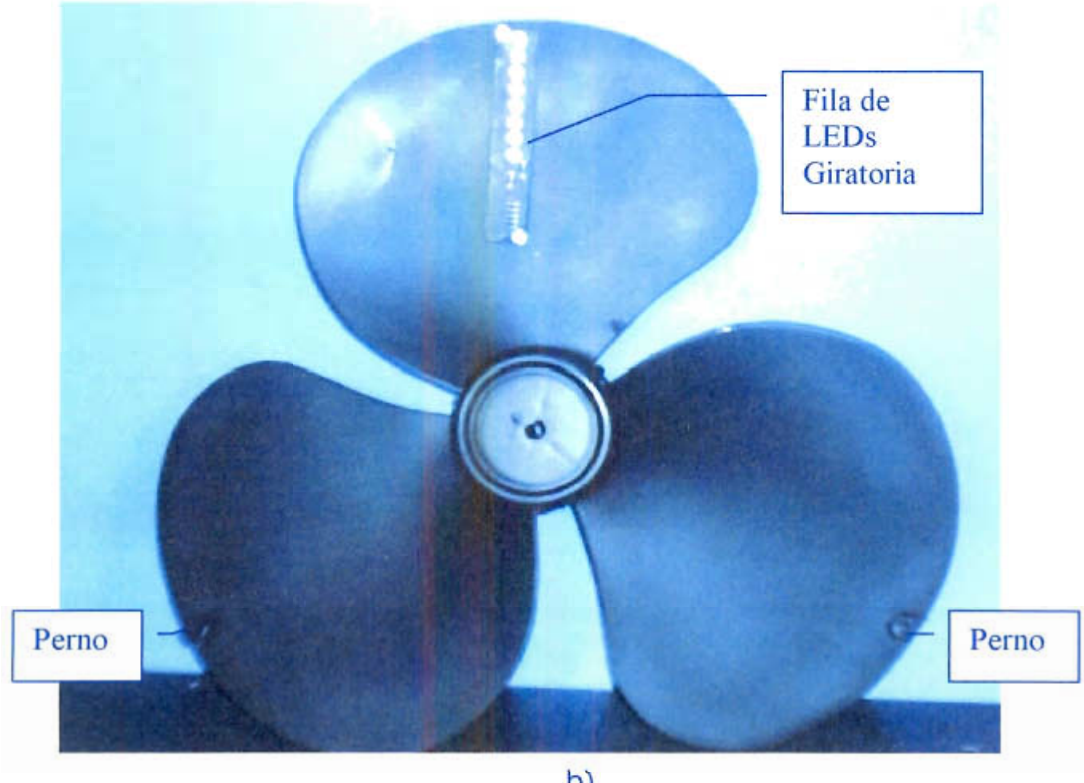


Figura 3.9 *a) Circuito impreso de la tarjeta de LEDs, b) tarjeta de LEDs*

Al colocar estas placas en el ventilador, se pudo comprobar que el desequilibrio producido por el peso de las mismas había disminuido y que la estructura del generador de caracteres era más robusta, sin embargo, aún existía vibración en el motor debido a este desequilibrio. Se colocaron pernos en las dos aspas restantes con el objetivo de que en cada una de ellas se tenga el mismo peso tal como se indica en la figura 3.10 a), en la figura 3.10 b) se puede observar la posición de la fila de LEDs y los pernos en las aspas del ventilador. De esta manera, se logró reducir la vibración producida.



a)



b)

Figura 3.10 *a) Perno colocado en una de las aspas del ventilador para equilibrar el peso generado por la tarjeta de LEDs, b) Posición de los pernos y de la fila de LEDs en las aspas del ventilador.*

3.1.2. FUENTE DE ALIMENTACIÓN

Una vez construida la estructura del generador de caracteres, se procedió a encontrar la forma de proporcionar una fuente de voltaje DC al circuito de control. Esta parte del sistema tiene una gran importancia, ya que debe asegurarse que el sistema este energizado en todo momento con el

problema de proporcionar una fuente de alimentación a un dispositivo móvil. En el diseño anterior, se ubicó una batería en el circuito de control, pero con esta solución la batería se descarga continuamente, por tanto se necesita cambiar la batería o recargarla periódicamente. Es mejor tener una fuente de voltaje que dependa de la red eléctrica. La solución planteada en el presente proyecto de titulación es construir un par de discos de material conductor, los cuales permanecen fijos en el ventilador y conectados a la fuente DC y acoplar en la parte móvil (aspas del ventilador) un par de conductores de carbón, los cuales están siempre en contacto con los discos fijos energizando a la parte móvil del generador de caracteres. El carbón ayuda a reducir la fricción que se produce al estar en contacto con la superficie de los discos. El regulador de voltaje incluido en la placa de control del generador de caracteres elimina cualquier efecto de rizado en la fuente. En la figura 3.11 se muestra el acoplamiento de los discos concéntricos elaborados en una baquelita de circuito impreso al ventilador y en la figura 3.12 se muestran los carbones acoplados a las aspas del ventilador en contacto con los discos conductores.

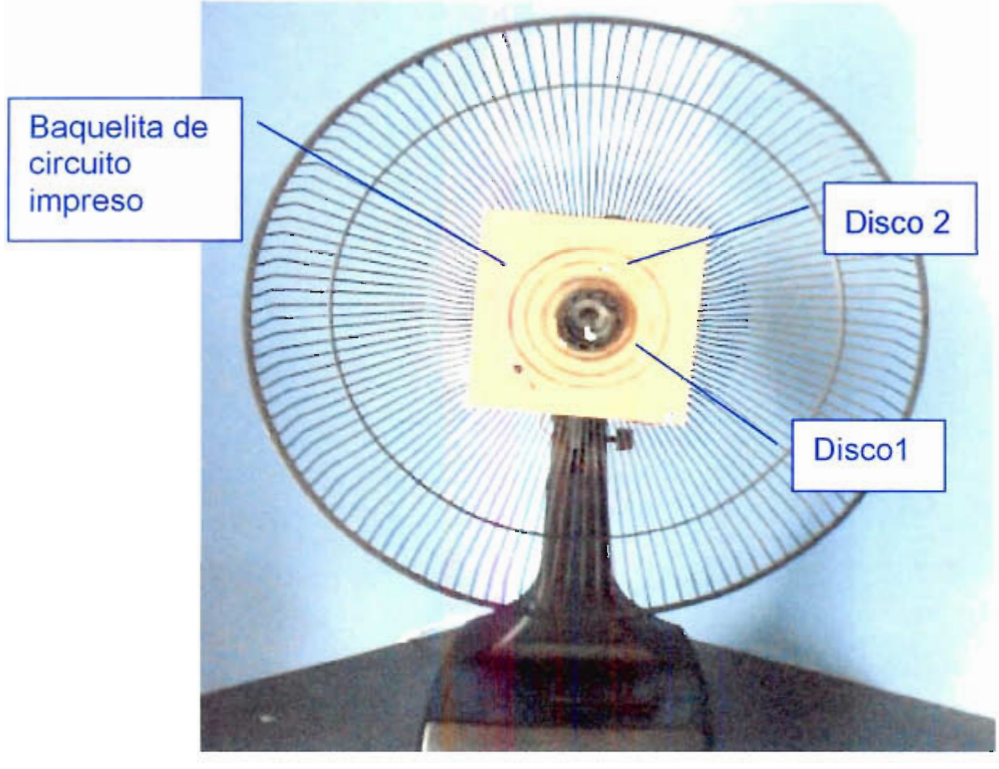


Figura 3.11 *Discos conductores acoplados al ventilador (Parte Fija).*

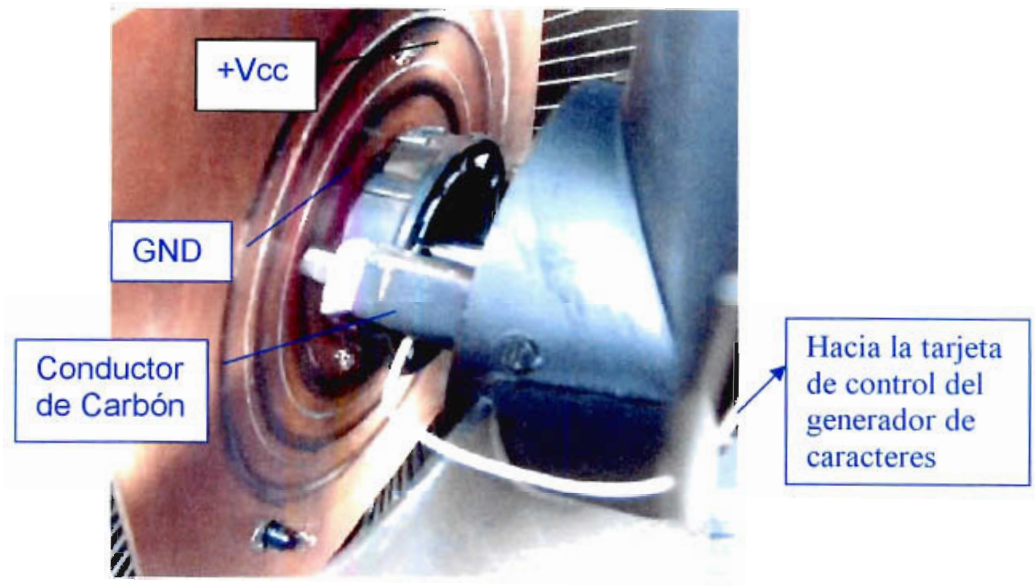


Figura 3.12 Circuito de alimentación DC.

3.1.3. CIRCUITO PARA DETERMINAR UN PUNTO DE ORIGEN

El circuito para la implementación del punto de origen desde el cual se empieza a pintar los caracteres está conformado por un fotodiodo y un fototransistor. Se colocó el fotodiodo en serie con una resistencia limitadora de corriente en la parte fija del ventilador, específicamente en la placa de alimentación DC y se encuentra siempre en conducción emitiendo luz infrarroja. En cambio, el fototransistor, funcionando en corte y saturación está ubicado en la parte móvil del sistema, en una de las aspas del ventilador y conectado a la tarjeta de control del generador de caracteres, de esta forma, al pasar frente al fotodiodo se forma el circuito de la figura 2.11 del capítulo anterior y el fototransistor se satura dando la interrupción al sistema microprocesado que indica que se ha pasado por el origen. En la figura 3.13 se muestra la placa de alimentación DC con el circuito del fotodiodo en conducción y el fototransistor colocado en una de las aspas del ventilador

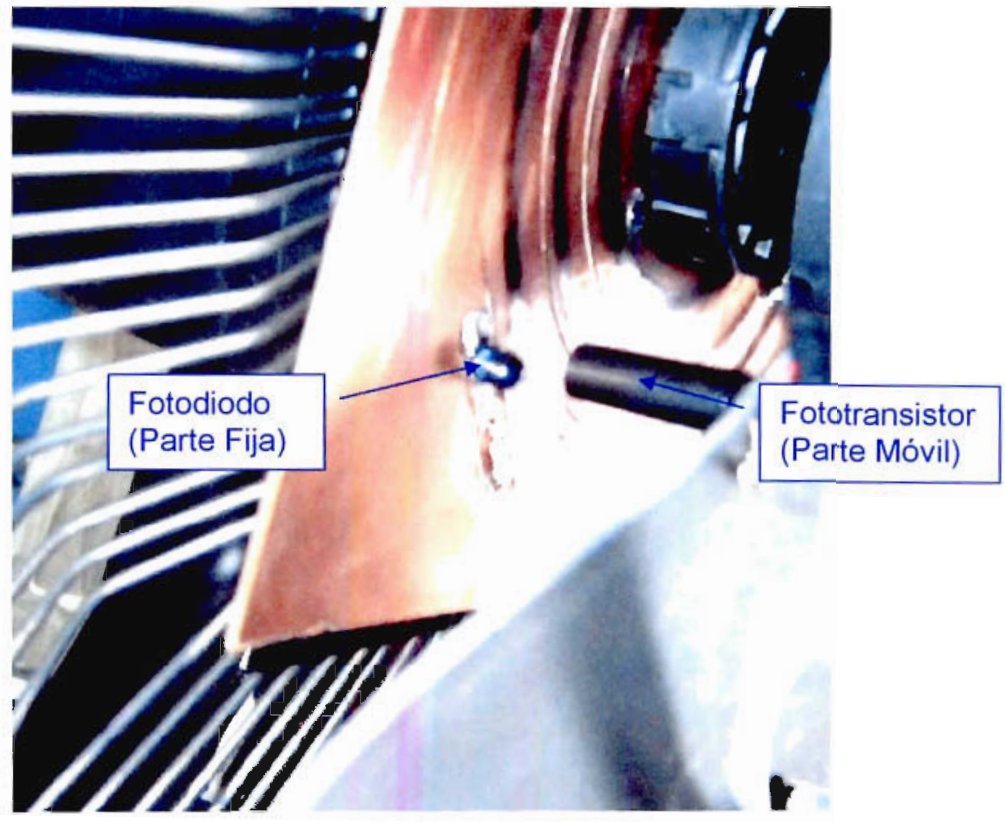


Figura 3.13 *Placa de alimentación DC con circuito indicador de cruce por el origen.*

3.1.4. DETERMINACION DE LA VELOCIDAD DE GIRO

Una vez construido el circuito de alimentación DC y ubicadas las placas en el ventilador se procede a determinar la velocidad de giro del ventilador. Como se mencionó anteriormente, el ventilador posee un control que le permite girar a tres velocidades diferentes, en principio, se determinó la velocidad mínima del ventilador para realizar el desarrollo del software para el microcontrolador, debido a que a esta velocidad se tiene una mayor estabilidad mecánica en el ventilador, es decir, el efecto producido por el desequilibrio del peso en las aspas provocado por los LEDs es menor, posteriormente se podrá escoger la velocidad de giro más adecuada para el funcionamiento correcto del generador de caracteres.

Utilizando el hardware existente, se pudo llegar a una aproximación de la velocidad de giro del ventilador implementando un software en el

microcontrolador que cuenta las vueltas que realiza la fila de LEDs durante un minuto, es decir, cada vez que la fila de LEDs pase por el origen se incrementa un contador de 16 bits. Esta cuenta se realiza durante un minuto y se muestra en los LEDs la velocidad del motor en revoluciones por minuto.

Al implementar este software, se llega a que la velocidad mínima del ventilador es aproximadamente de 600 rpm.

Con este dato, se pudo calcular el tiempo que demora la fila de LEDs en recorrer 3° que es el ángulo escogido en el capítulo anterior para obtener el *pitch* en el generador de caracteres.

Velocidad mínima del ventilador ≈ 600 rpm

Por lo tanto el tiempo que tarda el dar una revolución es:

$$t = \frac{1 \text{ min}}{600 \ rev} x \frac{60 \ s}{1 \text{ min}} = 0.1 \frac{s}{rev}$$

Luego, se puede determinar el tiempo para recorrer 3° de la siguiente manera:

$$t_{3\,grad} = 0.1 \frac{s}{rev} x \frac{1 rev}{360 \ °} x 3° = 833 \ us$$

Entonces, si los LEDs giran a una velocidad de 600 rpm se debe actualizar el dato en los LEDs cada 833 µs para poder obtener un *pitch* de 3°.

A continuación se calcula el ciclo de trabajo utilizando la figura 3.14 que indica la señal de corriente en un LED vs el tiempo. El LED recorre un ángulo de 3° en un tiempo T = 833 µs, el cual constituye el periodo de la señal de corriente en el LED, el cual permanece encendido durante un tiempo te.

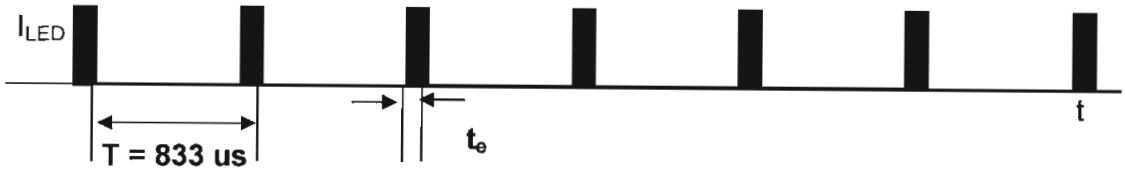


Figura 3.14 Cálculo del ciclo de trabajo

Como se menciona en el capítulo 1, el ciclo de trabajo está dado por la ecuación:

$$d = \frac{t_e}{T} \quad \text{(Ec. 3.1)}$$

La corriente efectiva que circula por el LED se calcula con la ecuación:

$$I_{efectiva} = \sqrt{d}.I_{máxima} \quad \text{(Ec. 3.2)}$$

Para obtener un buen nivel de brillo en los LEDs sin reducir su vida útil, se necesita que circule por ellos una corriente de 20mA (I_{media}). En el capítulo anterior se menciona que la corriente máxima que se va a obtener del *driver* es 80 mA ($I_{máxima}$).

Con estos valores, se obtiene el tiempo que el grupo de LEDs debe estar encendido con un determinado dato reemplazando la ecuación 3.1 en 3.2 con lo cual se tiene:

$$\frac{I_{máxima}}{I_{efectiva}} = \sqrt{\frac{T}{t_e}} \quad \text{(Ec. 3.3)}$$

Siendo:

T = Tiempo en que la fila de LEDs recorre un ángulo de 3°

Te = Tiempo en que un determinado dato permanece encendido

$I_{máxima}$ = Corriente que sale del *driver*

I_{media} = Valor medio de la corriente que circula por los LEDs

$$t_e = \frac{I_{efectiva}{}^2 \cdot T}{I_{máxima}{}^2} = \frac{(20\,mA)^2(833\,\mu s)}{(80\,mA)^2} \approx 52\,\mu s$$

Los valores de T y te son valores de referencia y permiten obtener caracteres visibles al realizar las pruebas. Posteriormente, estos valores podrán ser cambiados hasta obtener mejores imágenes de los caracteres.

De esta manera se puede obtener el máximo brillo posible de los LEDs para que la imagen de los caracteres pueda observarse claramente. Además, se pintó el ventilador y las tarjetas de circuito impreso de color negro con el fin de aumentar el contraste de la imagen en el generador de caracteres.

Con estas consideraciones se procedió a desarrollar el software para el microcontrolador.

3.1.5. DESARROLLO DEL SOFTWARE PARA EL MICROCONTROLADOR

El algoritmo diseñado para el generador de caracteres se describe de la siguiente manera:

Al iniciar el programa se inicializará el puerto serial del microcontrolador y el timer 1 para establecer comunicación con el computador a 1200 bps. El timer 0 se configura para dar una interrupción cada 800 μs y encender una bandera que indica que se debe pintar un nuevo dato en la fila de LEDs y los registros de memoria utilizados para pintar los caracteres se inicializan para que al principio el generador de caracteres aparezca apagado.

Cuando la fila de LEDs pasa por el origen se produce una interrupción que indica que se debe mostrar el primer caracter del mensaje, A partir de este origen se configura el temporizador del microcontrolador para generar interrupciones cada 800μs. Estas indican que se debe mostrar un nuevo dato en la fila de LEDs. Este dato permanece en los LEDs durante 100μs para luego apagarlos y pasar a esperar la siguiente interrupción del temporizador.

En cualquier momento se puede activar la interrupción del puerto serial para actualizar una tabla de datos que contiene los códigos ASCII del mensaje que se muestra en el generador de caracteres.

A continuación se espera a que la fila de LEDs pase por el origen para comenzar a pintar los caracteres. Una vez que se ha pasado por el origen arranca el *timer* 0 y cada vez que se desborda y genera una interrupción se pinta un nuevo dato que corresponde al caracter que se está mostrando en los LEDs en un instante dado. Los datos que se utilizan para dibujar los caracteres se encuentran almacenados en una tabla en la memoria de programa y se forman dibujando los caracteres en una matriz de 8 x 5 pixels. Se realizó un mapeo de todos los caracteres ASCII asignando un 1 lógico en la posición del LED que esté encendido y un 0 lógico en la posición del LED que esté apagado. En la figura 3.15 se muestra un ejemplo de este mapeo en donde se obtiene los datos que deben guardarse en la tabla de datos correspondientes a la letra "A".

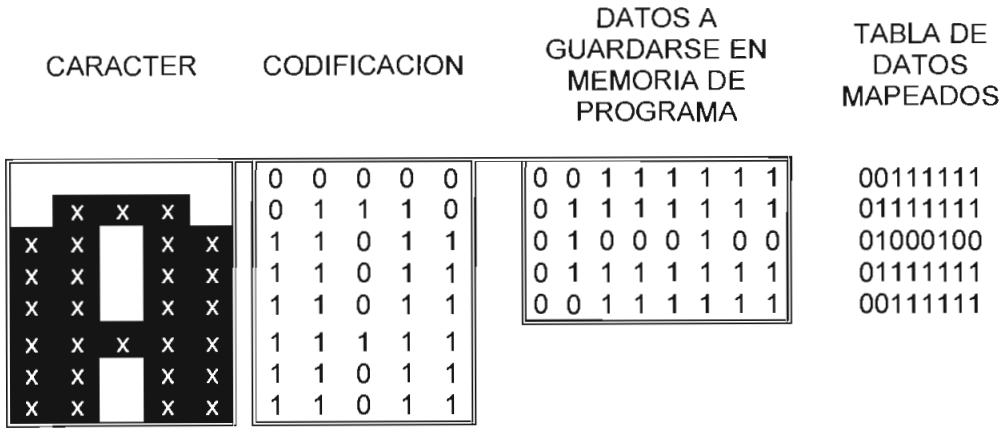


Figura 3.15 *Ejemplo de mapeo de la tabla de datos de los caracteres ASCII*

El mapeo de todos los caracteres ASCII se puede observar en el anexo B del presente proyecto de titulación.

En el momento que el usuario transmita datos al tablero utilizando el interfaz serial de comunicaciones, se producirá una interrupción, en la cual se actualizarán los registros de memoria RAM interna utilizados para guardar los caracteres ASCII que se muestran en el generador de caracteres.

Una vez descrito el algoritmo a desarrollarse, en la figura 3.16 se presenta el diagrama de flujo del programa principal y en la figura 3.17 se ilustran los diagramas de flujo de las subrutinas de interrupción del programa implementado en el microcontrolador.

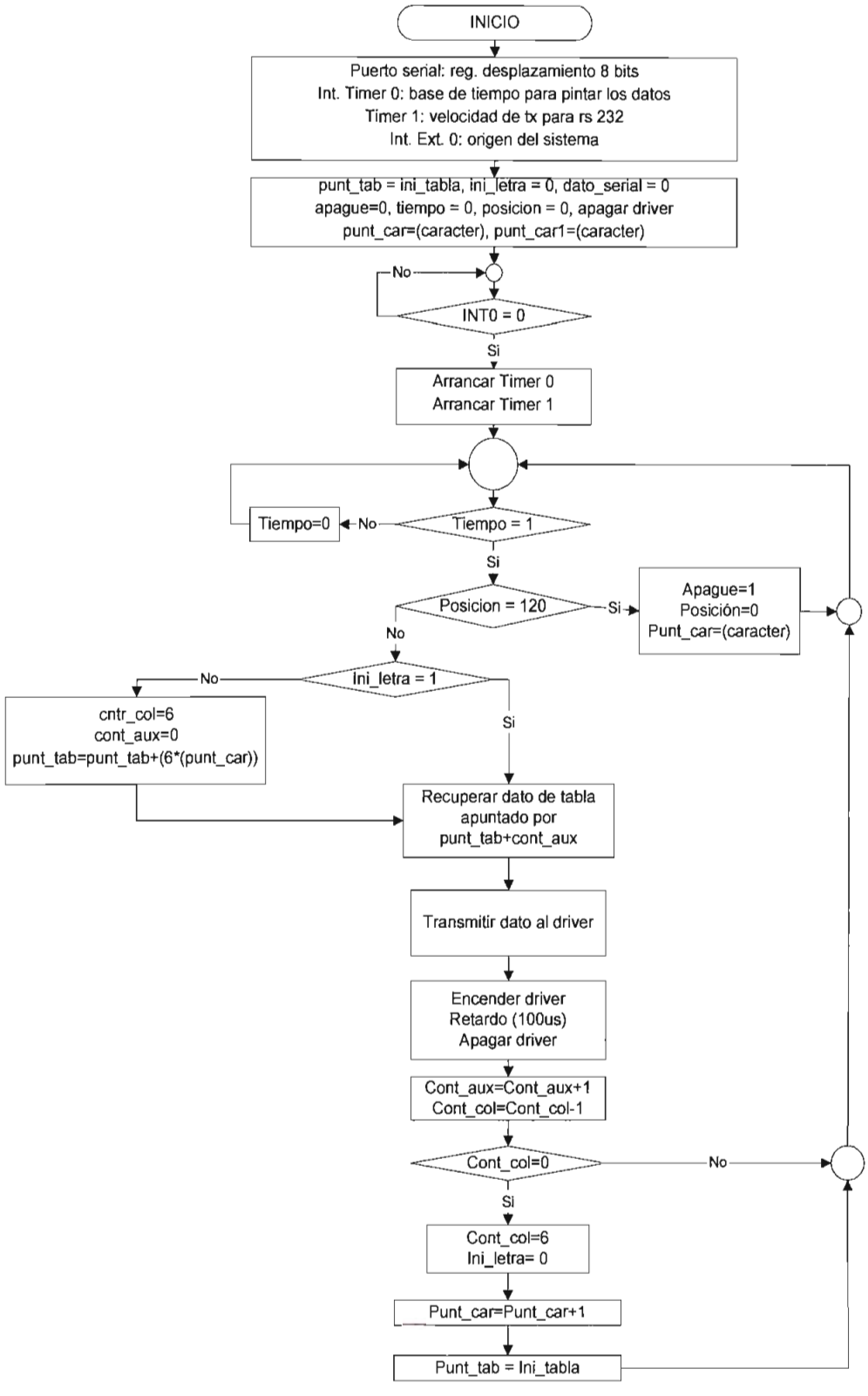


Figura 3.16 Diagrama de flujo del programa principal

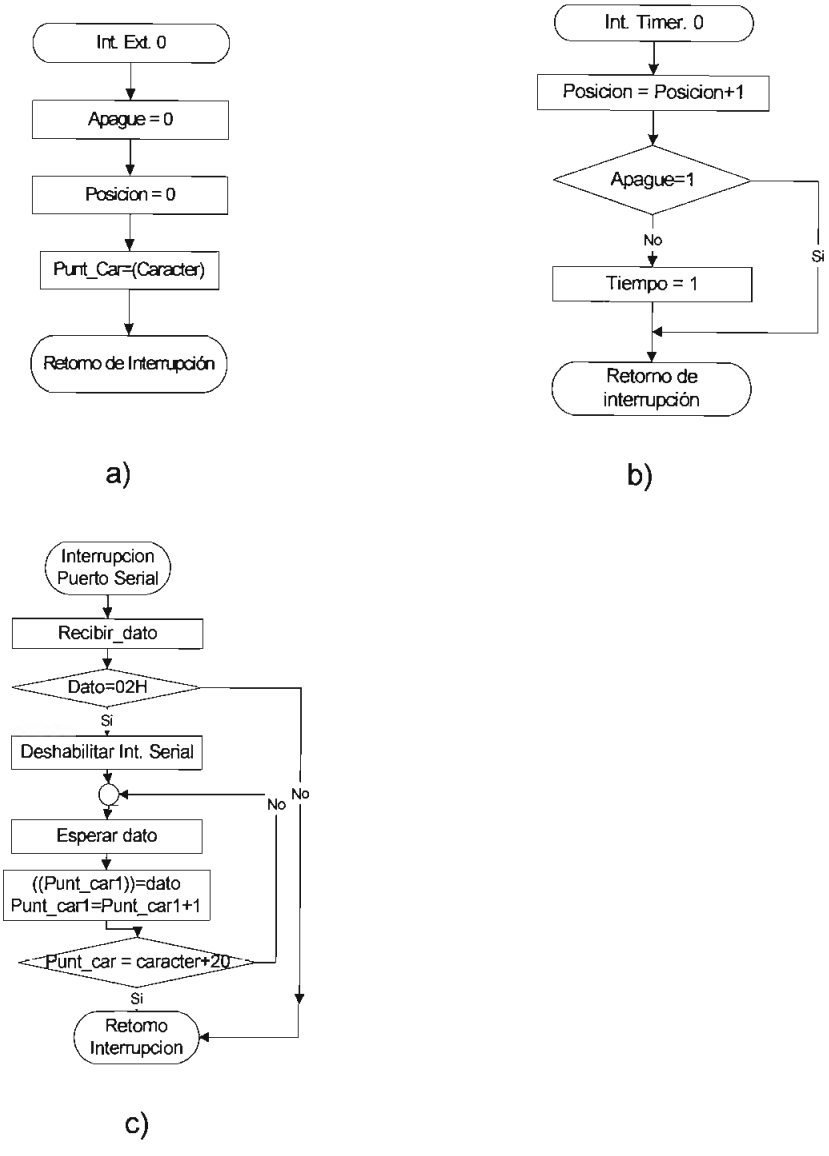


Figura 3.17 *Diagrama de flujo de las subrutinas de interrupción: a) Interrupción externa 0, b) Interrupción del Timer 0, c) Interrupción del puerto serial*

Se realizaron las pruebas del programa implementado inicializando al generador de caracteres con un texto por defecto que se muestra al encenderse. En la figura 3.18 se muestra el texto "ESCUELA POLITECNICA" programado para que se muestre al encender el generador de caracteres.

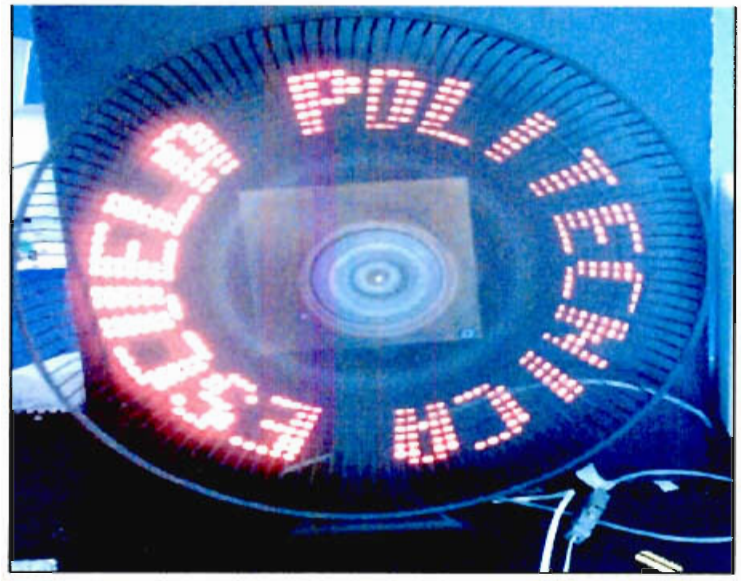


Figura 3.18 *Texto programado para mostrarse al encender el generador de caracteres*

3.1.6. INTERFAZ Y PROTOCOLO DE COMUNICACIÓN SERIAL CON EL GENERADOR DE CARACTERES.

Para establecer la comunicación con el generador de caracteres se utilizó como primera opción el circuito de la figura 2.12 para implementar un control remoto. El fototransistor se ubicó en el centro de giro con el fin de que su posición no cambie a medida que la tarjeta vaya girando.

Se implementó una comunicación de tipo serial RS232 con niveles de voltaje TTL, tal como se menciona en la sección 2.2.7 del capítulo anterior.

Al implementar el circuito de la figura 2.12 se pudo observar que el circuito funciona a una distancia muy pequeña de aproximadamente 10 cm y que el ángulo del haz de luz infrarroja es pequeño, por lo que se necesita apuntar el fotodiodo con bastante exactitud.

Existen en el mercado módulos detectores de infrarrojo diseñados exclusivamente para implementar circuitos de control remoto, Como segunda opción para el interfaz de comunicación se utilizó el módulo IRM – 8420, que es un detector de infrarrojo muy sensible, el cual permite implementar un canal de

comunicaciones inalámbrico utilizando luz infrarroja. Tiene la propiedad de poder captar luz en el espectro infrarrojo, modulada a 37.9 kHz. Es decir, si en el transmisor se tiene un fotodiodo encendiéndose y apagándose a esa frecuencia, a la salida del receptor se tendrá un 0L, de lo contrario, a la salida se tendrá un 1L.

Con estas características, el funcionamiento del circuito transmisor está compuesto de un oscilador a 37.9 kHz cuya señal de salida enciende y apaga un fotodiodo a esa frecuencia cuando en los datos se tenga un 0L y mantiene el fotodiodo apagado cuando en la señal de datos se tenga un 1L. Este circuito se muestra en la figura 3.19.

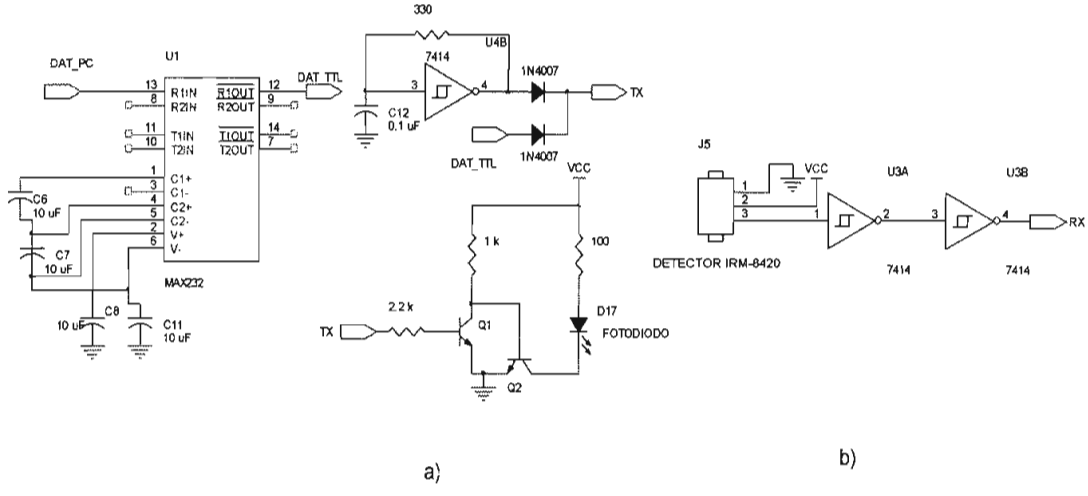


Figura 3.19 Circuito de Control Remoto a) Transmisión, b) Recepción.

Al probar este circuito en el generador de caracteres se presentaron una serie de problemas en la recepción de datos debido a que al hacer girar la fila de LEDs se introduce ruido y por lo tanto se produce la detección de datos errados.

En vista de que el sistema para implementar la fuente de alimentación DC funcionó correctamente, se decide implementar la segunda opción planteada en el numeral 2.3.5 del capítulo anterior para implementar el interfaz de comunicaciones. Esta solución consiste en agregar un tercer disco

conductor en el circuito de la fuente de voltaje y un tercer conductor de carbón en la parte móvil del ventilador, los datos se conectan a este tercer disco conductor y a través del conductor de carbón pasan al puerto serial del microcontrolador.

Al probar esta opción, se pudo recibir datos a 1200 bps en el generador de caracteres sin ningún inconveniente.

El protocolo de comunicaciones se planteó de la siguiente manera:

- Para enviar un nuevo texto al generador de caracteres, se antepone el código 02H luego del cual se envían los 20 caracteres a mostrarse.

- Para reiniciar el puntero que guarda los 20 caracteres que se muestran en las localidades de memoria interna del microcontrolador se enviará el código 07H

Para convertir los niveles de voltaje de RS232 a niveles TTL se utilizó el C.I. MAX232 y se construyó una tercera placa de circuito impreso cuyo diseño se muestra en la figura 3.20 a) y la placa construida se muestra en la figura 3.20b).

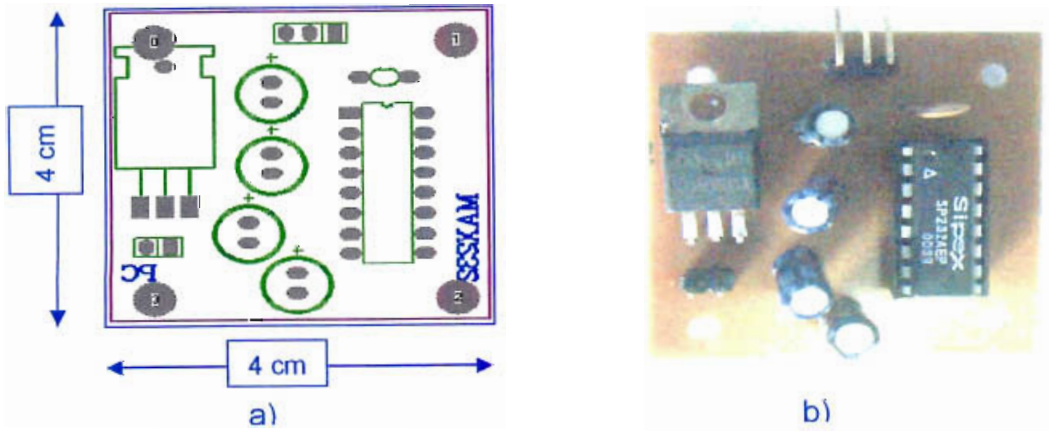


Figura 3.20 *a) Circuito Impreso de la tarjeta de recepción de datos desde el PC, b) Tarjeta de recepción de datos desde el PC*

3.2. DESARROLLO DEL PROGRAMA DE INTERFAZ ENTRE EL USUARIO Y EL GENERADOR DE CARACTERES

Como se menciona en la sección 2.3 del capítulo anterior, se utilizó un computador para implementar de manera fácil y rápida un dispositivo que permita actualizar los mensajes que se muestren en el generador de caracteres. Se escogió Visual Basic como lenguaje de programación ya que permite generar en forma sencilla aplicaciones amigables para el usuario en un entorno gráfico bajo Windows.

El programa a desarrollarse tendrá las características enumeradas en el numeral 2.3 del capítulo anterior.

3.2.1. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL PROGRAMA.

Se presenta en pantalla un cuadro de texto que permite modificar los mensajes que se presenten y un cuadro de gráfico para presentar la simulación de cómo se verá el texto en el tablero. El programa utiliza la misma tabla de datos que se encuentra en la memoria de programa del microcontrolador y los datos se pintan con una separación de 3 grados.

El programa puede transmitir los datos por el puerto serial del PC. Se utiliza el puerto COM 1 a una velocidad de 1200 bps debido a que la línea de datos que llega al generador de caracteres se encuentra en constante movimiento y se necesita minimizar los problemas de ruido que pueden producirse por este movimiento.

El programa se presenta en una única ventana la cual se muestra en la figura 3.21.

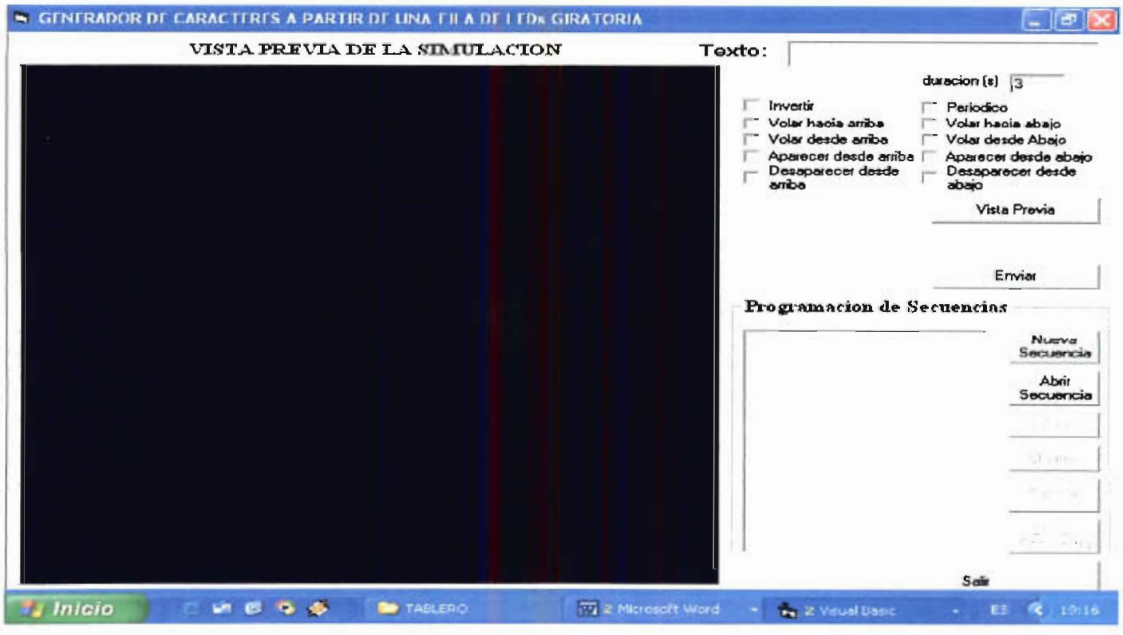


Figura 3.21 Programa de interfaz con el usuario

Para simular el texto que va a presentarse en el generador de caracteres, se escribe el mensaje en el cuadro de texto identificado con la etiqueta TEXTO y al presionar el botón VISTA PREVIA se dibujan los caracteres de acuerdo al texto que se escribe tal como se indica en la figura 3.22.

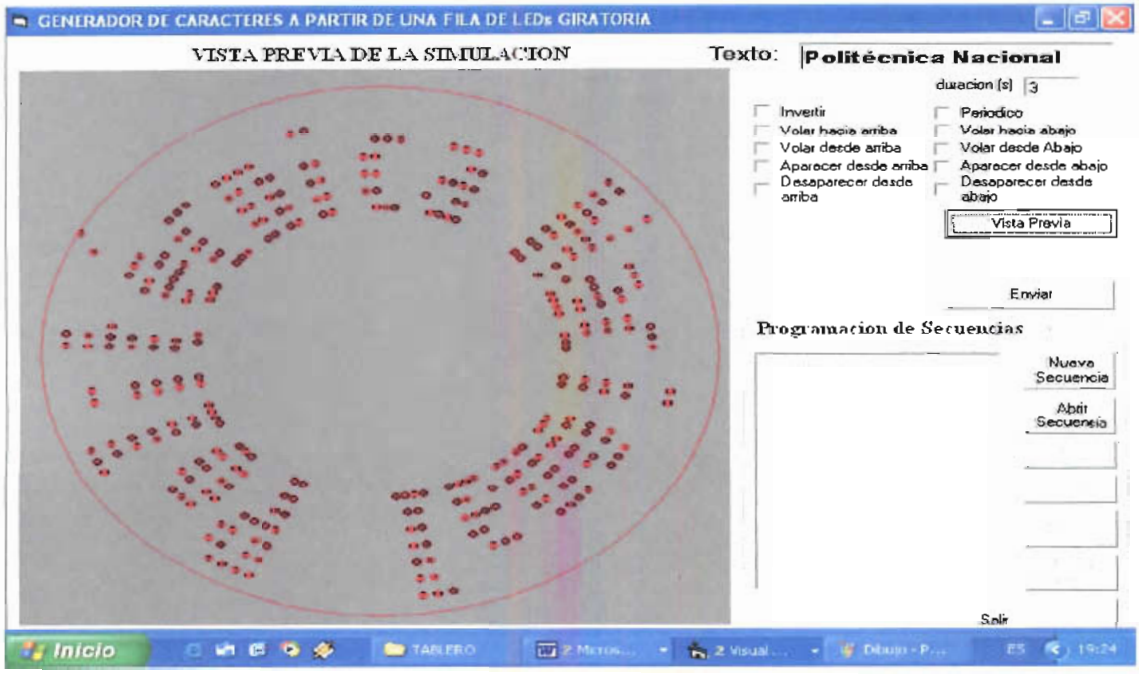


Figura 3.22 Simulación de los mensajes

Adicionalmente, se han programado diferentes efectos de animación en los textos con el fin de despertar un mayor interés en las personas que observen el generador de caracteres. Estos efectos se activan mediante las casillas de verificación indicadas en las figura 3.23

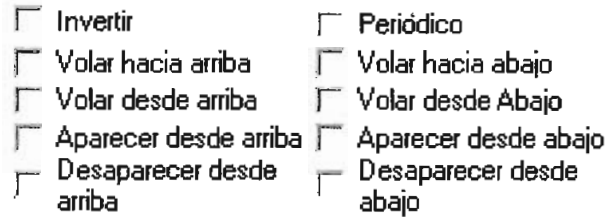


Figura 3.23 Efectos de animación.

En la tabla 3.1 se describen cada una de las animaciones y el código asignado para activar a cada ua de ellas en el generador de caracteres:

ANIMACION	CODIGO	DESCRIPCION
INVERTIR	80H	Los datos se pintan complementados
PERIODICO	10H	El texto va desplazándose a lo largo del espacio en que se pintan los caracteres
VOLAR HACIA ARRIBA	48H	Todo el texto se recorre hacia o desde la dirección indicada
VOLAR HACIA ABAJO	44H	
VOLAR DESDE ARRIBA	42H	
VOLAR DESDE ABAJO	41H	
APARECER DESDE ARRIBA	28H	El texto va apareciendo o desapareciendo desde la dirección indicada
APARECER DESDE ABAJO	24H	
DESAPARECER DESDE ARRIBA	22H	
DESAPARECER DESDE ABAJO	21H	

Tabla 3.1 Descripción de las animaciones

En las figuras 3.24, 3.25, 3.26, 3.27 y .3.28 se pueden observar algunas simulaciones de estas animaciones.

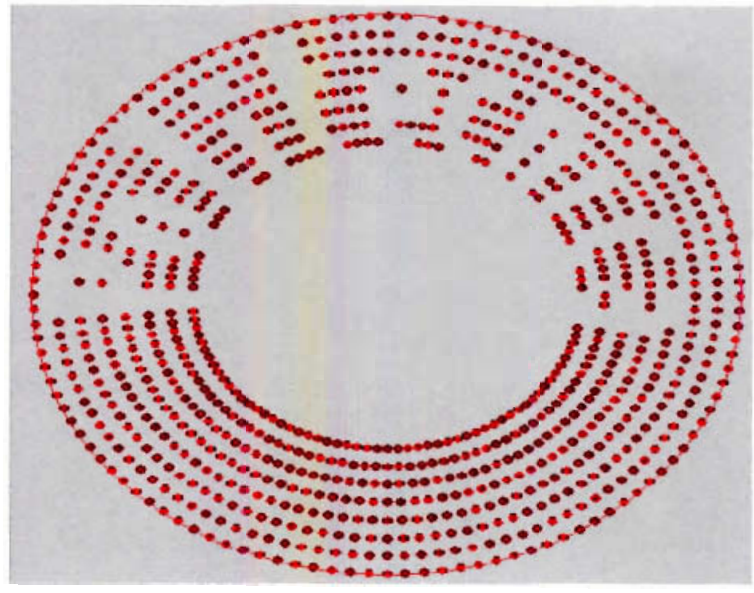


Figura 3.24 *Simulación de animación Invertir.*

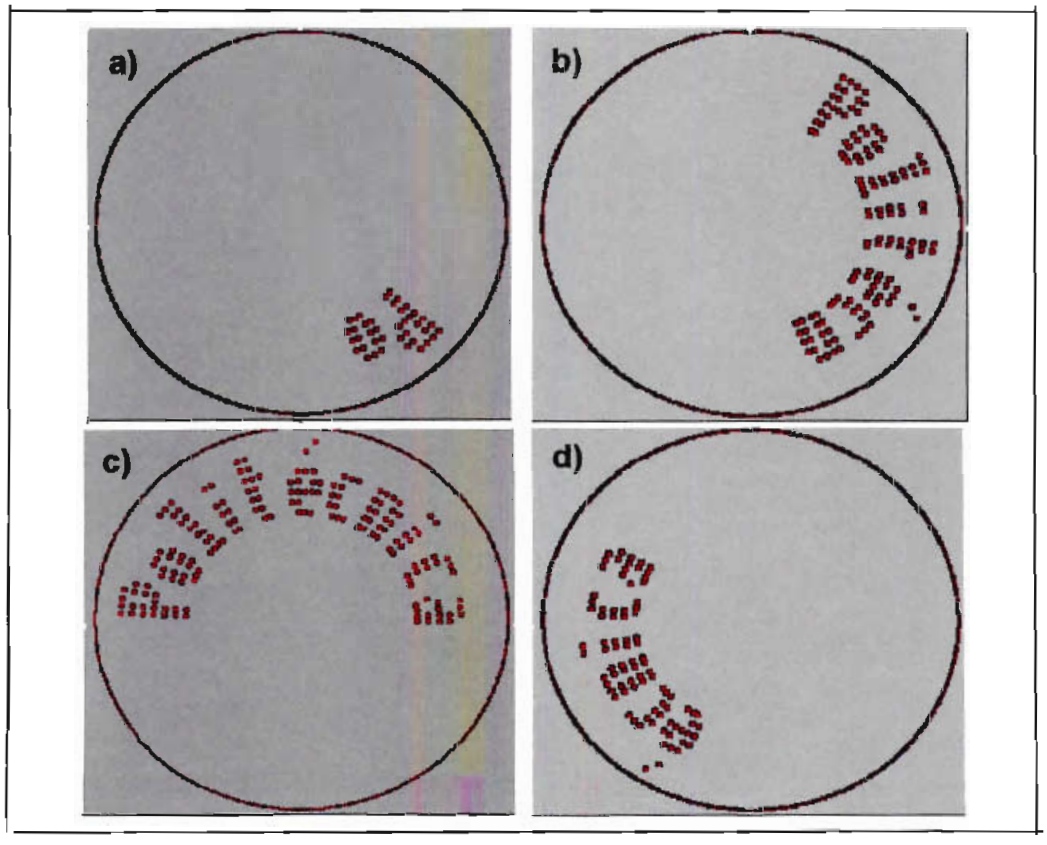


Figura 3.25 *Simulación de animación Periódico*

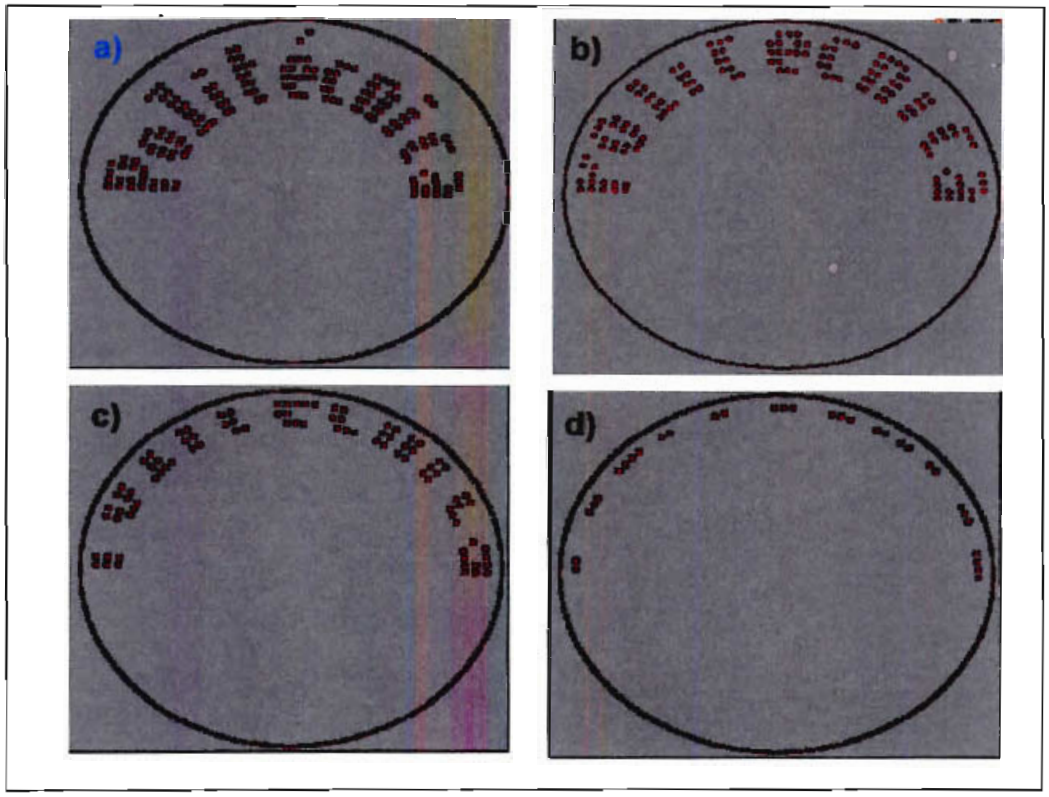


Figura 3.26 Simulación de animación Volar Hacia Arriba

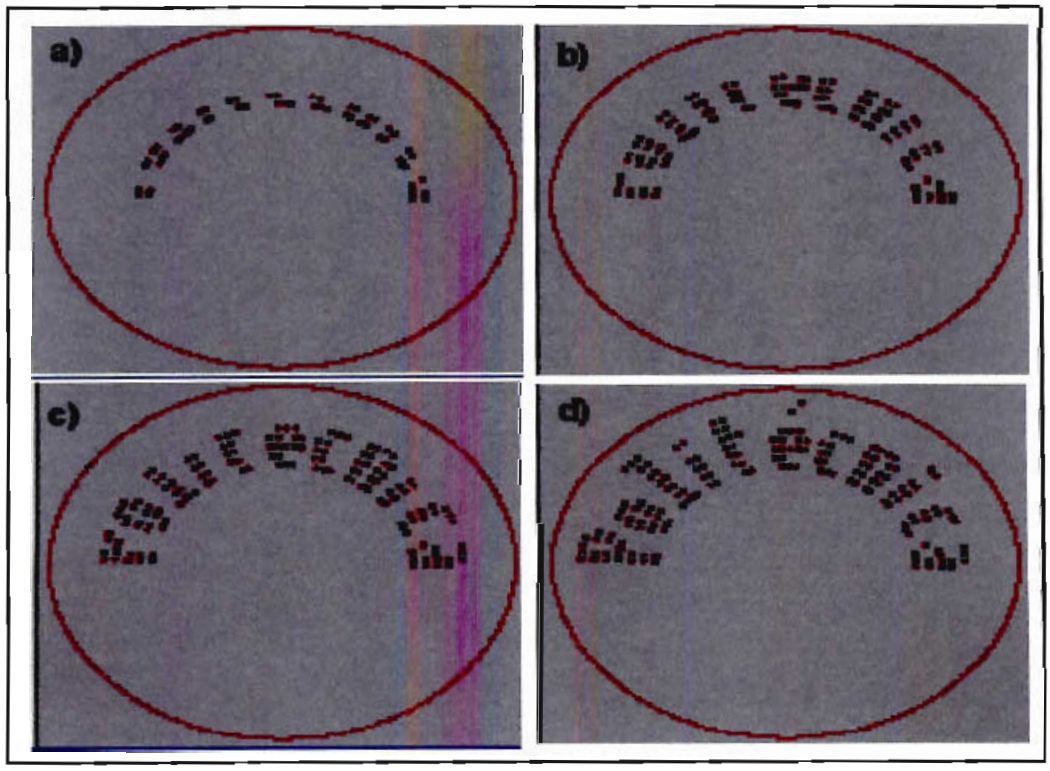


Figura 3.27 Simulación de animación Aparecer desde Abajo

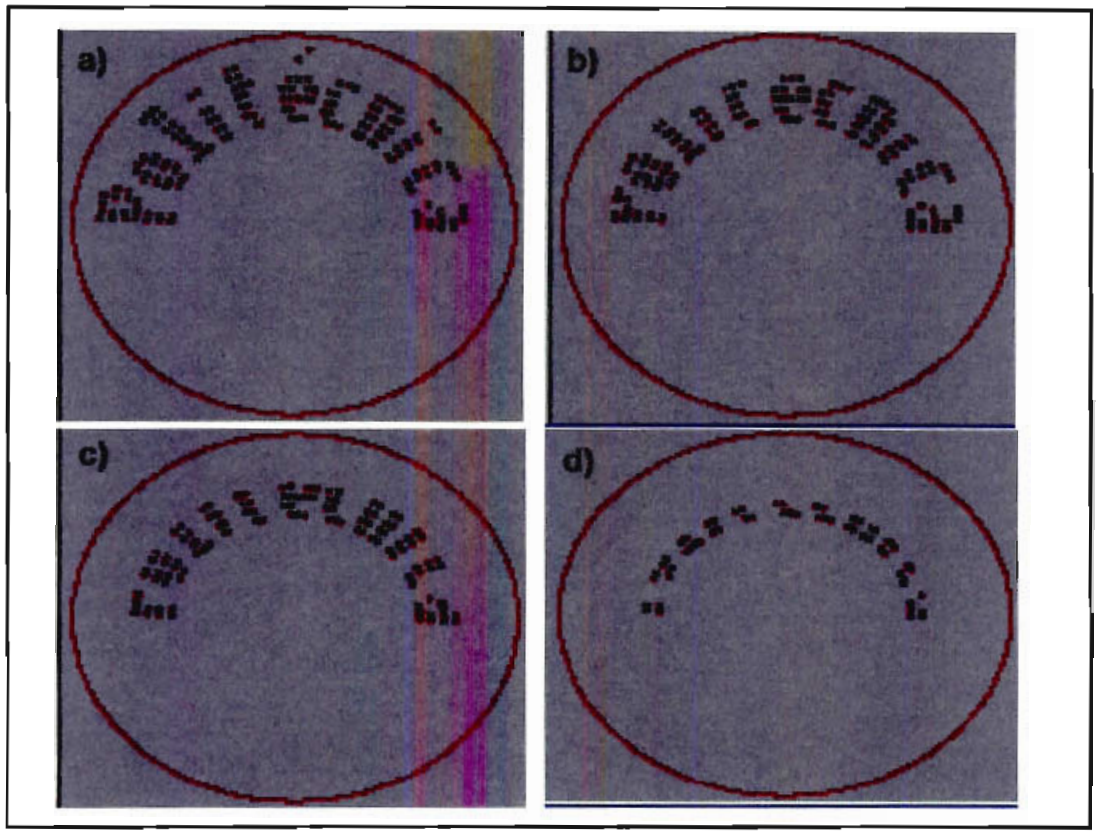


Figura 3.28 Simulación de animación Desaparecer desde Arriba

Estas animaciones se han implementado por software en el microcontrolador que maneja a la fila de LEDs giratoria y se añade al protocolo de comunicaciones un código asignado a cada animación anteponiendo el código 05H. De esta manera se puede cambiar la animación del texto que se esté mostrando en el generador de caracteres.

Además, se tiene la opción de enviar secuencias de mensajes y enviarlas al generador de caracteres desde el computador. En la figura 3.29 se muestran los cuadros de texto y botones de comando asignados para esta tarea.

Figura 3.29 Programación de secuencias para el generador de caracteres

Para programar una secuencia de mensajes se crea un archivo de datos con extensión *.dat mediante el botón de comando NUEVA SECUENCIA. Cada mensaje que se programe puede ser simulado en el programa y podrá formar parte del archivo de secuencia al presionar el botón AÑADIR. También se puede remover del archivo el último mensaje añadido al pulsar el botón ELIMINAR. En el cuadro de texto que se muestra en la figura 3.29 se observan los mensajes que forman parte de la secuencia seguidos de datos de control (encerrados por el símbolo "|") los cuales indican que animación se ejecuta y el tiempo de duración de cada mensaje. Este tiempo en segundos se configura en el cuadro de texto identificado con la etiqueta DURACION, una vez enviado el mensaje al generador de caracteres, el programa transmite el nuevo mensaje luego de transcurrir este tiempo.

Finalmente, el programa permite simular la secuencia pulsando el botón de comando SIMULAR.

Los archivos creados se pueden recuperar en cualquier momento mediante el botón ABRIR SECUENCIA, el cual permite abrir cualquier archivo con extensión *.dat.

Para enviar los mensajes al generador de caracteres se utiliza el botón ENVIAR y las secuencias podrán ser enviadas pulsando el botón ENVIAR SECUENCIA.

CAPITULO 4

RESULTADOS Y APLICACIONES

En el presente capitulo se describen las pruebas realizadas al generador de caracteres y los resultados de las mismas, y además se analizan las posibles aplicaciones para este dispositivo.

4.1. PRUEBAS REALIZADAS

Las pruebas realizadas en el generador de caracteres servirán para determinar la calidad de las imágenes que se muestren en el mismo y las facilidades que ofrece para editar los mensajes que se muestren. Luego de analizar los resultados podrán plantearse modificaciones en el hardware o software implementados con el objetivo de mejorar las características antes mencionadas.

4.1.1. VELOCIDAD DE GIRO

Como se menciona en el capítulo anterior, se utilizó la menor velocidad de giro de las tres que ofrece el ventilador para mostrar los caracteres, debido a que a esta velocidad el funcionamiento de la parte mecánica del generador de caracteres era mucho más estable. Al realizar las pruebas se pudo observar que las imágenes de los caracteres titilaban al mostrarse en el generador de caracteres, razón por la cual se procedió a realizar las pruebas a la velocidad intermedia, por lo tanto, fue necesario modificar el software del

microcontrolador para ajustar el tiempo en el cual la fila de LEDs giratoria recorre un ángulo de 3°.

Se ajustó el tiempo en 516 µs y a partir de este dato se pudo calcular un valor aproximado de la nueva velocidad de giro del motor de la siguiente manera:

$$\frac{3°}{516\,\mu s} x \frac{1x10^{6}\,\mu s}{1 s} x \frac{60\,s}{1\,\text{min}} x \frac{1 rev}{360°} \approx 970\,\frac{rev}{\text{min}}$$

Una vez implementada esta variación, se realizaron las pruebas nuevamente pudiendo comprobar que los caracteres dejaron de titilar.

Con esto, se desmiente la hipótesis planteada en la sección 2.2.2 del capítulo 2 en la que se plantea que el motor debería girar a una velocidad mínima de 3000 rpm. Esto se debe a que el ojo humano no distingue cuando una imagen cambia si ésta lo hace a razón de 24 cuadros por segundo o más. Es decir, si la velocidad de giro es de 24 revoluciones por segundo (1440 rpm) o mayor desaparece el efecto de titilado en los LEDs.

Al probar la velocidad más alta, se produjo una vibración muy fuerte en el ventilador, resultado del desequilibrio que producen los LEDs y la tarjeta de control del generador de caracteres en el ventilador, que a esa velocidad llega a tener una influencia mayor, por lo tanto, se utilizó la velocidad de giro intermedia para realizar el resto de pruebas en el generador de caracteres.

4.1.2. RESISTENCIA DEL MOTOR

Luego de 20 minutos de operación continua, el motor utilizado tuvo una avería debido al esfuerzo extra que realizaba por el peso de los LEDs. Por esta razón, se cambió el ventilador utilizado por un ventilador industrial con un motor más grande repitiendo el proceso descrito en el capítulo 3 y poniendo mayor cuidado en equilibrar el peso extra que producen los LEDs puestos en las aspas del ventilador. La velocidad mínima de este nuevo ventilador es de 1200rpm

4.1.3. SEPARACION ENTRE *PIXELS (PITCH)*

En el capítulo 2, en las figuras 2.7,2.8, y 2.9 se muestran las gráficas de los datos a mostrarse en el generador de caracteres con ángulos de separación de 2. 3 y 4 grados, llegando a la conclusión de que con 3° de separación se podía obtener un *pitch* adecuado. En las figuras 4.1, 4.2 y 4.3 se puede observar el generador de caracteres con distintos ángulos de separación, llegando a la misma conclusión.

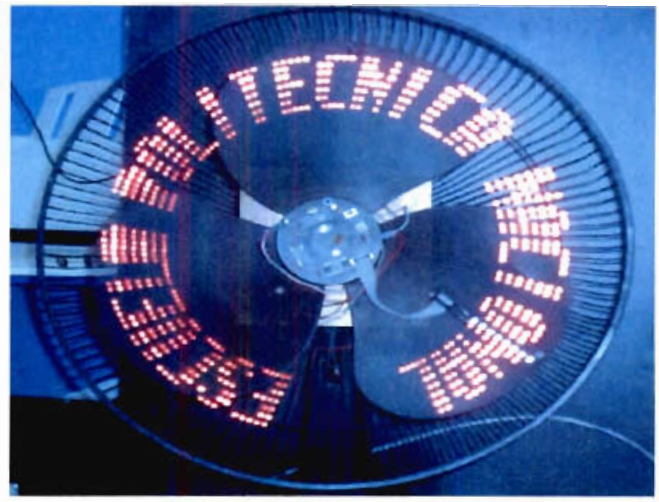


Figura 4.1. Generador de caracteres con 2 grados de separación (30 caracteres).

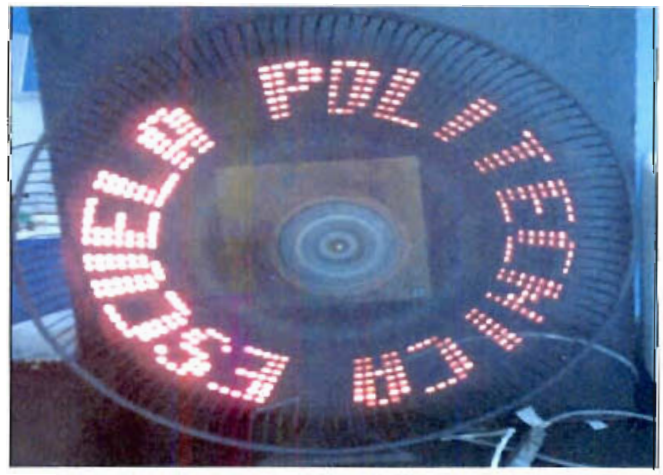


Figura 4.2. Generador de caracteres con 3 grados de separación (20 caracteres).

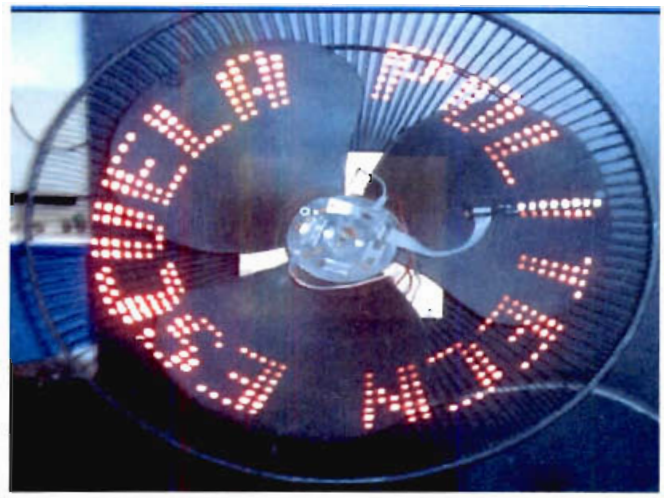


Figura 4.3. Generador de caracteres con 4 grados de separación (15 caracteres).

4.1.4. ALTURA DE CARACTER

La altura del caracter permite tener una idea de la distancia a la que los caracteres podrán ser vistos. En la sección 1.2.7 del capítulo 1 se menciona que por cada pulgada de altura de un caracter, la distancia máxima a la que puede ser visto aumenta 50 pies.

En el generador de caracteres se tiene una altura de caracter de 6 cm, con lo cual, la distancia máxima a la que puede ser visto es:

Con la cámara digital con que se realizaron las fotografías en este capítulo no se pudo obtener una fotografía con la resolución adecuada para mostrar el generador de caracteres a esta distancia, sin embargo en pruebas realizadas en el pasillo del tercer piso del Edificio de Ingeniería Eléctrica de la EPN se comprobó que los mensajes se distinguían a una distancia aproximada de 25m.

4.1.5. ANGULO DE INTENSIDAD MEDIA

Para determinar el ángulo de intensidad media del generador de caracteres se realizaron observaciones desde distintos ángulos medidos a partir del eje de 0°. Se pudo observar que el brillo de los LEDs decrece a medida que el ángulo de vista crece. La figura 4.4 muestra el generador de caracteres mostrando el texto "POLITECNICA NACIONAL" visto desde el eje de 0°

Figura 4.4. Generador de caracteres visto desde el eje de 0°

En las figuras 4.5, 4.6, 4.7 y 4.8, se muestra el generador de caracteres visto desde ángulos de 60°, 90°, 120° y170° respectivamente. Al observar las figuras, se puede decir que el ángulo de intensidad media del generador de caracteres es de aproximadamente 90° y el ángulo de máxima vista es de alrededor de 170°.

Figura 4.5. Angulo de vista de 60° (30° medidos desde el eje de 0°)

Figura 4.6. Angulo de vista de 90° (45° medidos desde el eje de 0°)

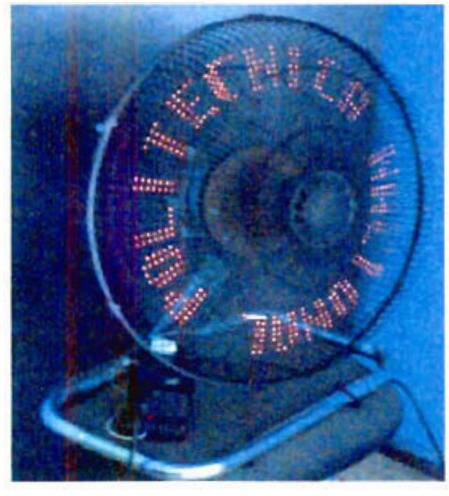
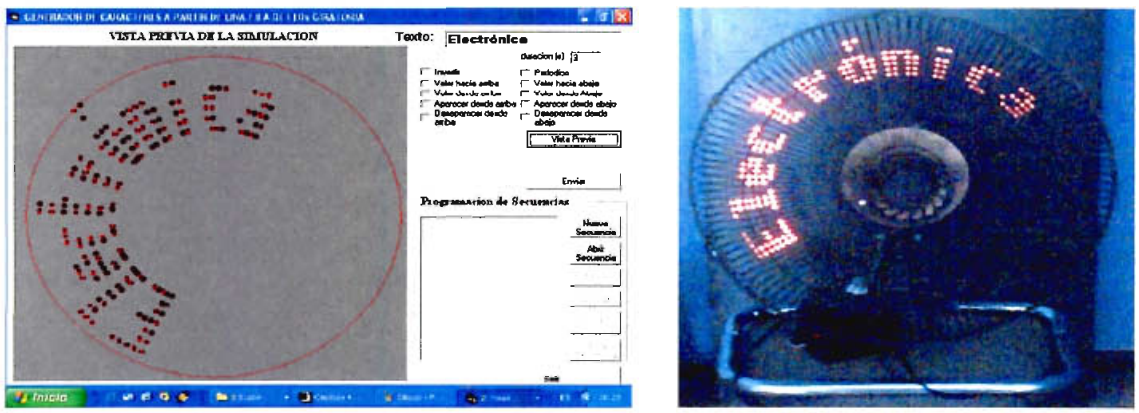


Figura 4.7. Angulo de vista de120°(60° medidos desde el eje de 0°)

Figura 4.8. Angulo de vista de 170°(85° medidos desde el eje de 0°)

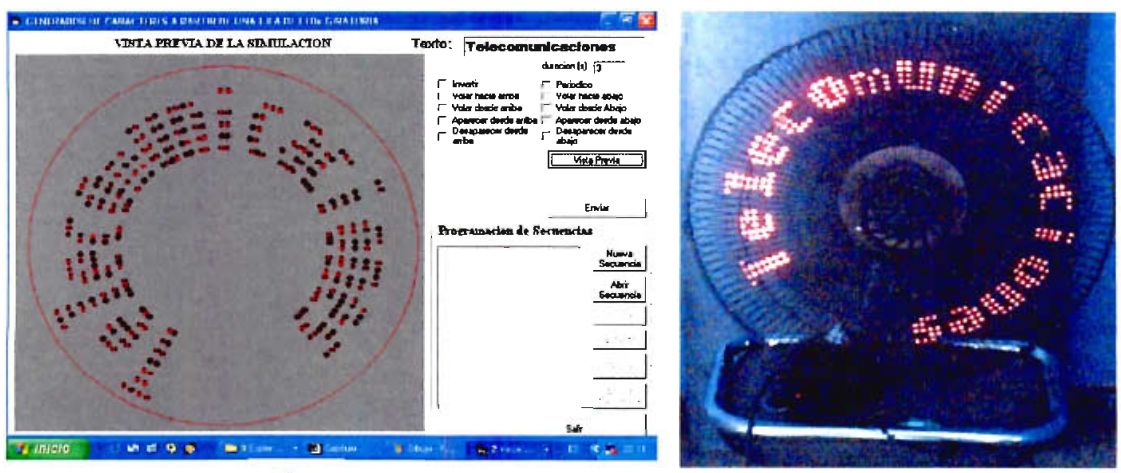
4.1.6. TRANSMISION DE DATOS DESDE EL PC

Al transmitir datos desde el PC a una velocidad de 1200 bps no se presentó ningún problema para la recepción de los datos en el generador de caracteres. Al intentar incrementar la velocidad de transmisión surgieron problemas en la recepción de los datos y aparecían datos errados. Sin embargo, para efectos de transmitir caracteres ASCII, la velocidad de 1200bps es suficiente. En las figuras 4.9, 4.10 y 4.11 se muestran distintos mensajes transmitidos desde el computador al generador de caracteres.



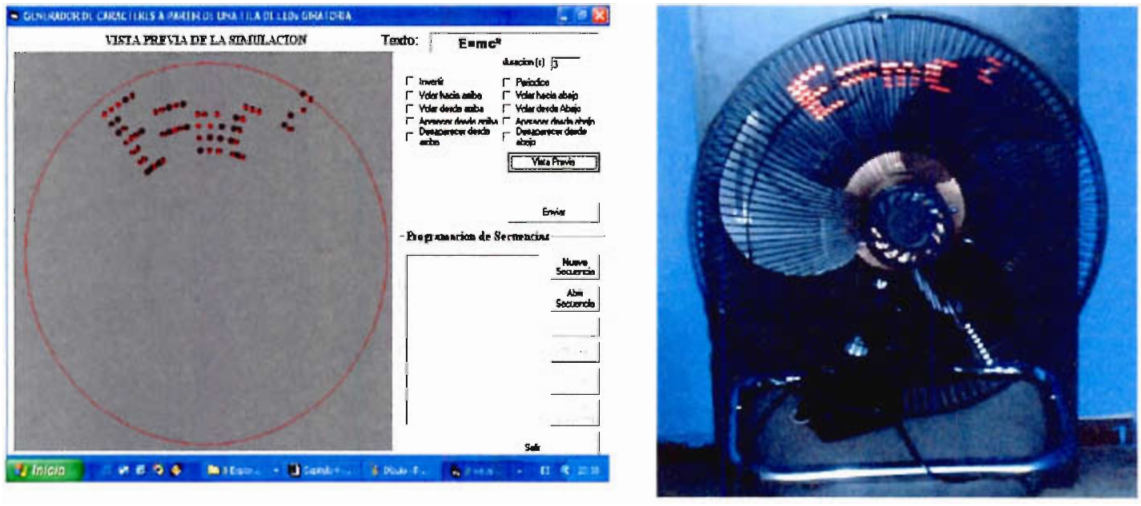
a) b)

Figura 4.9. Transmisión del texto "Electrónica" al generador de caracteres. a) Simulación en el PC, b) Texto transmitido.



a) b)

Figura 4.10. Transmisión del texto "Telecomunicaciones" al generador de caracteres. a) Simulación en el PC, b) Texto transmitido.



a) b)

Figura 4.11. Transmisión del texto "E=mc^2" al generador de caracteres. a) Simulación en el PC, b) Texto transmitido.

En el anexo C se presentan los caracteres ASCII que se pueden mostrar en el generador de caracteres.

4.1.7. ANIMACIONES EN LOS MENSAJES

En el generador de caracteres se implementaron las mismas animaciones descritas en la sección 3.2.1 del capítulo anterior.

Al implementar por software estas animaciones se puede transmitir los mensajes al generador de caracteres y transmitir un código precedido por el carácter 05H para cambiar la animación en el texto.

En las figuras 4.12, 4.13, 4.14 y 4.15 se puede observar al generador de caracteres mostrando en texto "ESCUELA POLITECNICA" con distintos efectos de animación.

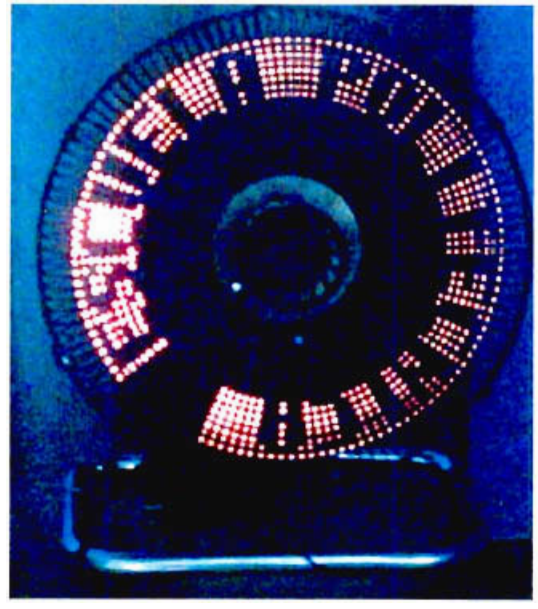


Figura 4.12. Animación invertir

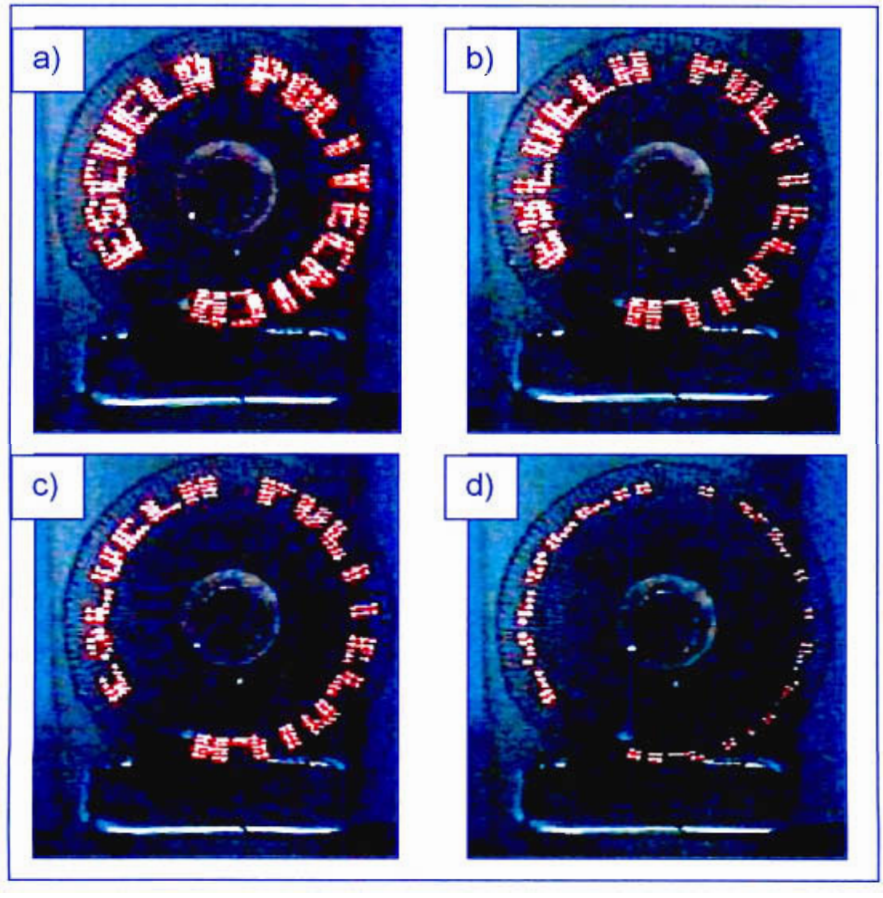


Figura 4.13. Animación Volar hacia Arriba

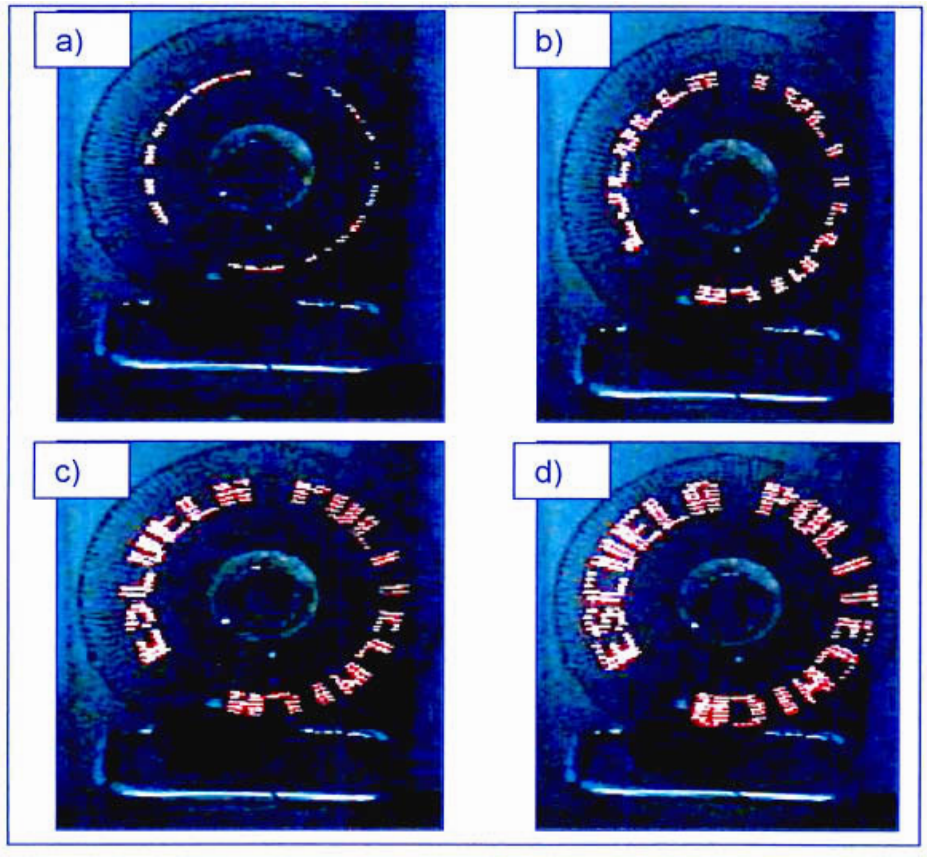


Figura 4.14. Animación aparecer desde abajo

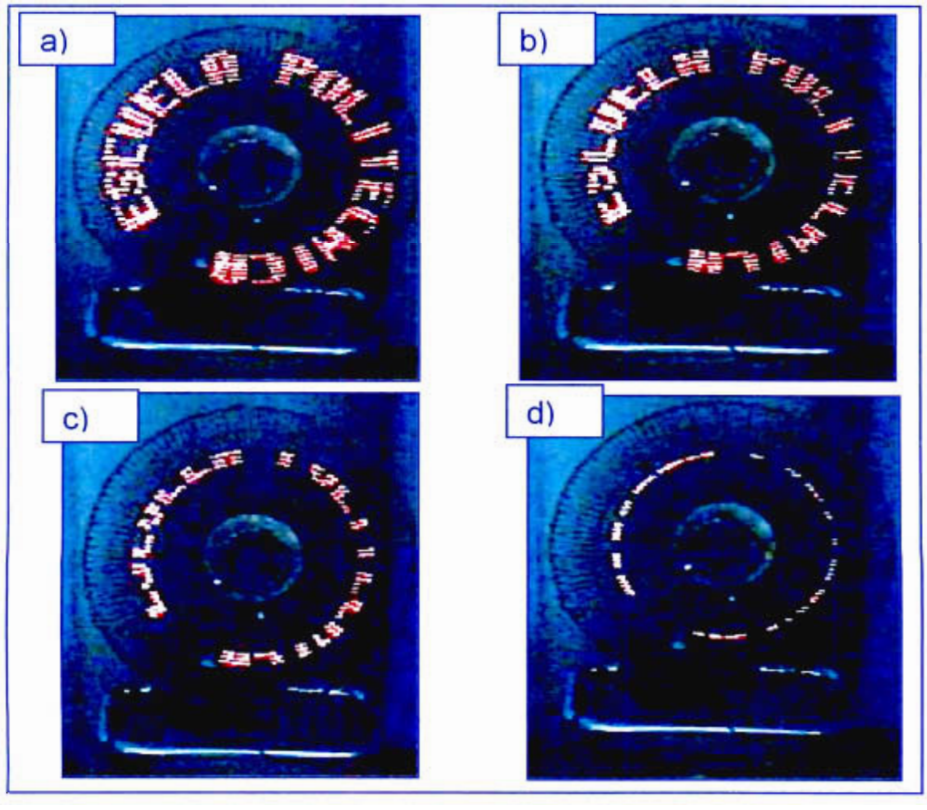


Figura 4.15. Animación desaparecer desde arriba

4.1.8. COLOR DE LOS LEDs

Al utilizar una sola fila de LEDs giratoria se facilita el cambio de color de los caracteres generados y se puede realizar a bajo costo. En las figuras 4.16, 4.17 y 4.18 se observa el generador de caracteres con LEDs de color amarillo, azul y blanco respectivamente.

Figura 4.16. Generador de caracteres con LEDs de color amarillo

Figura 4.17. Generador de caracteres con LEDs de color azul

Figura 4.18. Generador de caracteres con LEDs de color blanco

4.1.9. VENTAJAS Y DESVENTAJAS RESPECTO A LAS MATRICES DE LEDs

Al implementar el generador de caracteres a partir de un fila de LEDs giratoria, se ha logrado obtener una matriz de LEDs con una sola fila de LEDs en movimiento pudiendo mostrar mensajes de texto y animaciones. Sin embargo, es importante analizar que ventajas y desventajas presenta el generador de caracteres respecto a las matrices de LEDs para poder compararlos.

La principal ventaja es que el costo del generador de caracteres se reduce considerablemente respecto a una matriz de LEDs con la misma resolución, debido a que la cantidad de LEDs es mucho menor. Se debe tomar en cuenta que los LEDs influyen en más del 50 % del costo de un tablero electrónico.

Como desventajas se puede mencionar la baja velocidad de transmisión que permite el hardware implementado, ya que esta característica no resulta conveniente si se quiere mostrar gráficos y animaciones.

Otra desventaja que presenta el generador de caracteres respecto a una matriz de LEDs es que su construcción es más complicada. Sin embargo estas

desventajas se compensan con el hecho de haber construido un sistema de generación de caracteres muy llamativo y novedoso.

4.1.10. PRESUPUESTO DE CONSTRUCCION

CANTIDAD	DESCRIPCION	COSTO UNITARIO	COSTO TOTAL
1	Microprocesador AT89C51	$ 6.50	$ 6.50
1	Circuito Integrado A6275	$ 2.50	$ 2.50
1	Circuito Integrado 7414	$ 0.40	$ 0.40
1	Placa de circuito impreso de un lado 7.5 x 7.5 cm (Tarjeta de Control)	$ 5.00	$ 5.00
1	Placa de circuito impreso de un lado 4x4 cm (Tarjeta de comunicación con el PC)	$ 2.00	$ 2.00
4	Placa de circuito impreso de un lado 10x2 cm (Tarjetas de LEDs)	$ 2.00	$ 8.00
1	Baquelita de circuito impreso 30 x 20 cm	$ 3.00	$ 3.00
3	Zócalos	$ 0.40	$ 1.20
1	Fotodiodo (ECG3017)	$ 0.37	$ 0.37
1	Fototransistor (ECG3037)	$ 0.37	$ 0.37
1	Cristal de 11.0592 MHz	$ 0.75	$ 0.75
1	Ventilador industrial	$ 25.00	$ 25.00
1	Adaptador AC/DC 2000 mA	$ 8.00	$ 8.00
3	Carbones y resortes	$ 2.00	$ 6.00
3	Portacarbones	$ 2.00	$ 6.00
5	Capacitores de 10 uF	$ 0.40	$ 2.00
4	Capacitares de 0.1 uF	$ 0.30	$ 1.20
8	LEDs de color rojo	$ 0.12	$ 0.96
8	LEDs de color amarillo	$ 0.25	$ 2.00
8	LEDs de color azul	$ 0.65	$ 5.20
8	LEDs de color blanco	$ 1.80	$ 14.40
	Resistencias	$ 0.18	$ 0.18
	Espadines y Conectores	$ 2.00	$ 2.00
	Suelda y Cable	$ 1.00	$ 1.00
	Tornillos y pernos	$ 2.00	$ 2.00
Costo de construcción del equipo			$106.03

Tabla 4.1 Elementos utilizados en este proyecto y sus precios[1]

[1] La mayoría de los precios fueron tomados del catálogo de JAMECO en internet.

Además, fue necesario adquirir un programador de microcontroladores ATMEL cuyo valor en el mercado fue $ 72,2. Con esto, la suma total del costo de construcción del generador de caracteres a partir de una fila de LEDs giratoria es de $ 178,23 (ciento setenta y ocho dólares con 23 centavos de dólar)

4.2. APLICACIONES

Los dispositivos utilizados para transmitir mensajes de manera visual son muy utilizados debido a que captan fácilmente la atención de las personas. El generador de caracteres a partir de una fila de LEDs giratoria es un dispositivo muy novedoso que despierta un gran interés al verlo. A continuación se enumeran algunas de sus posibles aplicaciones:

- Información de hora y temperatura, con la ayuda de un hardware adicional que consista de un reloj en tiempo real y un sensor de temperatura.

- En bancos y casas cambiarias, para mostrar información de nuevos servicios, tasas de interés o tasas de cambio.

- Para locales comerciales, con información de productos nuevos, ofertas y promociones.

- Implementando mejores técnicas de comunicación se puede actualizar la información en el tablero vía remota utilizando la línea telefónica, por Internet, aplicaciones de red, o mediante un enlace inalámbrico.

- En cines, para dar a conocer estrenos, precios y horarios de proyección de películas.

- Para centros de diversión y entretenimiento.

- Para fines de marketing y publicidad en general.

En la figura 4.19 se ilustran algunas de las posibles aplicaciones mencionadas anteriormente para el generador de caracteres.

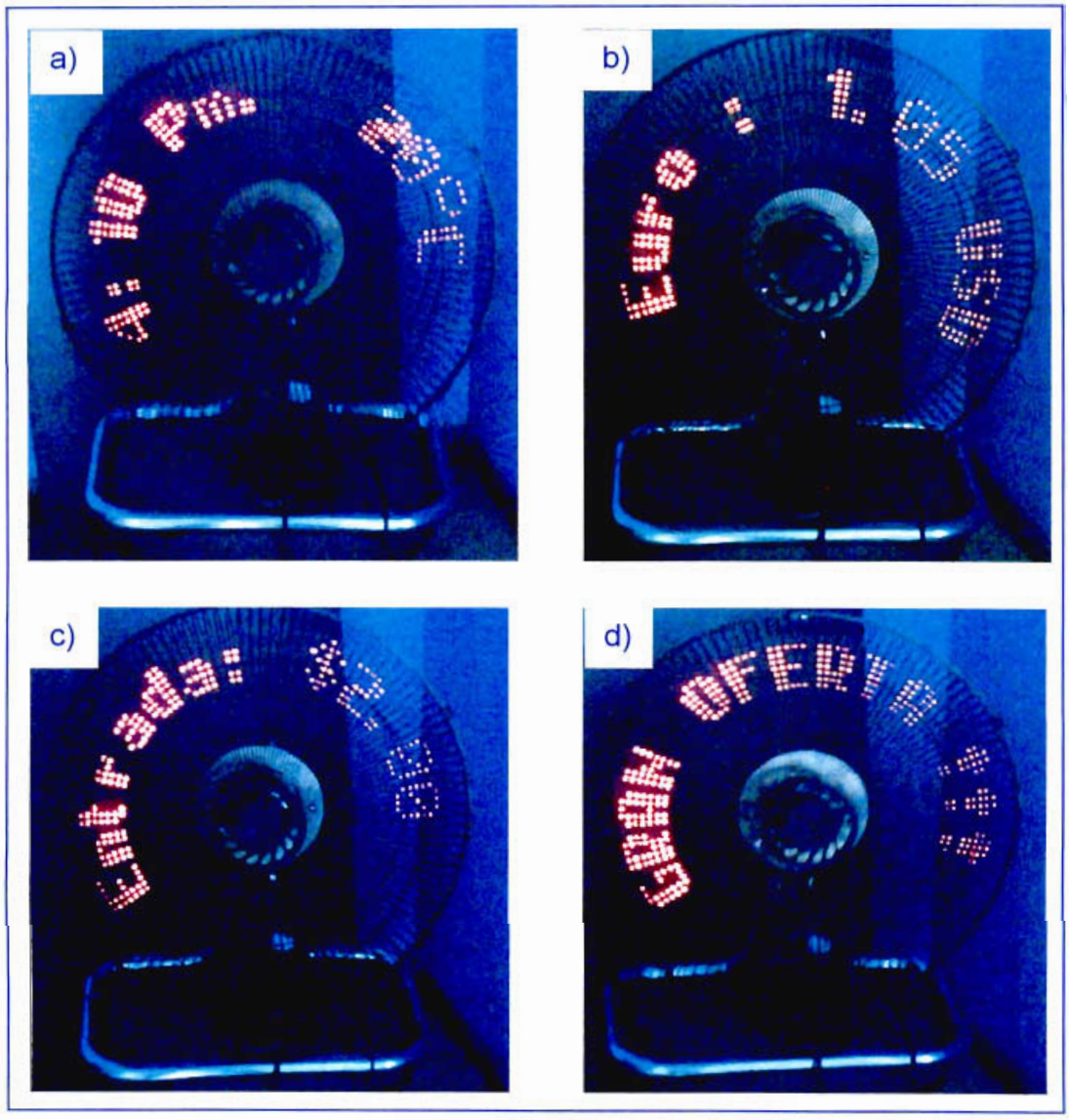


Figura 4.19. Posibles aplicaciones del generador de caracteres a) Información de hora y temperatura, b) Tasa de cambio, c) Valor de la entrada a un centro de entretenimiento, d) Locales comerciales

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- En el presente proyecto de titulación se ha logrado construir un generador de caracteres a partir de una fila de LEDs giratoria utilizando un sistema microprocesado.

- Los diodos emisores de luz (LEDs) son muy utilizados para la construcción de tableros electrónicos debido a que consumen poca energía y casi no necesitan mantenimiento. Estas características han posibilitado que los LEDs también sean utilizados para aplicaciones tales como; la iluminación de interiores y exteriores, semáforos, señalización de carreteras, faros de automóviles, linternas, etc. reemplazando a las lámparas incandescentes.

- El efecto de la persistencia de las imágenes en la retina del ojo humano permite generar caracteres con una sola fila de LEDs giratoria.

- Para que no se produzca el efecto de titilado en los caracteres se necesita que la fila de LEDs gire a una velocidad mínima de aproximadamente 1400 rpm.

- Se debe equilibrar el peso producido por los LEDs en las aspas del ventilador. Este es un factor muy importante puesto que, de no hacerlo, se puede producir una vibración muy fuerte en el motor, la cual reduce la vida útil del mismo.

- El costo del generador de caracteres respecto de las matrices de LEDs se reduce considerablemente debido al reducido número de LEDs, ya que el costo de los LEDs influye en más del 50% del costo de un tablero electrónico. Para el caso del generador de caracteres, el costo de los LEDs de color rojo para construir una matriz de 120 x 8 pixels es de $115.2 (ciento quince dólares con veinte centavos de dólar) que representa el 64% del costo estimado del generador de caracteres. Esta diferencia se hace más notoria para el caso de utilizar LEDs azules y/o blancos cuyo precio unitario bordea las unidades de dólar.

- Puesto que no se encontró ninguna referencia sobre un dispositivo de generación de caracteres similar, el método utilizado para la construcción del generador de caracteres a partir de una fila de LEDs giratoria se basó en el de prueba y error. Para construir el prototipo se eligió un dispositivo de giro y se acopló al mismo la fila de LEDs y la tarjeta de control equilibrando el peso extra de los LEDs y la tarjeta de control, luego se implementó una fuente de alimentación para los LEDs y la tarjeta de control que se encuentran en movimiento utilizando un sistema parecido al de las escobillas de un motor. El mismo sistema fue implementado para establecer la comunicación serial con el generador de caracteres. Finalmente, se estableció un origen desde el cual se comience a generar los caracteres con ayuda de un acoplamiento óptico que consiste de un fotodiodo y un fototransistor.

- La parte más laboriosa en la construcción del generador de caracteres fue la implementación de la fuente de alimentación para los LEDs y la tarjeta de control, así como el acoplamiento de todas las partes al ventilador escogido como dispositivo de giro.

- La dificultad en la construcción del generador de caracteres se compensa con la obtención de un dispositivo muy novedoso para visualización de mensajes.

- Es importante estar siempre al tanto de las novedades tecnológicas que ofrecen los fabricantes de dispositivos electrónicos, puesto que cada vez van apareciendo nuevos elementos que permiten implementar sistemas de forma más fácil y rápida.

- Los dispositivos de control utilizados para manejar los LEDs son el microcontrolador, encargado de cambiar los datos en los LEDs y un CI manejador de LEDs, el cual proporciona una fuente de corriente constante para los LEDs y permite tener el mismo nivel de brillo en los mismos.

- Se puede cambiar fácilmente y a bajo costo el color de los caracteres que se generen.

- Se ha logrado formar una matriz de LEDs a partir de una única fila de LEDs giratoria. Si se actualizan los datos que se muestren a partir de tablas generadas por el usuario en vez de una tabla de datos fija, se pueden mostrar gráficos y animaciones.

- Se puede cambiar los mensajes fácilmente utilizando una comunicación serial RS232 tal como se ha realizado en el software de interfaz con el usuario. También se puede transmitir los mensajes utilizando el computador emulando un terminal no inteligente comunicándose vía RS232 a una velocidad de 1200 bps tomando en cuenta el protocolo de comunicaciones establecido para el generador de caracteres.

- La velocidad de transmisión de 1200 bps resulta muy baja si se quiere mostrar gráficos y animaciones, pero para aplicaciones de texto ASCII resulta suficiente.

- El microprocesador actualiza los datos cada 340 µs y tarda 210µs para actualizarlos utilizando un cristal de 11.0592MHz, por lo tanto, se utiliza el 62% de la capacidad de procesamiento del microcontrolador. El ventilador gira a una velocidad de 1200 rpm y el tiempo de 210µs corresponde a un ángulo de giro de 1.5°, es decir, que se pueden actualizar los datos de hasta 238 columnas.

- La mayoría de las personas que observaron el generador de caracteres se interesaron mucho y quedaron impresionadas por la forma de mostrar los caracteres. Cabe resaltar que a la fecha no se ha encontrado ninguna referencia de algún dispositivo parecido en el mercado.

5.2. RECOMENDACIONES

- Por seguridad, el generador de caracteres debe contar con una cubierta como las rejillas del ventilador para evitar accidentes especialmente al construir el generador de caracteres.

- Para implementar la fuente de alimentación se podría aprovechar la energía mecánica generada cuando gira el motor para implementar un sistema parecido a un generador de voltaje DC, es decir, acoplar un campo magnético para inducir una corriente en los LEDs y la tarjeta de control del generador de caracteres.

- Se puede aumentar el número de LEDs para generar matrices con mayor número de filas y, por lo tanto, con mayor resolución. El número de LEDs que se puede colocar es limitado debido a que a medida que la distancia del LED al centro de giro crece, también crece la distancia de separación entre dos pixels contiguos (*pitch*) dando como resultado una baja calidad de la imagen que se muestre.

- Si se ubican tres placas de LEDs con los colores básicos (Rojo, Verde y Azul) o se utilizan LEDs de dos o tres colores, se puede combinar las tonalidades para generar matrices de LEDs a color.

- Para transmitir los mensajes al generador de caracteres podría implementarse un hardware adicional que consista de un sistema microprocesado que almacene un conjunto de mensajes en memoria y los transmita serialmente al generador de caracteres en un tiempo determinado. De esta manera, se puede prescindir fácilmente del uso de un computador para actualizar los mensajes.

- Se puede añadir también un reloj en tiempo real y un sensor de temperatura para poder presentar la información de hora y temperatura en el generador de caracteres.

- Mediante la implementación de tecnologías como la transmisión de datos vía telefónica o la comunicación en red se puede construir un generador de caracteres que pueda actualizarse vía remota.

- Para incrementar la distancia máxima en que se pueden ver los mensajes se puede incrementar la separación de los LEDs que forman parte de la fila de LEDs giratoria.

- Durante el desarrollo de esta investigación, se encontró otro tipo de microcontrolador del fabricante ATMEL denominado AVR, el cual permite ejecutar instrucciones a velocidades mucho mayores que el microcontrolador AT89C51 (12 MIPS comparado con 1 MIPS del AT89C2051 utilizando un cristal de 12MHz). Utilizando este tipo de microcontroladores se puede desarrollar un algoritmo que actualice los datos en los LEDs según la posición que ocupe cada uno de ellos en vez de actualizarlos tomando en cuenta la posición que ocupa toda la fila de LEDs.

BIBLIOGRAFIA

- SMITH, Steven, "The Scientist and Engineer's Guide to Digital Signal Processing", USA, 1999

- FERRER, Ricardo, "Tratado Práctico de Luminotecnia", Primera Edición, Barcelona, 1934

- STEWART, David, National Semiconductor, Application Note 371.

- ATMEL, "ATMEL Corporation 851 Flash Microcontroller Data Book", San Jose CA, 1997

ANEXO A

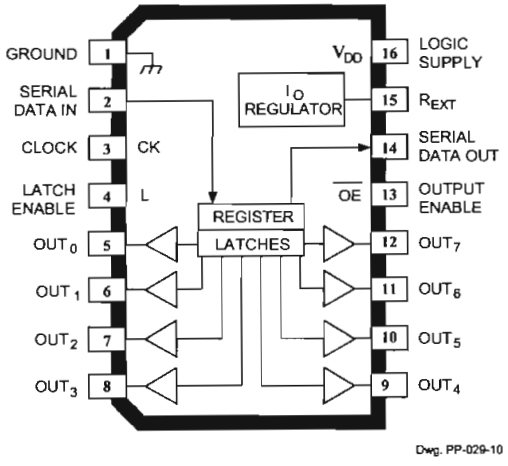
HOJAS DE ESPECIFICACIONES

6275

Data Sheet
26185.200B†

8-BIT SERIAL-INPUT, CONSTANT-CURRENT LATCHED LED DRIVER

A6275ELW



Note that the A6275EA (DIP) and the A6275ELW (SOIC) are electrically identical and share a common terminal number assignment.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V_{DD} ..... **7.0 V**
Output Voltage Range,
V_O ..... **-0.5 V to +17 V**
Output Current, I_O ..... **90 mA**
Ground Current, I_{GND} ..... **750 mA**
Input Voltage Range,
V_I ..... **-0.4 V to V_{DD} + 0.4 V**
Package Power Dissipation,
P_D ..... **See Graph**
Operating Temperature Range,
T_A ..... **-40°C to +85°C**
Storage Temperature Range,
T_S ..... **-55°C to +150°C**

Caution: These CMOS devices have input static protection (Class 2) but are still susceptible to damage if exposed to extremely high static electrical charges.

The A6275EA and A6275ELW are specifically designed for LED-display applications. Each BiCMOS device includes an 8-bit CMOS shift register, accompanying data latches, and eight npn constant-current sink drivers. Except for package style and allowable package power dissipation, the two devices are identical.

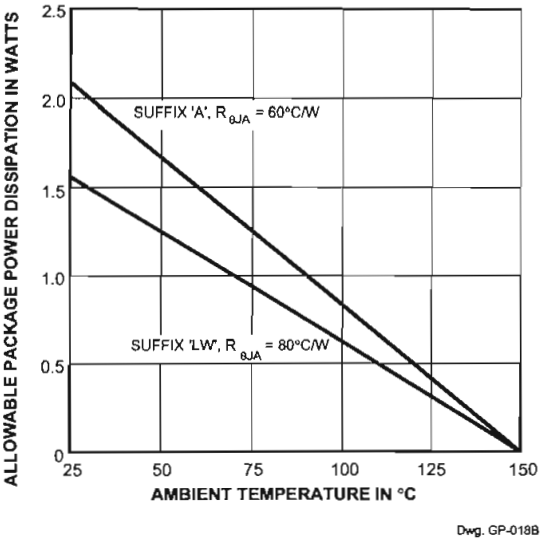
The CMOS shift register and latches allow direct interfacing with microprocessor-based systems. With a 5 V logic supply, typical serial data-input rates are up to 20 MHz. The LED drive current is determined by the user's selection of a single resistor. A CMOS serial data output permits cascade connections in applications requiring additional drive lines. For inter-digit blanking, all output drivers can be disabled with an ENABLE input high. Similar 150 mA output devices are available as the A6277EA and A6277ELW; similar 16-bit devices are available as the A6276EA and A6276ELW.

Two package styles are provided for through-hole DIP (suffix A) or surface-mount SOIC (suffix LW). Under normal applications, copper lead frames and low logic-power dissipation allow these devices to sink maximum rated current through all outputs continuously over the operating temperature range (90 mA, 0.9 V drop, +85°C). Both devices are also available for operation over the standard temperature range of -20°C to +85°C. To order, change the suffix letter 'E' to 'S'.

FEATURES

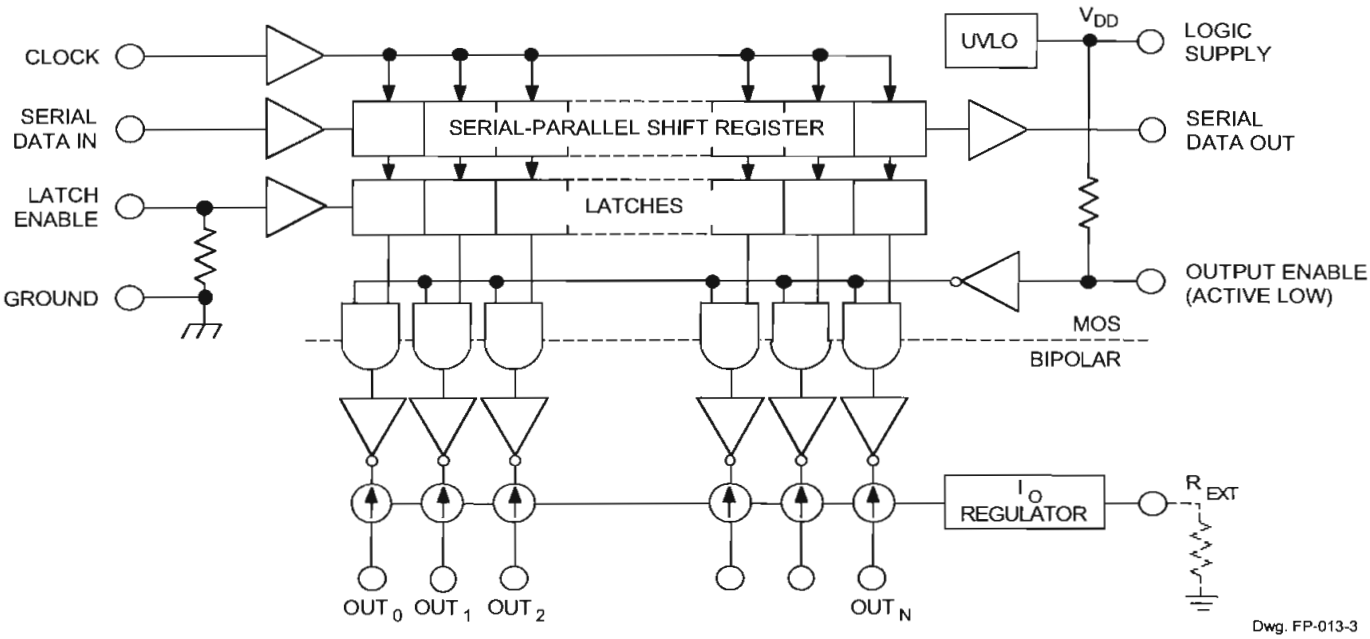
- To 90 mA Constant-Current Outputs
- Under-Voltage Lockout
- Low-Power CMOS Logic and Latches
- High Data Input Rate
- Pin-Compatible with TB62705CP

Always order by complete part number, e.g., **A6275EA**.



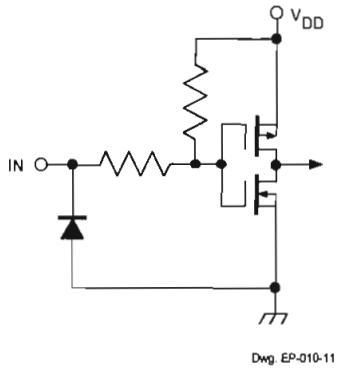
Dwg. GP-018B

FUNCTIONAL BLOCK DIAGRAM



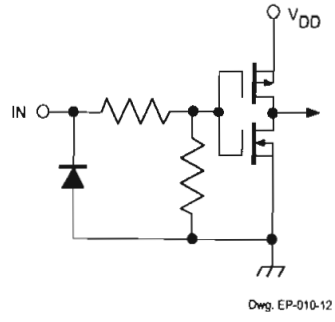
Dwg. FP-013-3

115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000



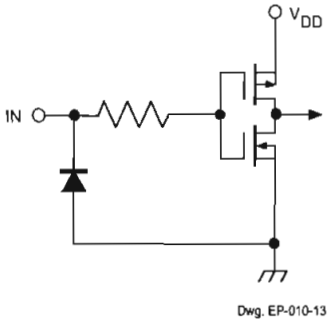
Dwg. EP-010-11

OUTPUT ENABLE (active low)



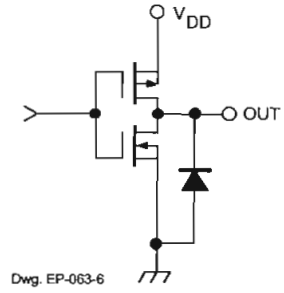
Dwg. EP-010-12

LATCH ENABLE



Dwg. EP-010-13

CLOCK and SERIAL DATA IN



Dwg. EP-063-6

SERIAL DATA OUT

TRUTH TABLE

Serial Data Input	Clock Input	Shift Register Contents I_1 I_2 I_3 ... I_{N-1} I_N	Serial Data Output	Latch Enable Input	Latch Contents I_1 I_2 I_3 ... I_{N-1} I_N	Output Enable Input	Output Contents I_1 I_2 I_3 ... I_{N-1} I_N
H	↑	H R_1 R_2 ... R_{N-2} R_{N-1}	R_{N-1}				
L	↑	L R_1 R_2 ... R_{N-2} R_{N-1}	R_{N-1}				
X	↓	R_1 R_2 R_3 ... R_{N-1} R_N	R_N				
		X X X ... X X	X	L	R_1 R_2 R_3 ... R_{N-1} R_N		
		P_1 P_2 P_3 ... P_{N-1} P_N	P_N	H	P_1 P_2 P_3 ... P_{N-1} P_N	L	P_1 P_2 P_3 ... P_{N-1} P_N
					X X X ... X X	H	H H H ... H H

L = Low Logic (Voltage) Level H = High Logic (Voltage) Level X = Irrelevant P = Present State R = Previous State

6275
8-BIT SERIAL-INPUT, CONSTANT-CURRENT LATCHED LED DRIVER

ELECTRICAL CHARACTERISTICS at T_A = +25°C, V_{DD} = 5 V (unless otherwise noted).

Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Unit
Supply Voltage Range	V_{DD}	Operating	4.5	5.0	5.5	V
Under-Voltage Lockout	$V_{DD(UV)}$	V_{DD} = 0 → 5 V	3.4	–	4.0	V
Output Current (any single output)	I_O	V_{CE} = 0.7 V, R_{EXT} = 250 Ω	64.2	75.5	86.8	mA
		V_{CE} = 0.7 V, R_{EXT} = 470 Ω	34.1	40.0	45.9	mA
Output Current Matching (difference between any two outputs at same V_{CE})	ΔI_O	0.4 V ≤ $V_{CE(A)}$ = $V_{CE(B)}$ ≤ 0.7 V:				
		R_{EXT} = 250 Ω	–	±1.5	±6.0	%
		R_{EXT} = 470 Ω	–	±1.5	±6.0	%
Output Leakage Current	I_{CEX}	V_{OH} = 15 V	–	1.0	5.0	µA
Logic Input Voltage	V_{IH}		0.7V_{DD}	–	V_{DD}	V
	V_{IL}		GND	–	0.3V_{DD}	V
SERIAL DATA OUT Voltage	V_{OL}	I_{OL} = 500 µA	–	–	0.4	V
	V_{OH}	I_{OH} = -500 µA	4.6	–	–	V
Input Resistance	R_I	ENABLE Input, Pull Up	150	300	600	kΩ
		LATCH Input, Pull Down	100	200	400	kΩ
Supply Current	$I_{DD(OFF)}$	R_{EXT} = open, V_{OE} = 5 V	–	0.8	1.4	mA
		R_{EXT} = 470 Ω, V_{OE} = 5 V	3.5	6.0	8.0	mA
		R_{EXT} = 250 Ω, V_{OE} = 5 V	6.5	11	15	mA
	$I_{DD(ON)}$	R_{EXT} = 470 Ω, V_{OE} = 0 V	5.0	10	14	mA
		R_{EXT} = 250 Ω, V_{OE} = 0 V	8.0	16	24	mA

Typical Data is at V_{DD} = 5 V and is for design information only.

SWITCHING CHARACTERISTICS at T_A = 25°C, V_{DD} = V_{IH} = 5 V, V_{CE} = 0.4 V, V_{IL} = 0 V, R_{EXT} = 470 Ω, I_O = 40 mA, V_L = 3 V, R_L = 65 Ω, C_L = 10.5 pF.

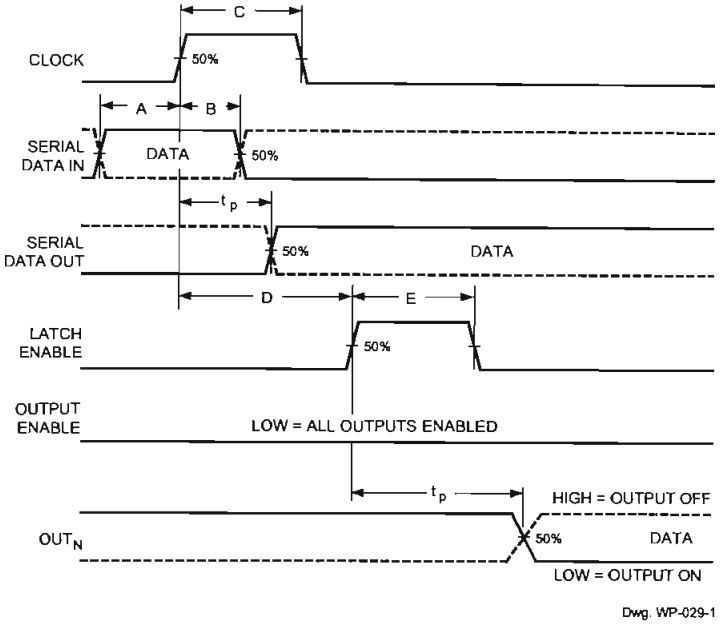
Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Unit
Propagation Delay Time	t_{pHL}	CLOCK-OUT_n	–	350	1000	ns
		LATCH-OUT_n	–	350	1000	ns
		ENABLE-OUT_n	–	350	1000	ns
		CLOCK-SERIAL DATA OUT	–	40	–	ns
Propagation Delay Time	t_{pLH}	CLOCK-OUT_n	–	300	1000	ns
		LATCH-OUT_n	–	300	1000	ns
		ENABLE-OUT_n	–	300	1000	ns
		CLOCK-SERIAL DATA OUT	–	40	–	ns
Output Fall Time	t_f	90% to 10% voltage	150	350	1000	ns
Output Rise Time	t_r	10% to 90% voltage	150	300	600	ns

RECOMMENDED OPERATING CONDITIONS

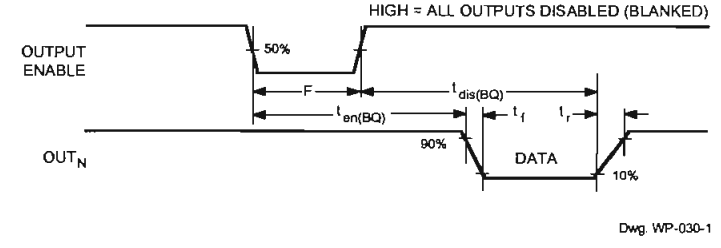
Characteristic	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DD}		4.5	5.0	5.5	V
Output Voltage	V_O		–	1.0	4.0	V
Output Current	I_O	Continuous, any one output	–	–	90	mA
	I_{OH}	SERIAL DATA OUT	–	–	-1.0	mA
	I_{OL}	SERIAL DATA OUT	–	–	1.0	mA
Logic Input Voltage	V_{IH}		$0.7V_{DD}$	–	V_{DD} + 0.3	V
	V_{IL}		-0.3	–	$0.3V_{DD}$	V
Clock Frequency	f_{CK}	Cascade operation	–	–	10	MHz

TIMING REQUIREMENTS and SPECIFICATIONS

(Logic Levels are V_{DD} and Ground)



Dwg. WP-029-1



Dwg. WP-030-1

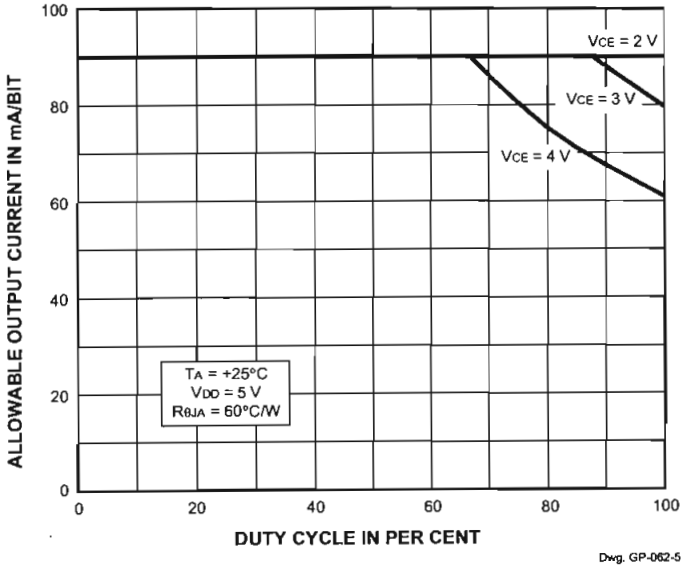
A. Data Active Time Before Clock Pulse (Data Set-Up Time), $t_{su(D)}$ ..... **60 ns**
B. Data Active Time After Clock Pulse (Data Hold Time), $t_{h(D)}$ ..... **20 ns**
C. Clock Pulse Width, $t_{w(CK)}$ ..... **50 ns**
D. Time Between Clock Activation and Latch Enable, $t_{su(L)}$ ..... **100 ns**
E. Latch Enable Pulse Width, $t_{w(L)}$ ..... **100 ns**
F. Output Enable Pulse Width, $t_{w(OE)}$ ..... **4.5 µs**
NOTE – Timing is representative of a 10 MHz clock. Significantly higher speeds are attainable.
– Max. Clock Transition Time, t_r or t_f ..... **10 µs**

Information present at any register is transferred to the respective latch when the LATCH ENABLE is high (serial-to-parallel conversion). The latches will continue to accept new data as long as the LATCH ENABLE is held high. Applications where the latches are bypassed (LATCH ENABLE tied high) will require that the OUTPUT ENABLE input be high during serial data entry.

When the OUTPUT ENABLE input is high, the output source drivers are disabled (OFF). The information stored in the latches is not affected by the OUTPUT ENABLE input. With the OUTPUT ENABLE input low, the outputs are controlled by the state of their respective latches.

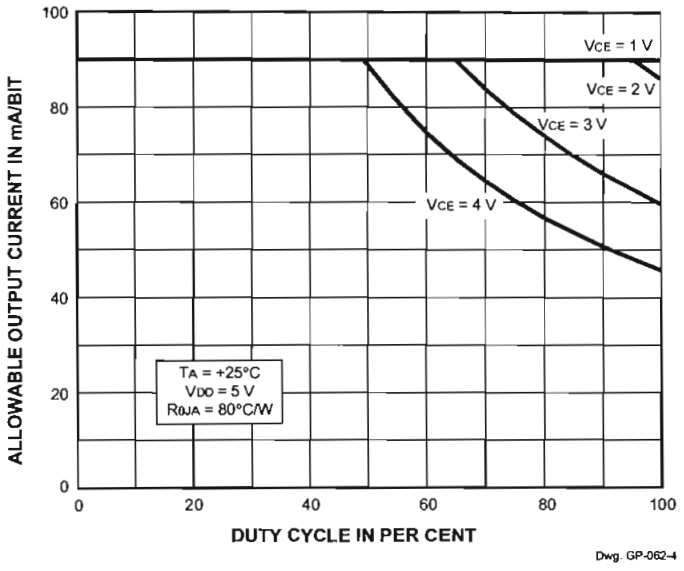
ALLOWABLE OUTPUT CURRENT AS A FUNCTION OF DUTY CYCLE

A6275EA

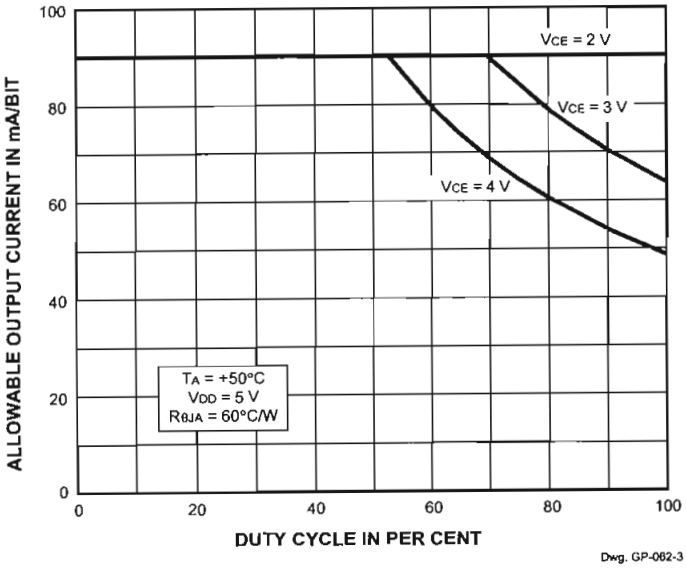


Dwg. GP-062-5

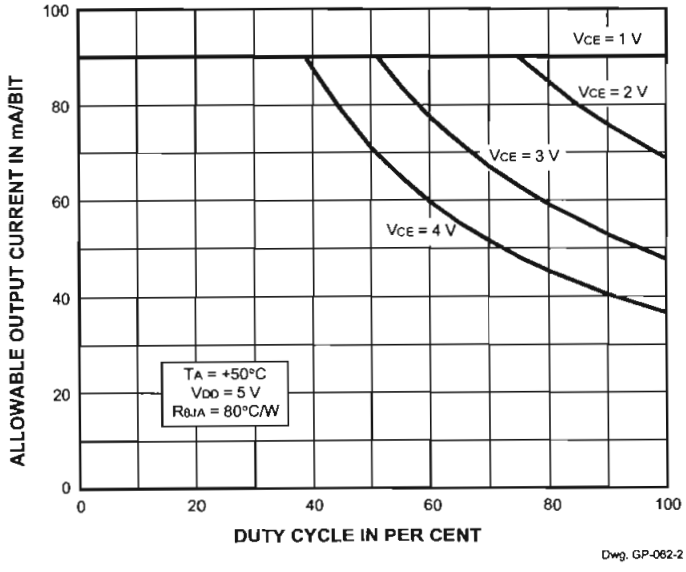
A6275ELW



Dwg. GP-062-4



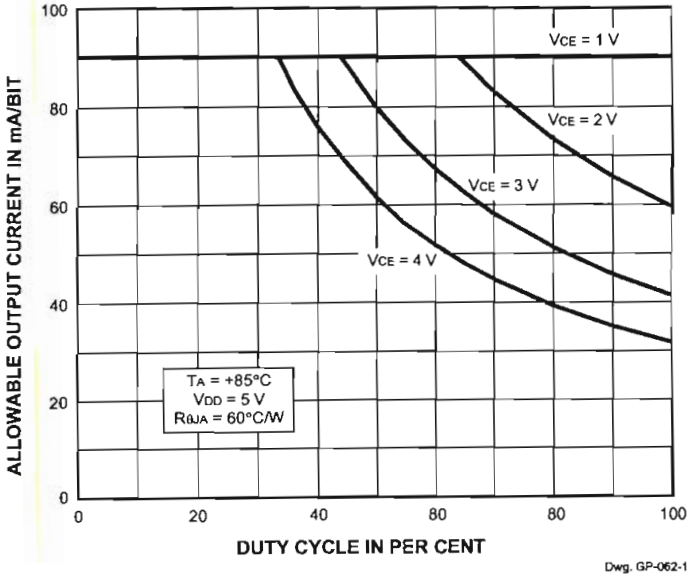
Dwg. GP-062-3



Dwg. GP-062-2

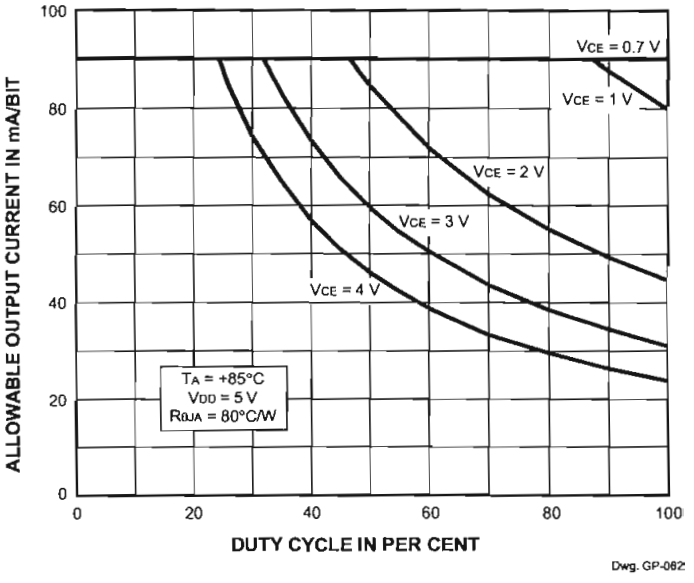
ALLOWABLE OUTPUT CURRENT AS A FUNCTION OF DUTY CYCLE (cont.)

A6275EA



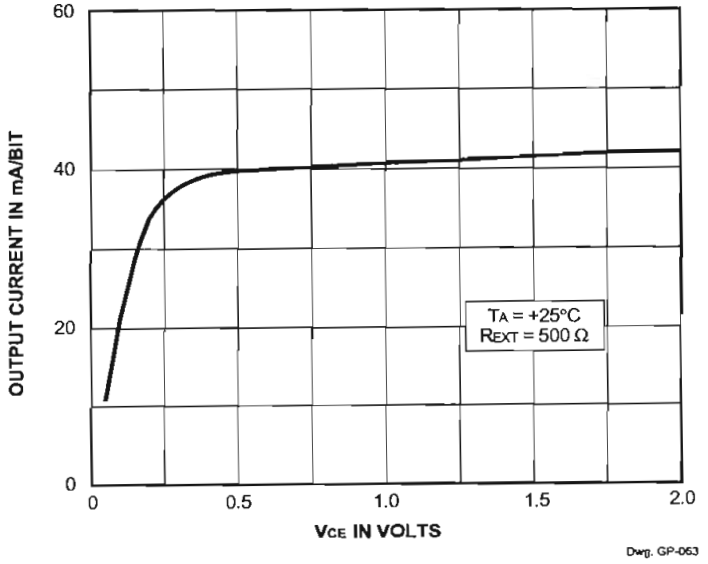
Dwg. GP-062-1

A6275ELW



Dwg. GP-062

TYPICAL CHARACTERISTICS



Dwg. GP-063

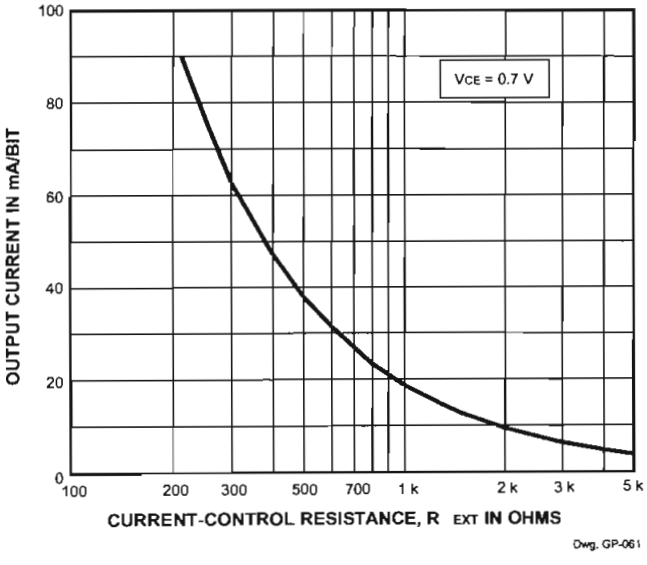
Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

TERMINAL DESCRIPTION

Terminal No.	Terminal Name	Function
1	GND	Reference terminal for control logic.
2	SERIAL DATA IN	Serial-data input to the shift-register.
3	CLOCK	Clock input terminal for data shift on rising edge.
4	LATCH ENABLE	Data strobe input terminal; serial data is latched with high-level input.
5-12	OUT_{0-7}	The eight current-sinking output terminals.
13	OUTPUT ENABLE	When (active) low, the output drivers are enabled; when high, all output drivers are turned OFF (blanked).
14	SERIAL DATA OUT	CMOS serial-data output to the following shift-register.
15	R_{EXT}	An external resistor at this terminal establishes the output current for all sink drivers.
16	SUPPLY	(V_{DD}) The logic supply voltage (typically 5 V).

Applications Information

The load current per bit (I_O) is set by the external resistor (R_{EXT}) as shown in the figure below.



Package Power Dissipation (P_D). The maximum allowable package power dissipation is determined as

$$P_D(max) = (150 - T_A)/R_{\theta JA}.$$

The actual package power dissipation is

$$P_D(act) = dc(V_{CE} \bullet I_O \bullet 8) + (V_{DD} \bullet I_{DD}).$$

When the load supply voltage is greater than 3 V to 5 V, considering the package power dissipating limits of these devices, or if $P_D(act) > P_D(max)$, an external voltage reducer (V_{DROP}) should be used.

Load Supply Voltage (V_{LED}). These devices are designed to operate with driver voltage drops (V_{CE}) of 0.4 V to 0.7 V with LED forward voltages (V_F) of 1.2 V to 4.0 V. If higher voltages are dropped across the driver, package power dissipation will be increased significantly. To minimize package power dissipation, it is recommended to use the lowest possible load supply voltage or to set any series dropping voltage (V_{DROP}) as

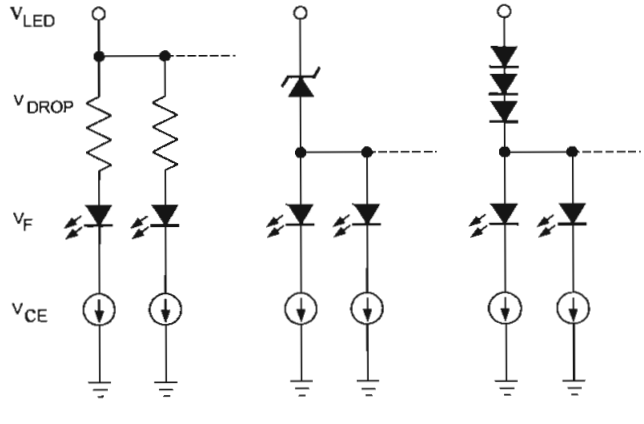
$$V_{DROP} = V_{LED} - V_F - V_{CE}$$

with $V_{DROP} = I_o \bullet R_{DROP}$ for a single driver, or a Zener diode (V_Z), or a series string of diodes (approximately 0.7 V per diode) for a group of drivers. If the available voltage source will cause unacceptable dissipation and series resistors or diode(s) are undesirable, a regulator such as the Sanken Series SAI or Series SI can be used to provide supply voltages as low as 3.3 V.

For reference, typical LED forward voltages are:

Blue	3.0 – 4.0 V
Green	1.8 – 2.2 V
Yellow	2.0 – 2.1 V
Amber	1.9 – 2.65 V
Red	1.6 – 2.25 V
Infrared	1.2 – 1.5 V

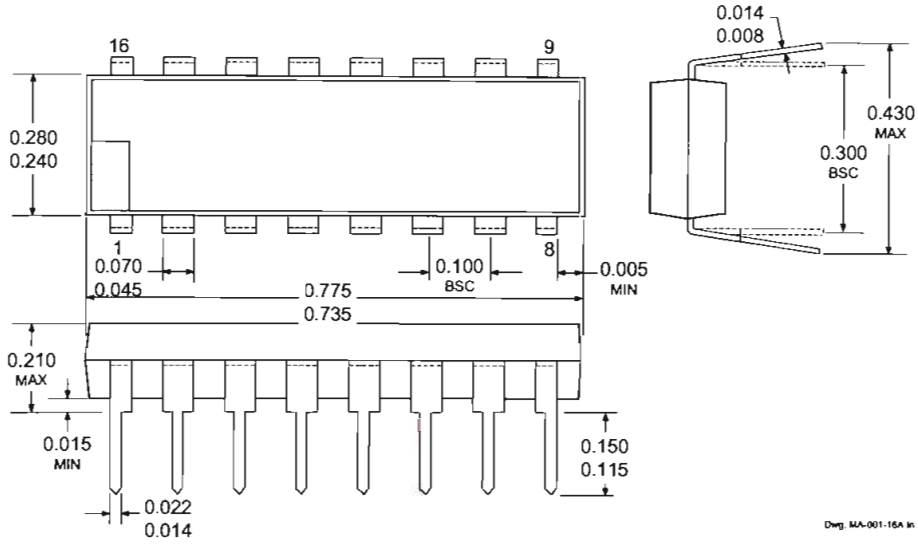
Pattern Layout. This device has a common logic-ground and power-ground terminal. If ground pattern layout contains large common-mode resistance, and the voltage between the system ground and the LATCH ENABLE or CLOCK terminals exceeds 2.5 V (because of switching noise), these devices may not operate correctly.



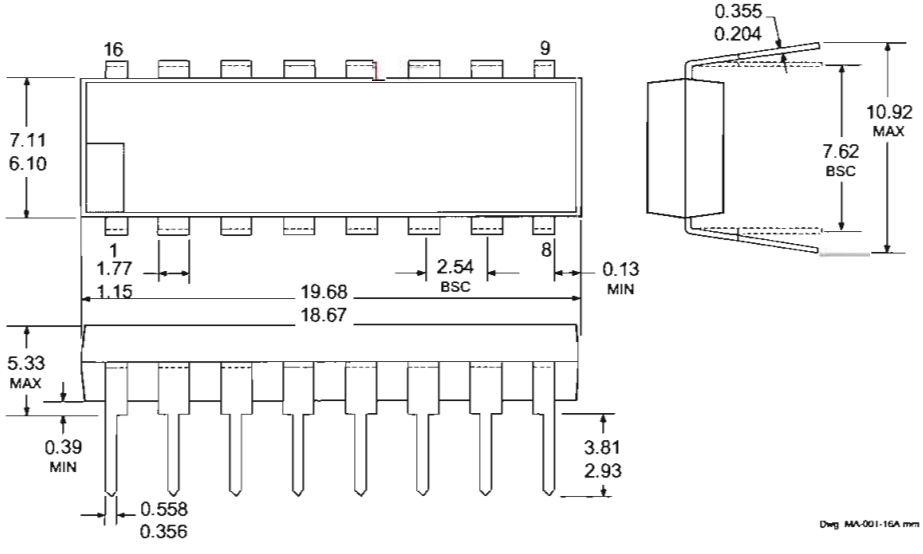
Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

A6275EA

Dimensions in Inches
(controlling dimensions)



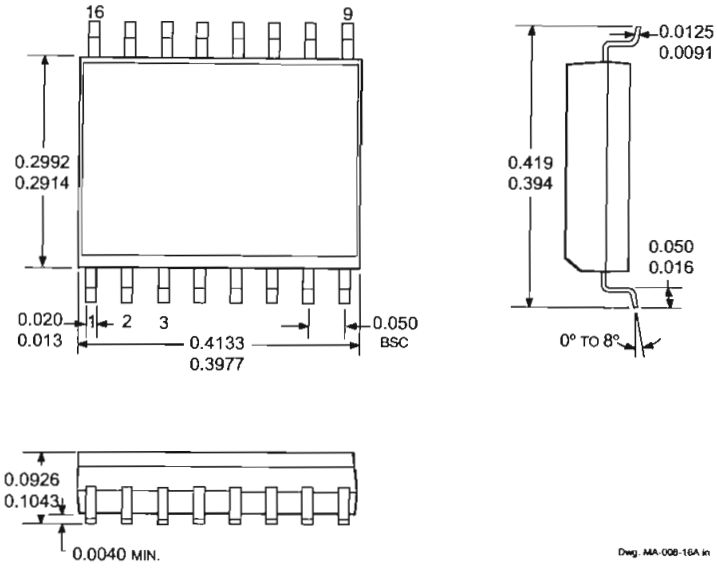
Dimensions in Millimeters
(for reference only)



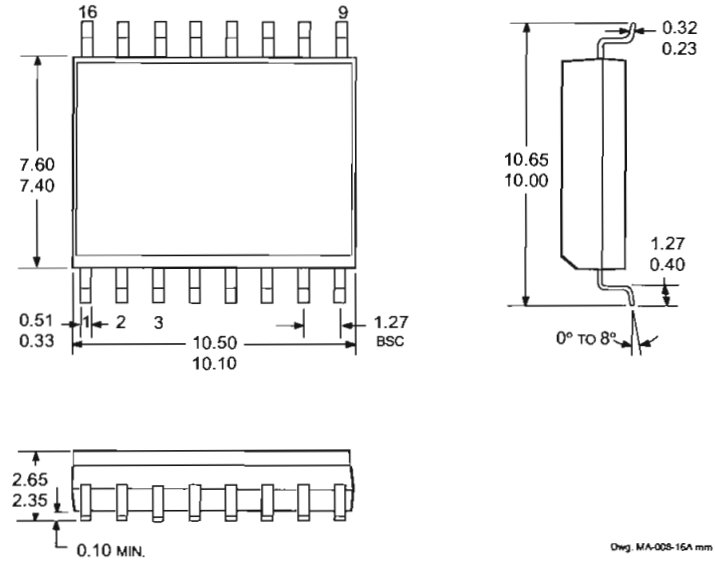
NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
2. Lead spacing tolerance is non-cumulative
3. Lead thickness is measured at seating plane or below.
4. Supplied in standard sticks/tubes of 25 devices.

A6275ELW

Dimensions in Inches
(for reference only)



Dimensions in Millimeters
(controlling dimensions)



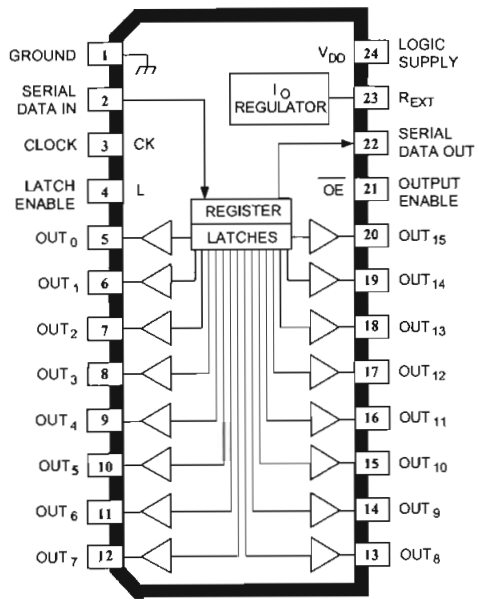
NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
2. Lead spacing tolerance is non-cumulative.
3. Supplied in standard sticks/tubes of 47 devices or add "TR" to part number for tape and reel.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

6276

16-BIT SERIAL-INPUT, CONSTANT-CURRENT LATCHED LED DRIVER

A6276ELW



Dwg. PP-029-11

Note that the A6276EA (DIP) and the A6276ELW (SOIC) are electrically identical and share a common terminal number assignment.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V_{DD} ..... **7.0 V**
Output Voltage Range,
V_O ..... **-0.5 V to +17 V**
Output Current, I_O ..... **90 mA**
Ground Current, I_{GND} ..... **1475 mA**
Input Voltage Range,
V_I ..... **-0.4 V to V_{DD} + 0.4 V**
Package Power Dissipation,
P_D ..... **See Graph**
Operating Temperature Range,
T_A ..... **-40°C to +85°C**
Storage Temperature Range,
T_S ..... **-55°C to +150°C**

Caution: These CMOS devices have input static protection (Class 2) but are still susceptible to damage if exposed to extremely high static electrical charges.

The A6276EA and A6276ELW are specifically designed for LED-display applications. Each BiCMOS device includes a 16-bit CMOS shift register, accompanying data latches, and 16 npn constant-current sink drivers. Except for package style and allowable package power dissipation, the two devices are identical.

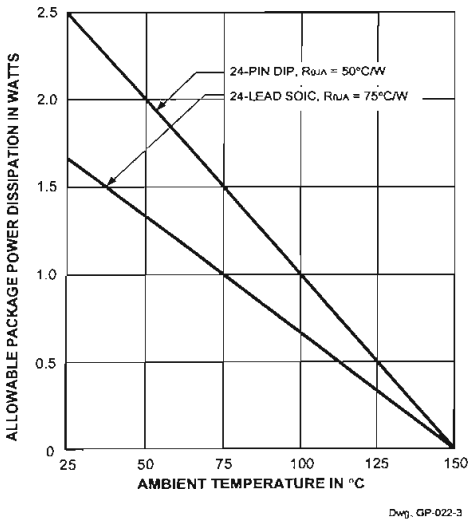
The CMOS shift register and latches allow direct interfacing with microprocessor-based systems. With a 5 V logic supply, typical serial data-input rates are up to 20 MHz. The LED drive current is determined by the user's selection of a single resistor. A CMOS serial data output permits cascade connections in applications requiring additional drive lines. For inter-digit blanking, all output drivers can be disabled with an ENABLE input high. Similar 8-bit devices are available as the A6275EA and A6275ELW.

Two package styles are provided for through-hole DIP (suffix A) or surface-mount SOIC (suffix LW). Under normal applications, a copper lead frame and low logic-power dissipation allow the dual in-line package to sink maximum rated current through all outputs continuously over the operating temperature range (90 mA, 0.75 V drop, +85°C). Both devices are also available for operation over the standard temperature range of -20°C to +85°C. To order, change the suffix letter 'E' to 'S'.

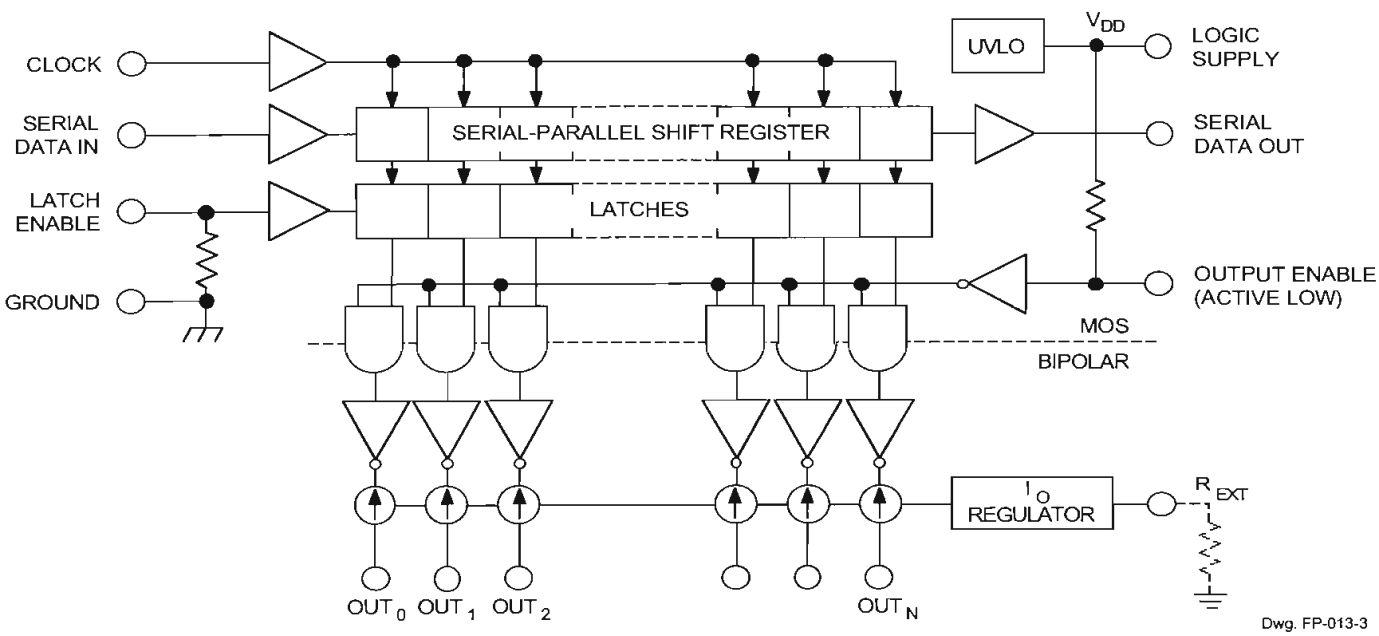
FEATURES

- To 90 mA Constant-Current Outputs
- Under-Voltage Lockout
- Low-Power CMOS Logic and Latches
- High Data Input Rate
- Functional Replacement for TB62706BN/BF

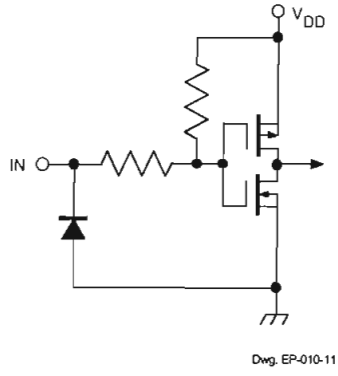
Always order by complete part number, e.g., **A6276EA**.



FUNCTIONAL BLOCK DIAGRAM

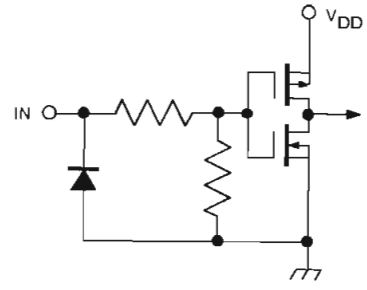


Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000



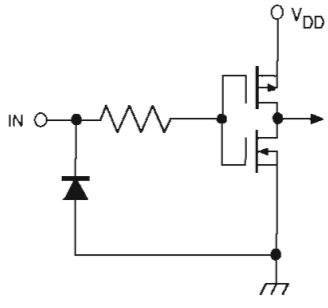
Dwg. EP-010-11

OUTPUT ENABLE (active low)



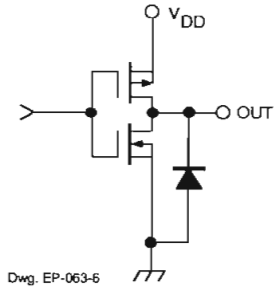
Dwg. EP-010-12

LATCH ENABLE



Dwg. EP-010-13

CLOCK and SERIAL DATA IN



Dwg. EP-063-6

SERIAL DATA OUT

TRUTH TABLE

Serial Data Input	Clock Input	Shift Register Contents	Serial Data Output	Latch Enable Input	Latch Contents	Output Enable Input	Output Contents
H	↑	H R_1 R_2 ... R_{N-2} R_{N-1}	R_{N-1}				
L	↑	L R_1 R_2 ... R_{N-2} R_{N-1}	R_{N-1}				
X	↓	R_1 R_2 R_3 ... R_{N-1} R_N	R_N				
		X X X ... X X	X	L	R_1 R_2 R_3 ... R_{N-1} R_N		
		P_1 P_2 P_3 ... P_{N-1} P_N	P_N	H	P_1 P_2 P_3 ... P_{N-1} P_N	L	P_1 P_2 P_3 ... P_{N-1} P_N
					X X X ... X X	H	H H H ... H H

L = Low Logic (Voltage) Level H = High Logic (Voltage) Level X = Irrelevant P = Present State R = Previous State

ELECTRICAL CHARACTERISTICS at T_A = +25°C, V_{DD} = 5 V (unless otherwise noted).

Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Unit
Supply Voltage Range	V_{DD}	Operating	4.5	5.0	5.5	V
Under-Voltage Lockout	$V_{DD(UV)}$	V_{DD} = 0 → 5 V	3.4	–	4.0	V
Output Current (any single output)	I_O	V_{CE} = 0.7 V, R_{EXT} = 250 Ω	64.2	75.5	86.8	mA
		V_{CE} = 0.7 V, R_{EXT} = 470 Ω	34.1	40.0	45.9	mA
Output Current Matching (difference between any two outputs at same V_{CE})	ΔI_O	0.4 V ≤ $V_{CE(A)}$ = $V_{CE(B)}$ ≤ 0.7 V:				
		R_{EXT} = 250 Ω	–	±1.5	±6.0	%
		R_{EXT} = 470 Ω	–	±1.5	±6.0	%
Output Leakage Current	I_{CEX}	V_{OH} = 15 V	–	1.0	5.0	µA
Logic Input Voltage	V_{IH}		0.7V_{DD}	–	V_{DD}	V
	V_{IL}		GND	–	0.3V_{DD}	V
SERIAL DATA OUT Voltage	V_{OL}	I_{OL} = 500 µA	–	–	0.4	V
	V_{OH}	I_{OH} = -500 µA	4.6	–	–	V
Input Resistance	R_I	ENABLE Input, Pull Up	150	300	600	kΩ
		LATCH Input, Pull Down	100	200	400	kΩ
Supply Current	$I_{DD(OFF)}$	R_{EXT} = open, V_{OE} = 5 V	–	0.8	1.4	mA
		R_{EXT} = 470 Ω, V_{OE} = 5 V	3.5	6.0	8.0	mA
		R_{EXT} = 250 Ω, V_{OE} = 5 V	6.5	11	15	mA
	$I_{DD(ON)}$	R_{EXT} = 470 Ω, V_{OE} = 0 V	7.0	13	20	mA
		R_{EXT} = 250 Ω, V_{OE} = 0 V	10	22	32	mA

Typical Data is at V_{DD} = 5 V and is for design information only.

Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

SWITCHING CHARACTERISTICS at T_A = 25°C, V_{DD} = V_{IH} = 5 V, V_{CE} = 0.4 V, V_{IL} = 0 V, R_{EXT} = 470 Ω, I_O = 40 mA, V_L = 3 V, R_L = 65 Ω, C_L = 10.5 pF.

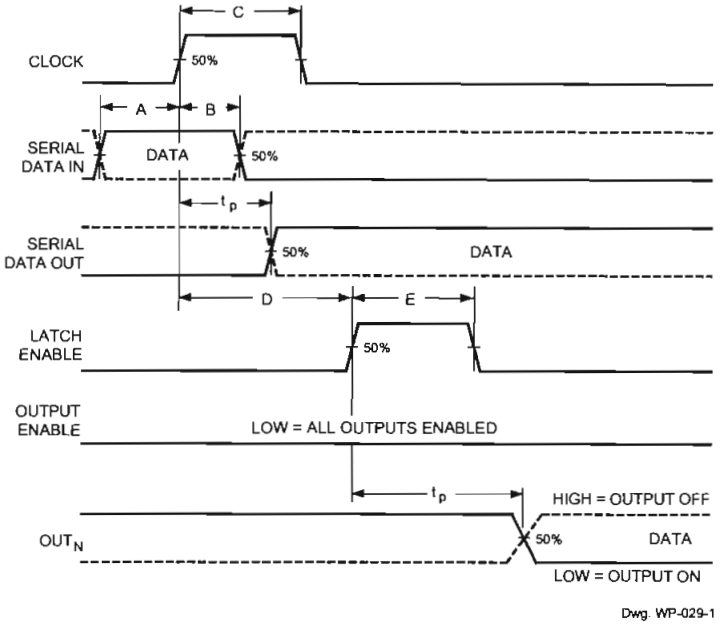
Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Unit
Propagation Delay Time	t_{pHL}	CLOCK-OUT_n	–	350	1000	ns
		LATCH-OUT_n	–	350	1000	ns
		ENABLE-OUT_n	–	350	1000	ns
		CLOCK-SERIAL DATA OUT	–	40	–	ns
Propagation Delay Time	t_{pLH}	CLOCK-OUT_n	–	300	1000	ns
		LATCH-OUT_n	–	300	1000	ns
		ENABLE-OUT_n	–	300	1000	ns
		CLOCK-SERIAL DATA OUT	–	40	–	ns
Output Fall Time	t_f	90% to 10% voltage	150	350	1000	ns
Output Rise Time	t_r	10% to 90% voltage	150	300	600	ns

RECOMMENDED OPERATING CONDITIONS

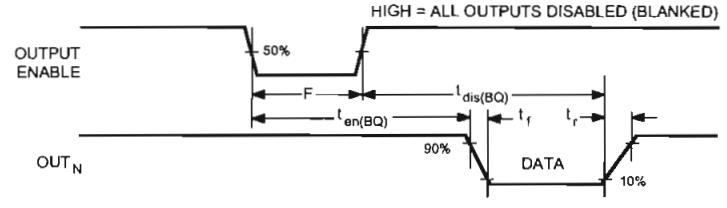
Characteristic	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DD}		4.5	5.0	5.5	V
Output Voltage	V_O		–	1.0	4.0	V
Output Current	I_O	Continuous, any one output	–	–	90	mA
	I_{OH}	SERIAL DATA OUT	–	–	-1.0	mA
	I_{OL}	SERIAL DATA OUT	–	–	1.0	mA
Logic Input Voltage	V_{IH}		$0.7V_{DD}$	–	V_{DD} + 0.3	V
	V_{IL}		-0.3	–	$0.3V_{DD}$	V
Clock Frequency	f_{CK}	Cascade operation	–	–	10	MHz

TIMING REQUIREMENTS and SPECIFICATIONS

(Logic Levels are V_{DD} and Ground)



Dwg. WP-029-1



Dwg. WP-030-1

A. Data Active Time Before Clock Pulse (Data Set-Up Time), $t_{su(D)}$ ..... **60 ns**
B. Data Active Time After Clock Pulse (Data Hold Time), $t_{h(D)}$ ..... **20 ns**
C. Clock Pulse Width, $t_{w(CK)}$ ..... **50 ns**
D. Time Between Clock Activation and Latch Enable, $t_{su(L)}$ ..... **100 ns**
E. Latch Enable Pulse Width, $t_{w(L)}$ ..... **100 ns**
F. Output Enable Pulse Width, $t_{w(OE)}$ ..... **4.5 µs**
NOTE – Timing is representative of a 10 MHz clock. Significantly higher speeds are attainable.
— Max. Clock Transition Time, t_r or t_f ..... **10 µs**

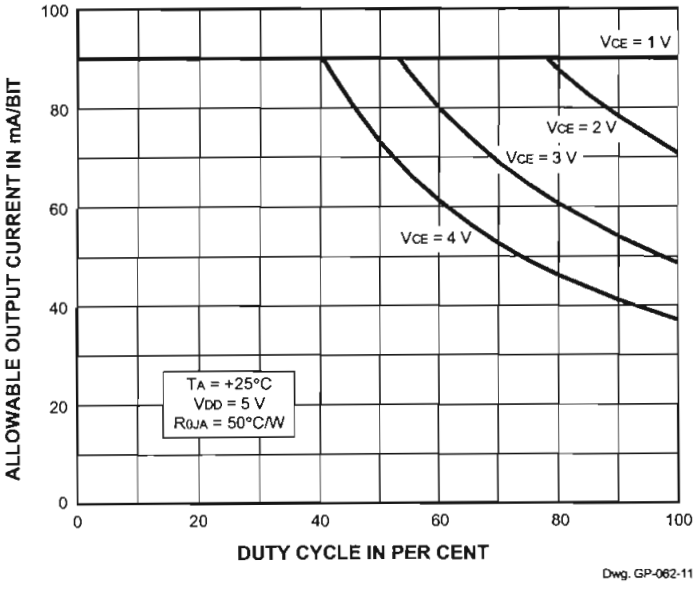
Information present at any register is transferred to the respective latch when the LATCH ENABLE is high (serial-to-parallel conversion). The latches will continue to accept new data as long as the LATCH ENABLE is held high. Applications where the latches are bypassed (LATCH ENABLE tied high) will require that the OUTPUT ENABLE input be high during serial data entry.

When the OUTPUT ENABLE input is high, the output source drivers are disabled (OFF). The information stored in the latches is not affected by the OUTPUT ENABLE input. With the OUTPUT ENABLE input low, the outputs are controlled by the state of their respective latches.

Allegro® MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

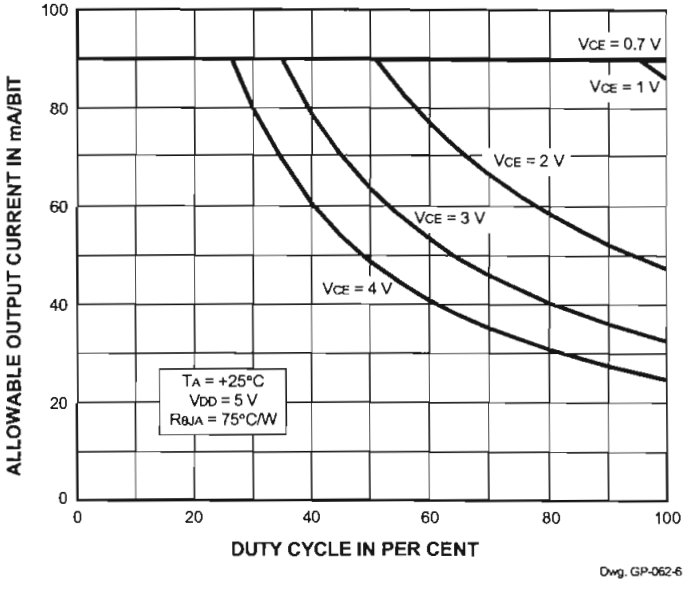
ALLOWABLE OUTPUT CURRENT AS A FUNCTION OF DUTY CYCLE

A6276EA

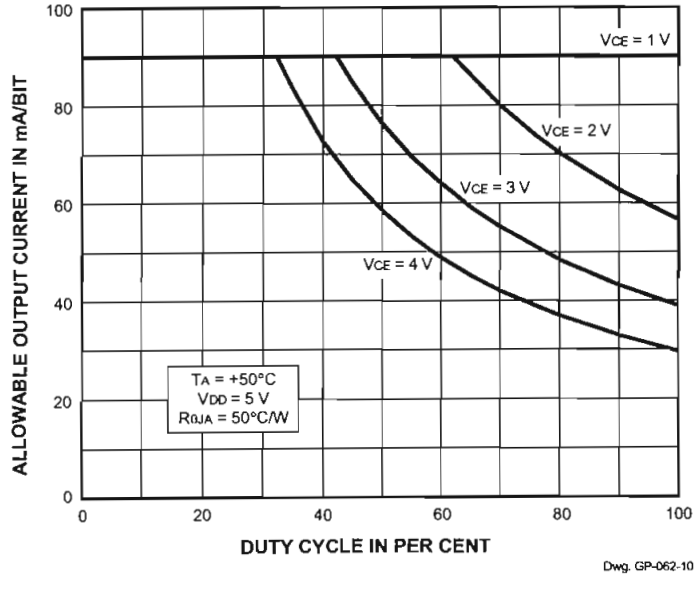


Dwg. GP-062-11

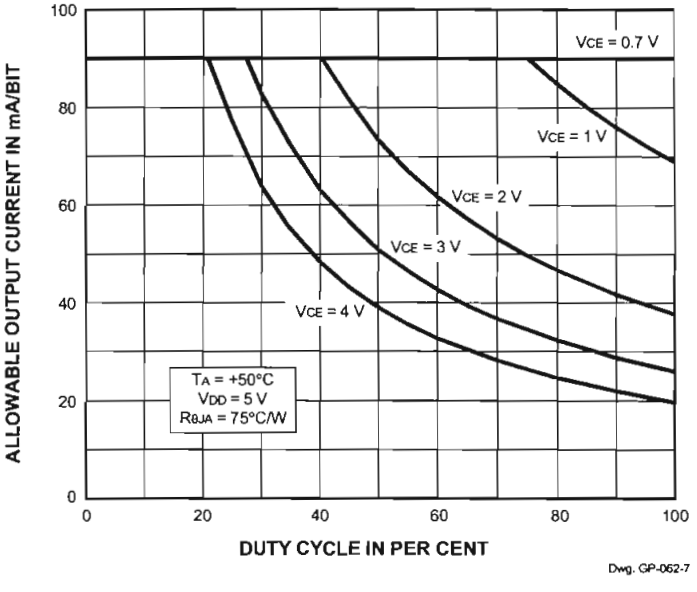
A6276ELW



Dwg. GP-062-6



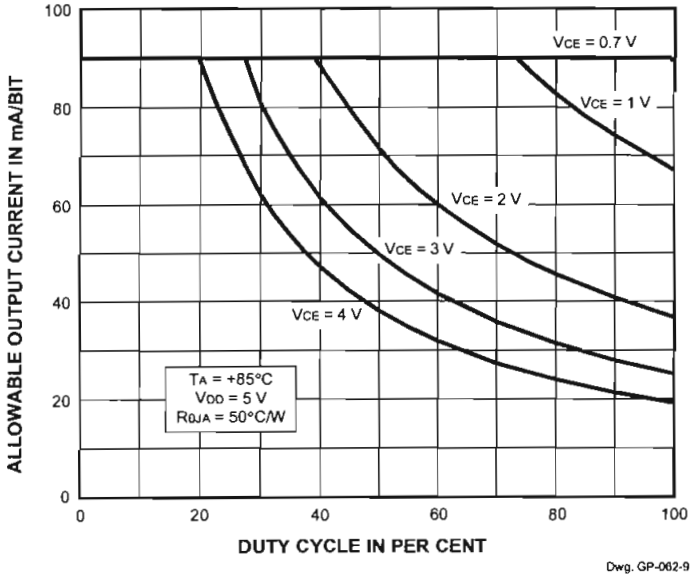
Dwg. GP-062-10



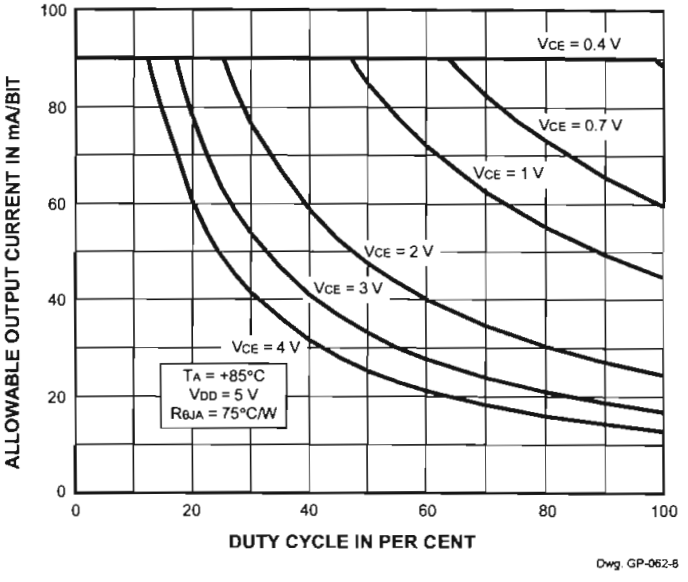
Dwg. GP-062-7

ALLOWABLE OUTPUT CURRENT AS A FUNCTION OF DUTY CYCLE (cont.)

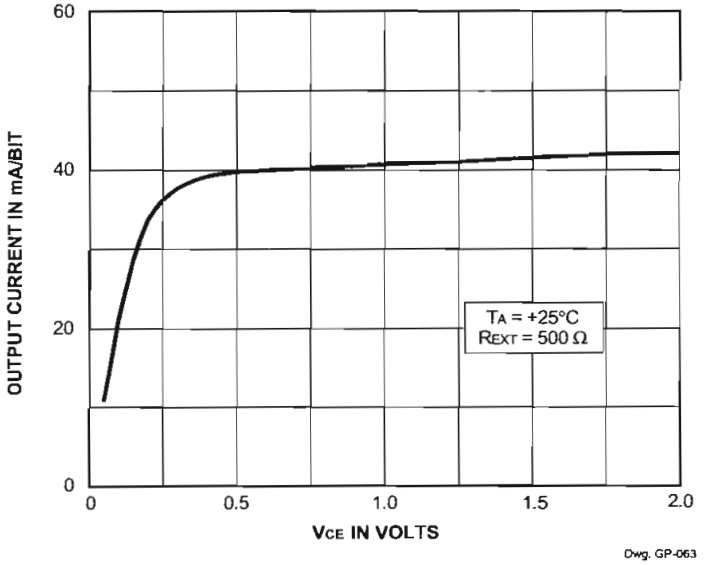
A6276EA



A6276ELW



TYPICAL CHARACTERISTICS



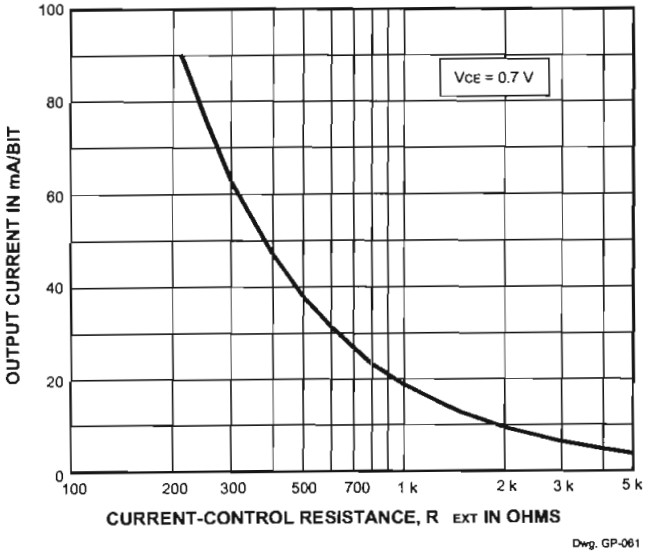
Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

TERMINAL DESCRIPTION

Terminal No.	Terminal Name	Function
1	GND	Reference terminal for control logic.
2	SERIAL DATA IN	Serial-data input to the shift-register.
3	CLOCK	Clock input terminal for data shift on rising edge.
4	LATCH ENABLE	Data strobe input terminal; serial data is latched with high-level input.
5-20	OUT_{0-15}	The 16 current-sinking output terminals.
21	OUTPUT ENABLE	When (active) low, the output drivers are enabled; when high, all output drivers are turned OFF (blanked).
22	SERIAL DATA OUT	CMOS serial-data output to the following shift-register.
23	R_{EXT}	An external resistor at this terminal establishes the output current for all sink drivers.
24	SUPPLY	(V_{DD}) The logic supply voltage (typically 5 V).

Applications Information

The load current per bit (I_O) is set by the external resistor (R_{EXT}) as shown in the figure below.



Package Power Dissipation (P_D). The maximum allowable package power dissipation is determined as

$$P_D(max) = (150 - T_A)/R_{\theta JA}.$$

The actual package power dissipation is

$$P_D(act) = dc(V_{CE} \bullet I_O \bullet 16) + (V_{DD} \bullet I_{DD}).$$

When the load supply voltage is greater than 3 V to 5 V, considering the package power dissipating limits of these devices, or if $P_D(act) > P_D(max)$, an external voltage reducer (V_{DROP}) should be used.

Load Supply Voltage (V_{LED}). These devices are designed to operate with driver voltage drops (V_{CE}) of 0.4 V to 0.7 V with LED forward voltages (V_F) of 1.2 V to 4.0 V. If higher voltages are dropped across the driver, package power dissipation will be increased significantly. To minimize package power dissipation, it is recommended to use the lowest possible load supply voltage or to set any series dropping voltage (V_{DROP}) as

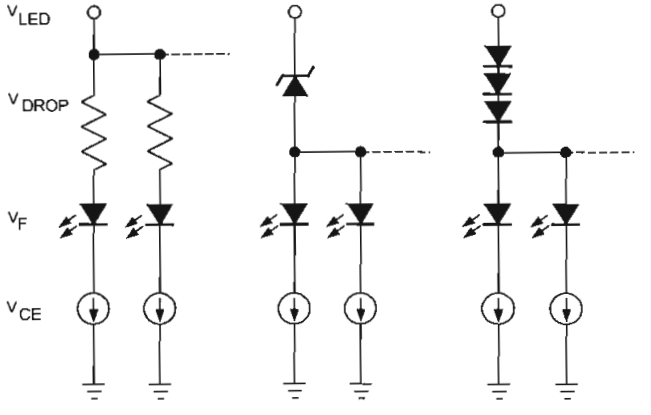
$$V_{DROP} = V_{LED} - V_F - V_{CE}$$

with $V_{DROP} = I_o \bullet R_{DROP}$ for a single driver, or a Zener diode (V_Z), or a series string of diodes (approximately 0.7 V per diode) for a group of drivers. If the available voltage source will cause unacceptable dissipation and series resistors or diode(s) are undesirable, a regulator such as the Sanken Series SAI or Series SI can be used to provide supply voltages as low as 3.3 V.

For reference, typical LED forward voltages are:

Blue	3.0 – 4.0 V
Green	1.8 – 2.2 V
Yellow	2.0 – 2.1 V
Amber	1.9 – 2.65 V
Red	1.6 – 2.25 V
Infrared	1.2 – 1.5 V

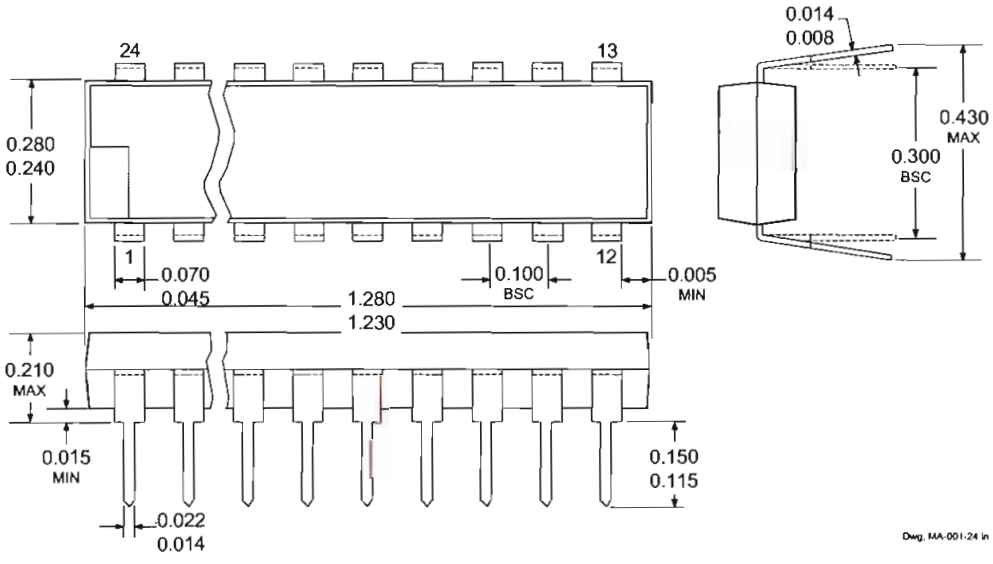
Pattern Layout. This device has a common logic-ground and power-ground terminal. If ground pattern layout contains large common-mode resistance, and the voltage between the system ground and the LATCH ENABLE or CLOCK terminals exceeds 2.5 V (because of switching noise), these devices may not operate correctly.



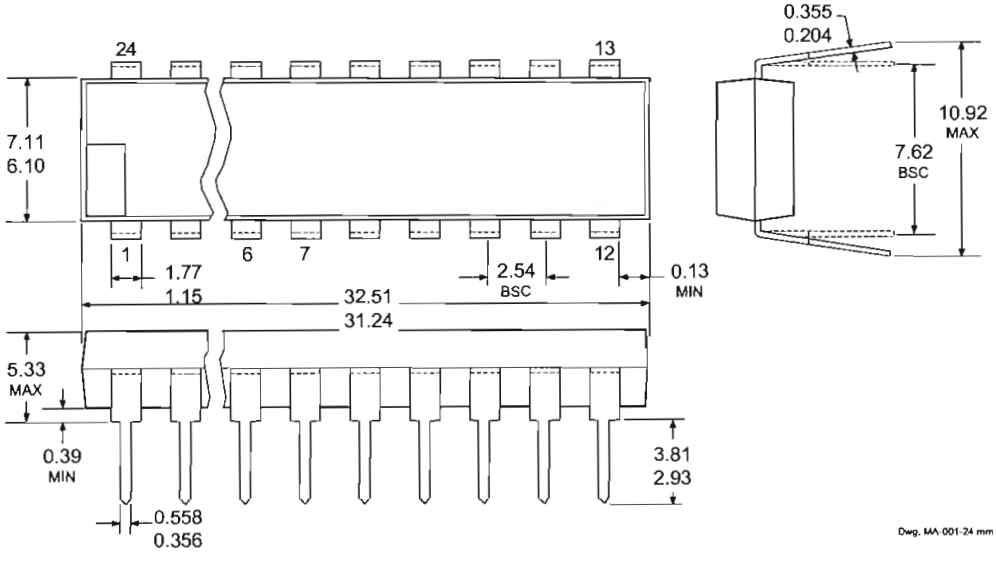
Allegro MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

A6276EA

Dimensions in Inches
(controlling dimensions)



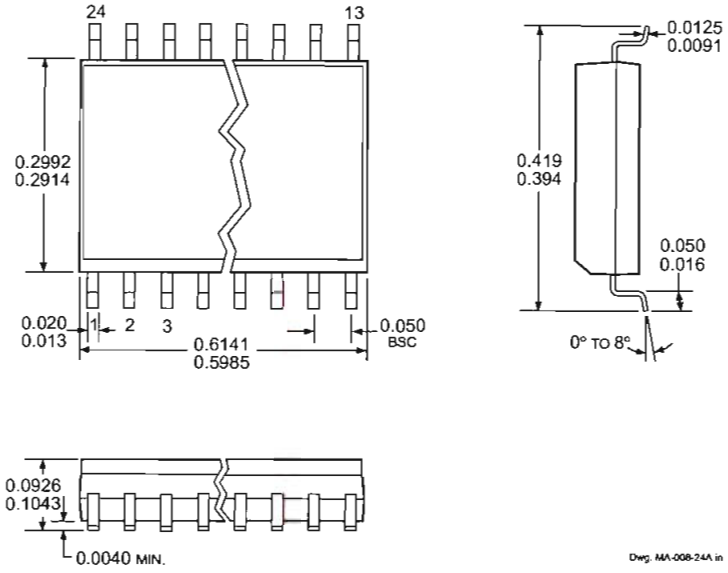
Dimensions in Millimeters
(for reference only)



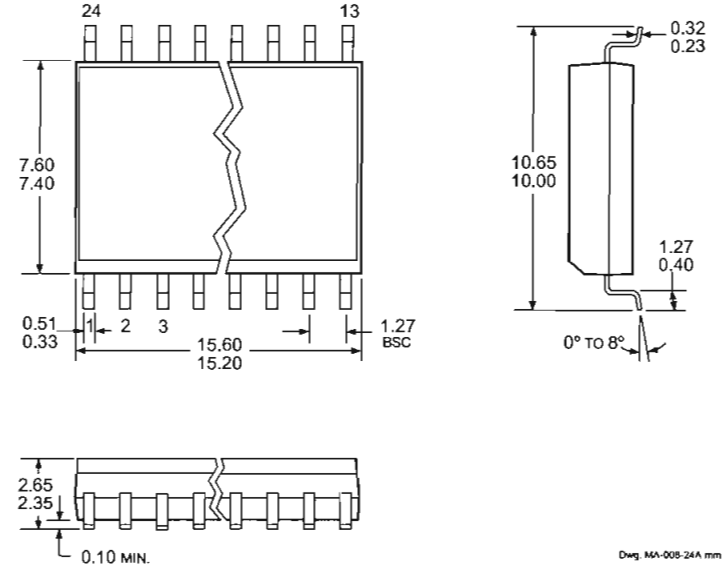
NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
2. Lead spacing tolerance is non-cumulative
3. Lead thickness is measured at seating plane or below.
4. Supplied in standard sticks/tubes of 15 devices.

A6276ELW

Dimensions in Inches
(for reference only)



Dimensions in Millimeters
(controlling dimensions)



NOTES: 1. Exact body and lead configuration at vendor's option within limits shown.
2. Lead spacing tolerance is non-cumulative.
3. Supplied in standard sticks/tubes of 31 devices or add "TR" to part number for tape and reel.

Allegro® MicroSystems, Inc.
115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

Features

- *Compatible with* MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

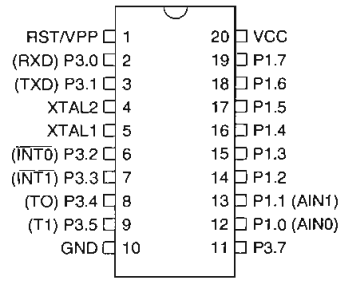
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Configuration

PDIP/SOIC

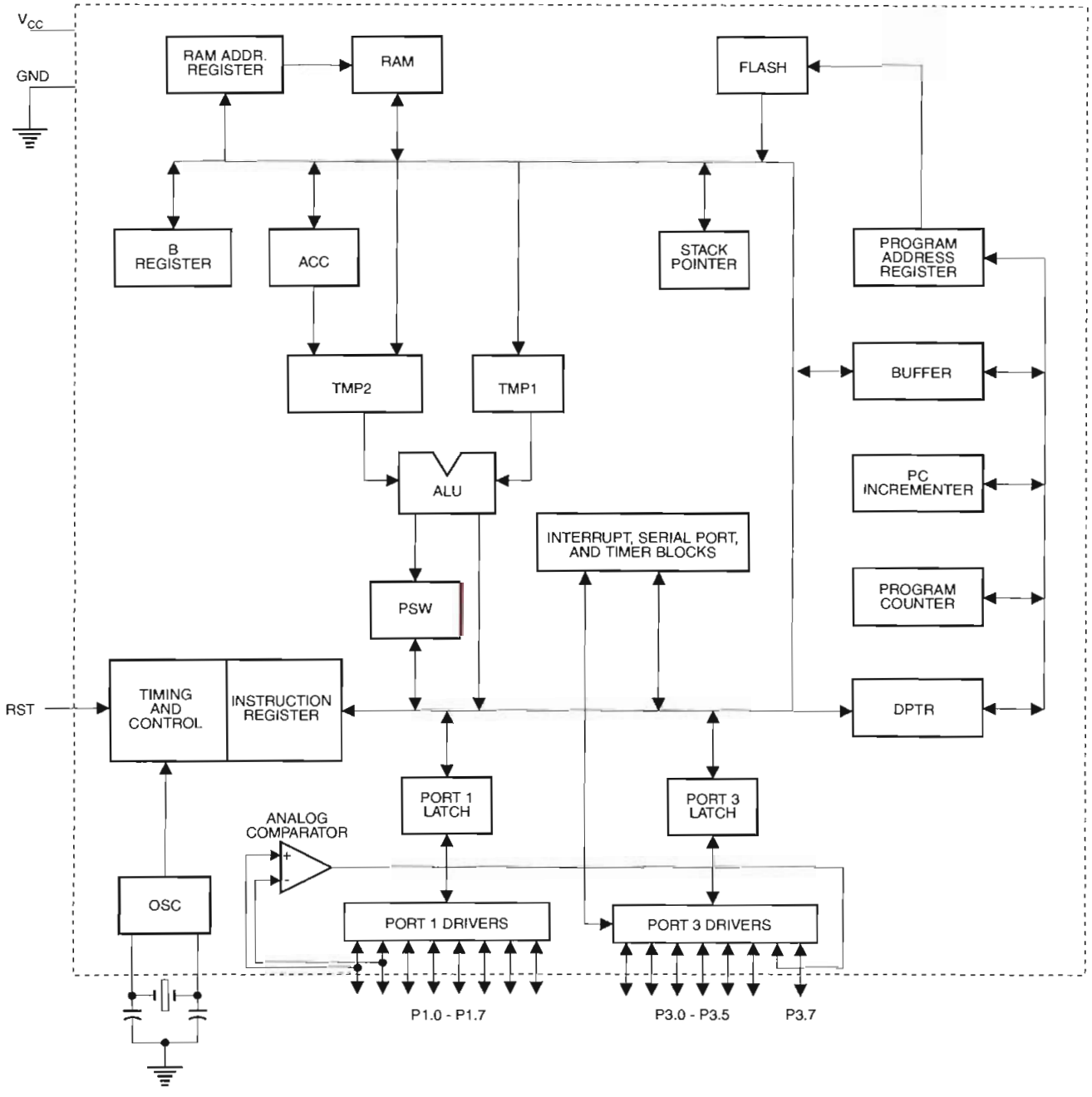


8-bit Microcontroller with 2K Bytes Flash

AT89C2051

Rev. 0368E–02/00

Block Diagram



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bi-irectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-irectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

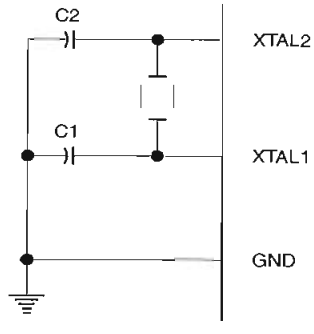
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

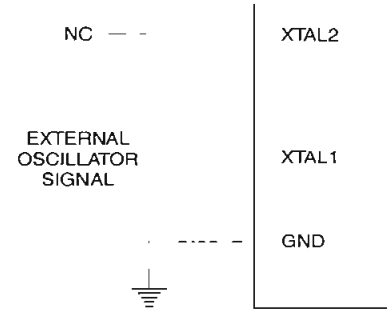
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power-down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
2. Set pin RST to "H"
   Set pin P3.2 to "H"
3. Apply the appropriate combination of "H" or "L" logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic "H" level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
10. Power-off sequence:
    set XTAL1 to "L"
    set RST to "L"
    Turn V_{CC} power off

$\overline{\text{Data}}$ Polling: The AT89C2051 features $\overline{\text{Data}}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The Progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from "L" to "H".
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST/VPP	P3.2/$\overline{PROG}$	P3.3	P3.4	P3.5	P3.7
Write Code Data[1][3]		12V		L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V	(2)	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.
3. P3.1 is pulled Low during programming to indicate RDY/$\overline{BSY}$.

Figure 3. Programming the Flash Memory

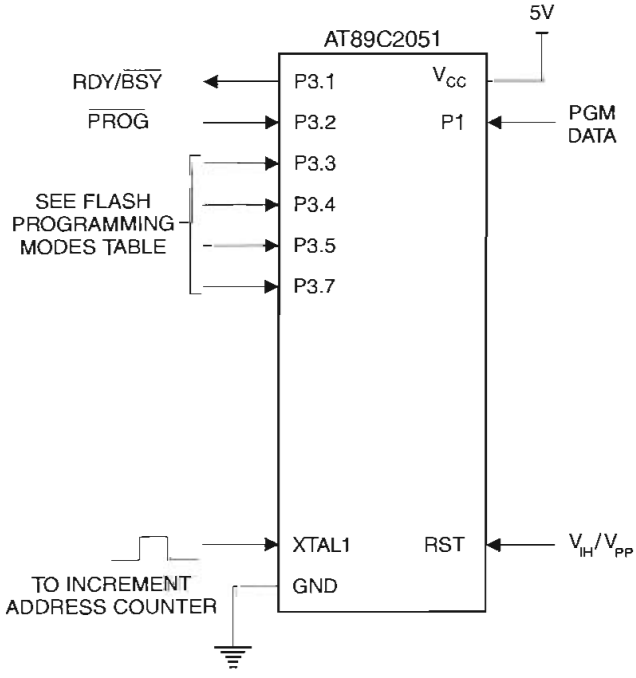
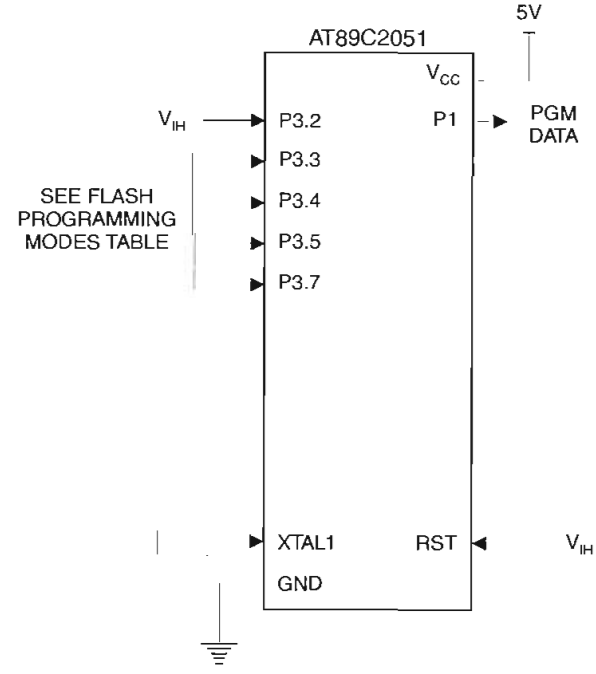


Figure 4. Verifying the Flash Memory



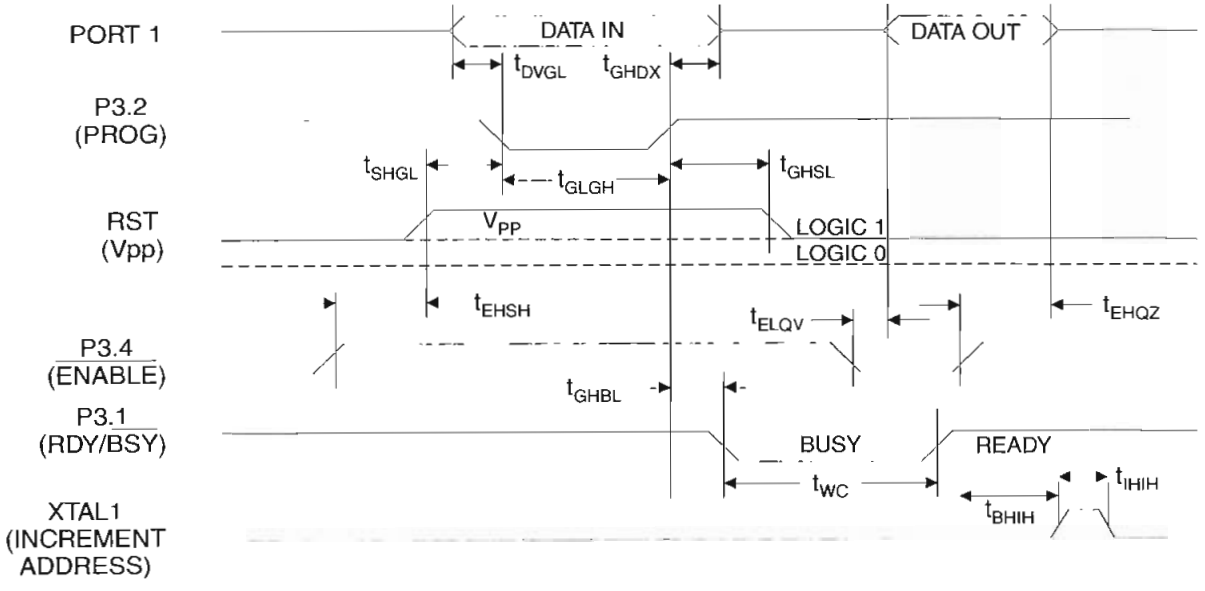
Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$\ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

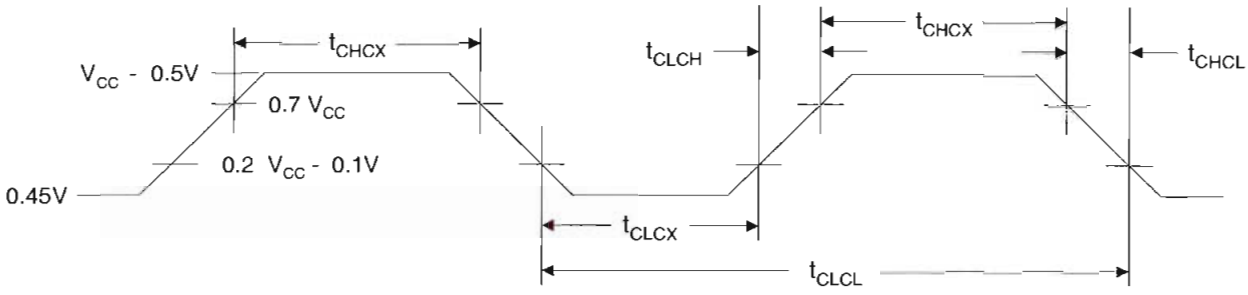
DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.0V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.2 V_{CC} - 0.1	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1, 3)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 2.7V		0.5	V
V_{OH}	Output High-voltage (Ports 1, 3)	I_{OH} = -80 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -30 µA	0.75 V_{CC}		V
		I_{OH} = -12 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-750	µA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V_{IN} < V_{CC}		±10	µA
V_{OS}	Comparator Input Offset Voltage	V_{CC} = 5V		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 6V/3V		15/5.5	mA
		Idle Mode, 12 MHz, V_{CC} = 6V/3V P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power-down Mode[2]	V_{CC} = 6V P1.0 & P1.1 = 0V or V_{CC}		100	µA
		V_{CC} = 3V P1.0 & P1.1 = 0V or V_{CC}		20	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 20 mA
   Maximum total I_{OL} for all output pins: 80 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

External Clock Drive Waveforms



External Clock Drive

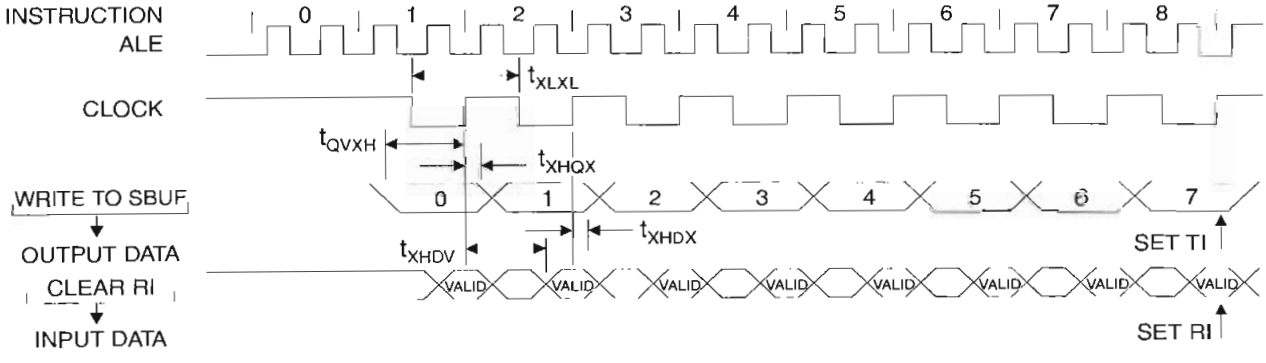
Symbol	Parameter	V_{CC} = 2.7V to 6.0V		V_{CC} = 4.0V to 6.0V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

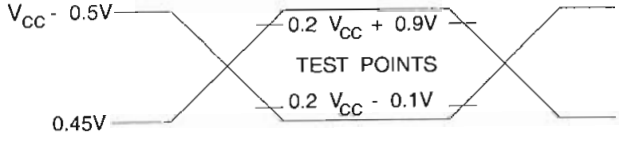
V_{CC} = 5.0V ± 20%; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

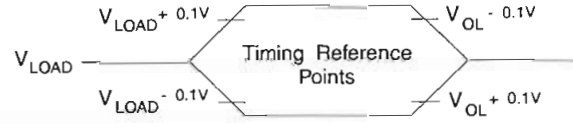


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

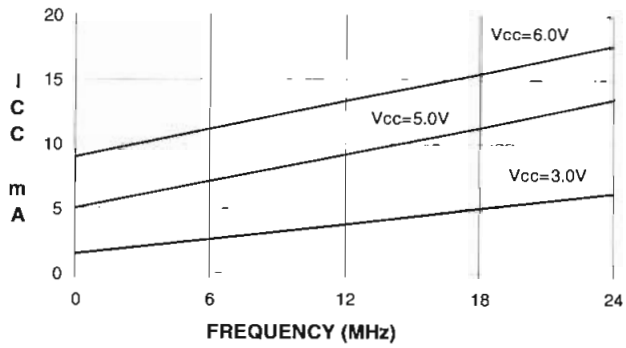
Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change frothe loaded V_{OH}/V_{OL} level occurs.

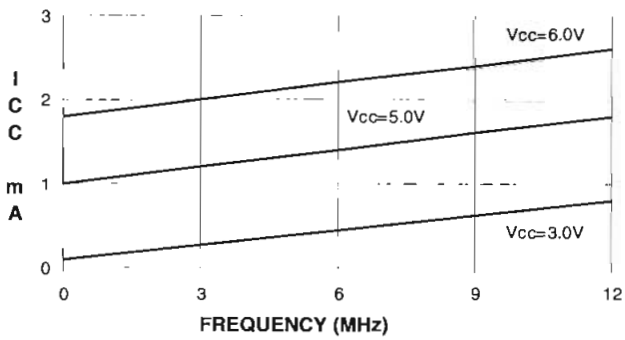
AT89C2051

TYPICAL ICC - ACTIVE (85°C)



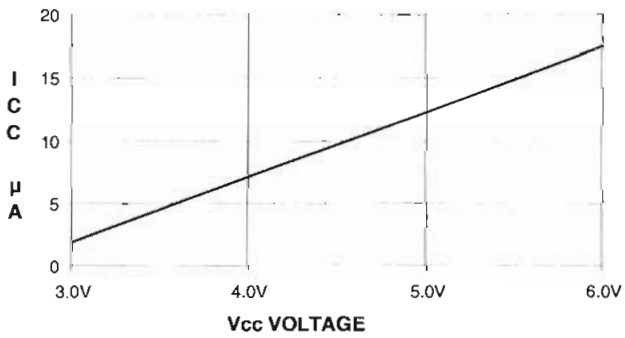
AT89C2051

TYPICAL ICC - IDLE (85°C)



AT89C2051

TYPICAL ICC vs. VOLTAGE- POWER DOWN (85°C)



Notes:
1. XTAL1 tied to GND for I_{CC} (power-down)
2. P.1.0 and P1.1 = V_{CC} or GND
3. Lock bits programmed

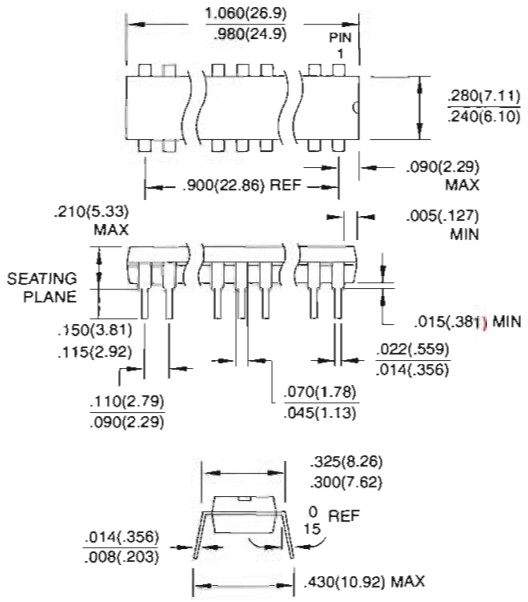
Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0V to 6.0V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)

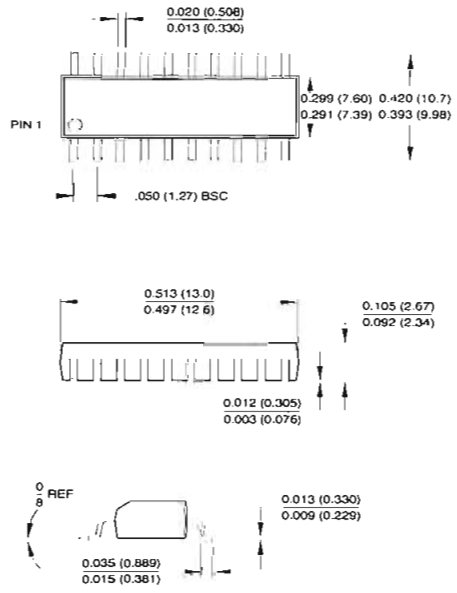
Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

Packaging Information

20P3, 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-001 AD



20S, 20-lead, 0.300" Wide, Plastic Gull WIng Small Outline (SOIC)
Dimensions in Inches and (Millimeters)



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

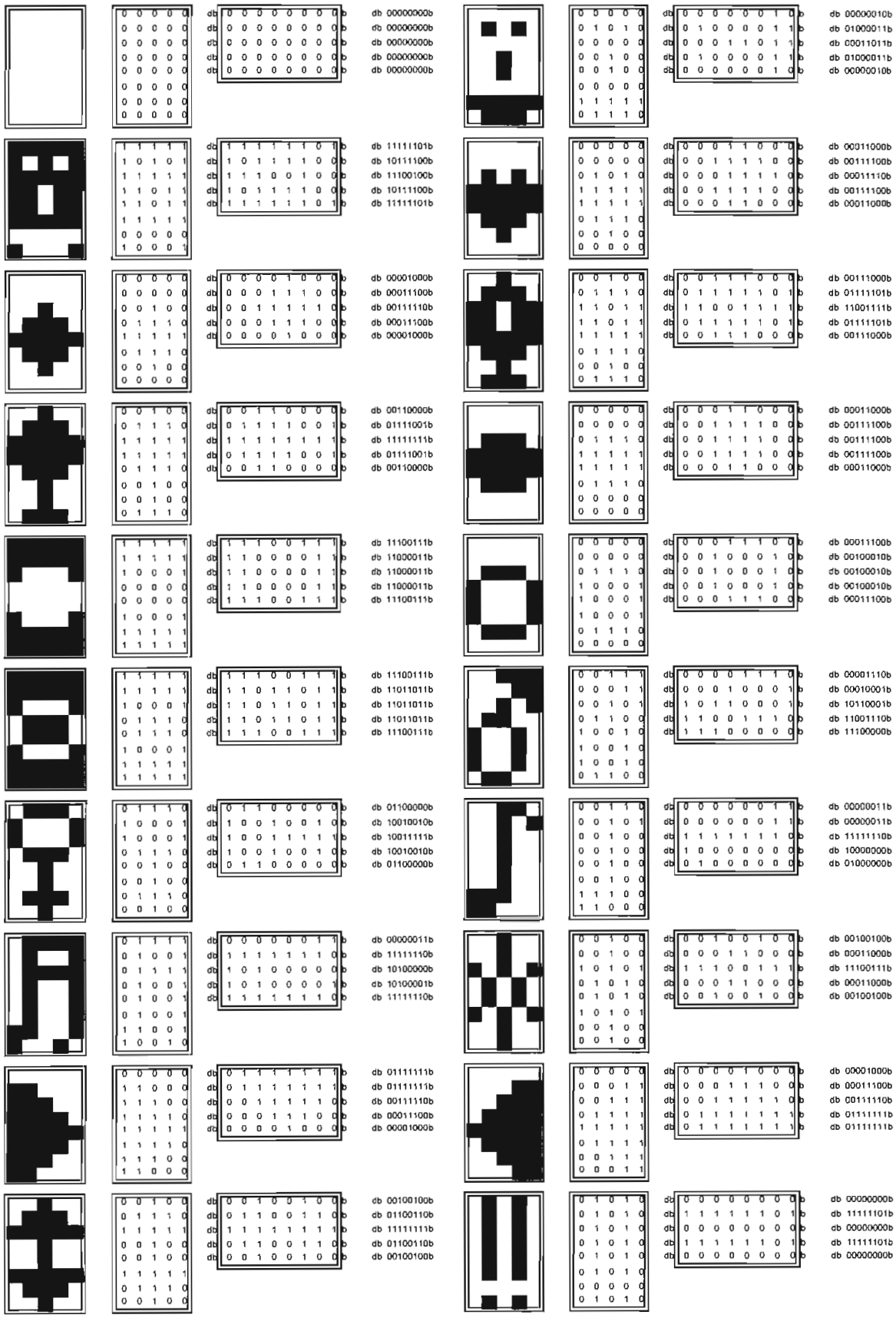
Web Site
http://www.atmel.com

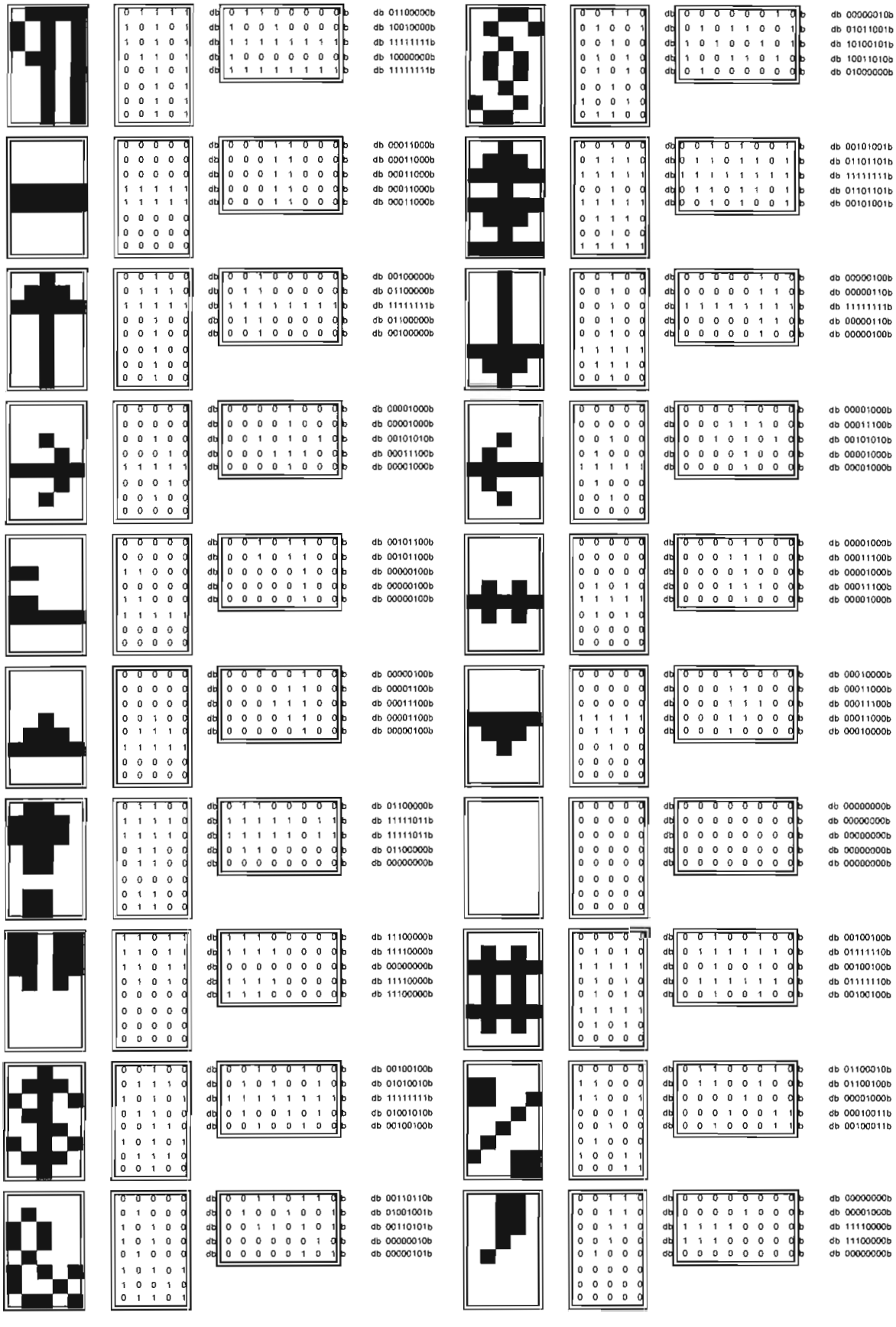
BBS
1-(408) 436-4309

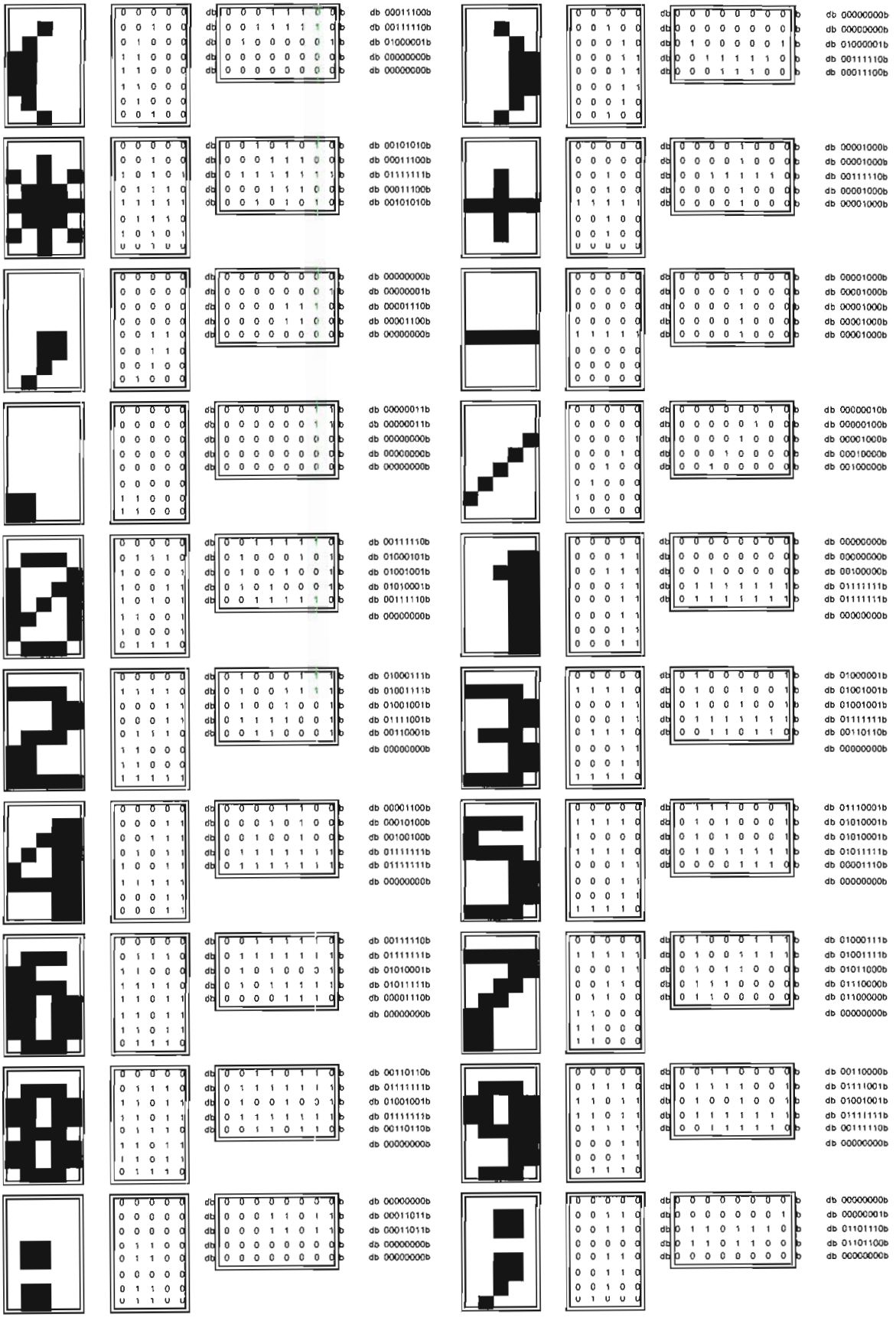
0368E–02/00/xM

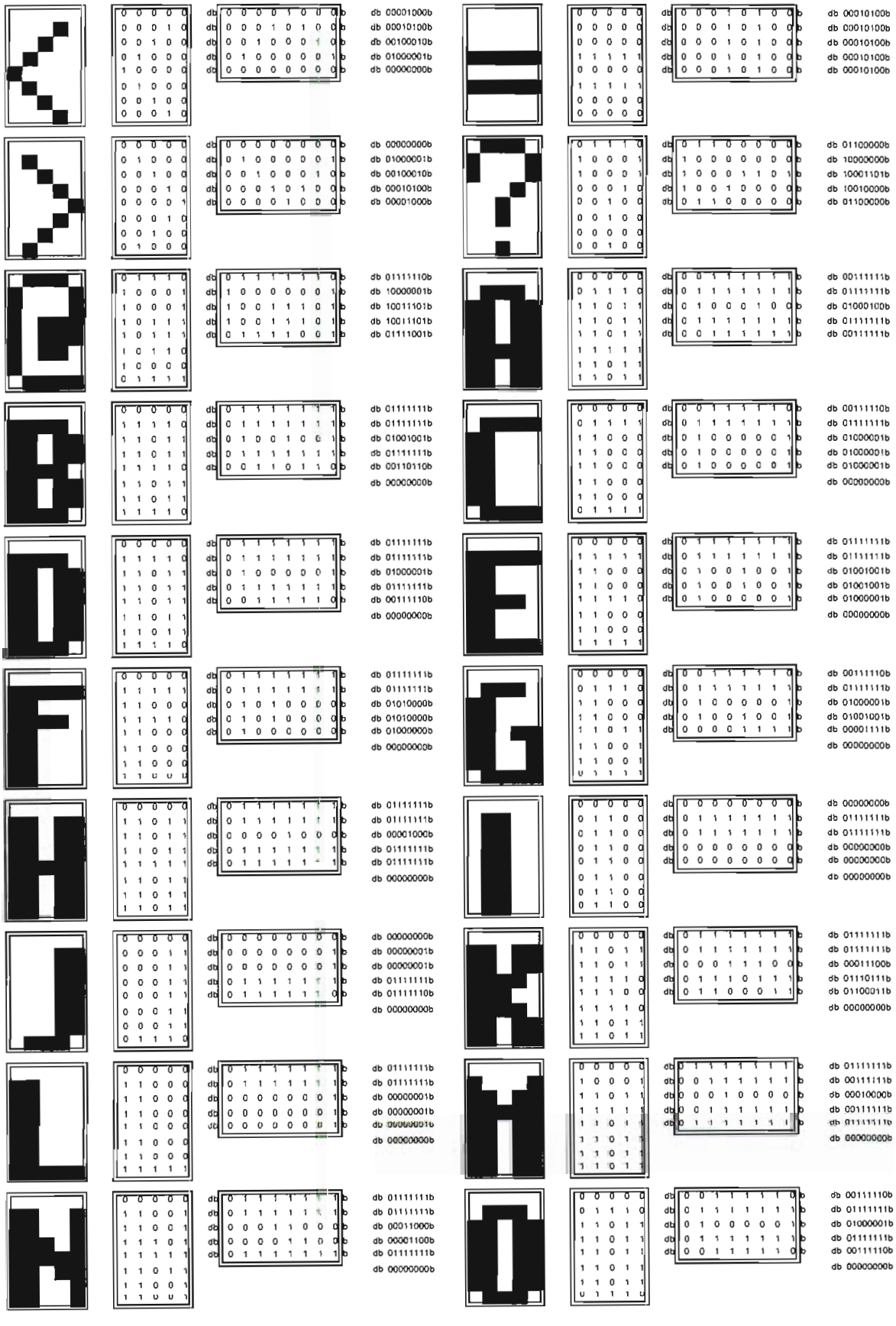
ANEXO B

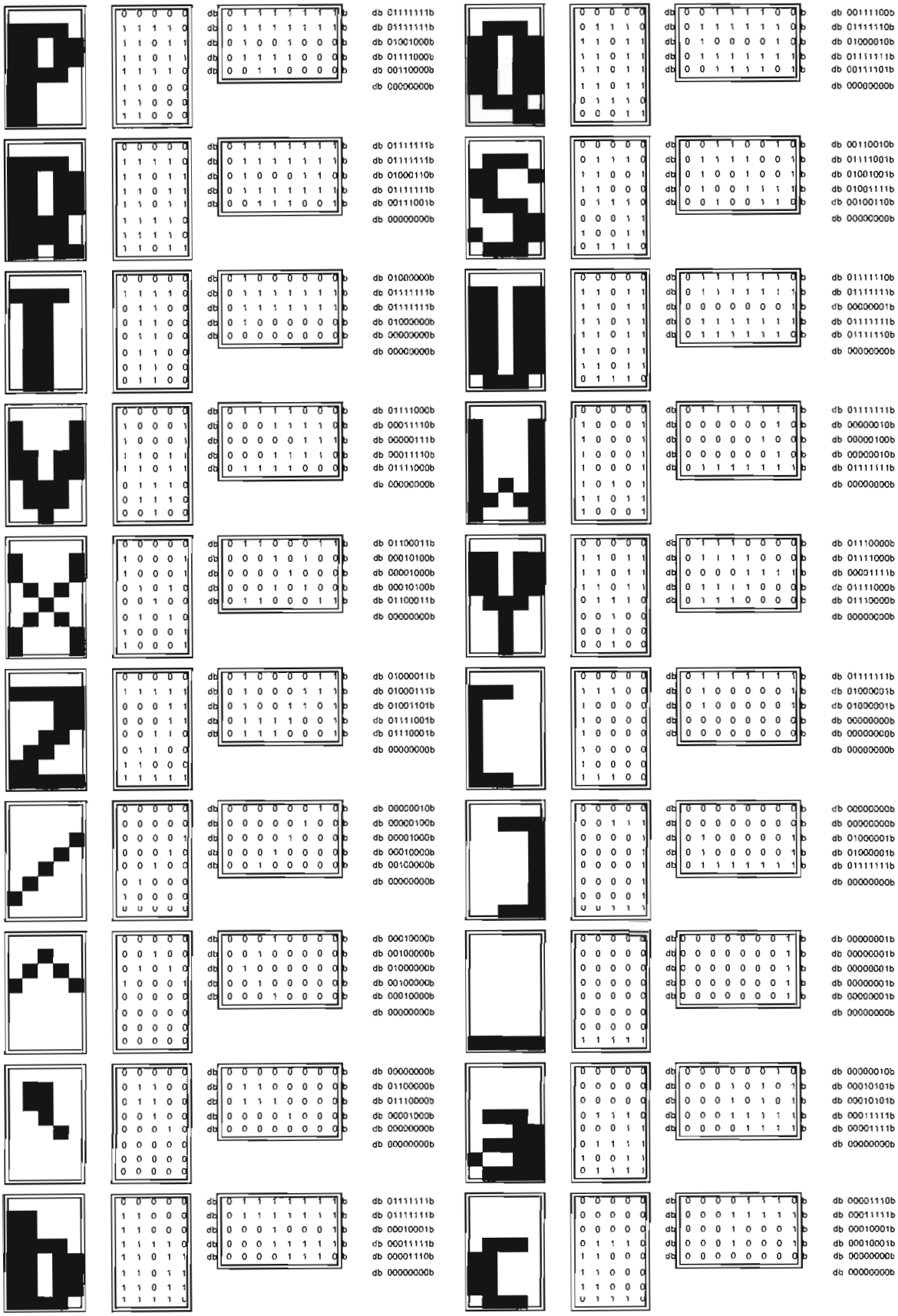
MAPEO DE LA TABLA DE DATOS DE LOS CARACTERES ASCII

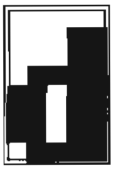



db 01100000b
db 10010000b
db 11111111b
db 10000000b
db 11111111b
db 00000010b
db 01011001b
db 10100101b
db 10011010b
db 01000000b
db 00011000b
db 00011000b
db 00011000b
db 00011000b
db 00011000b
db 00101001b
db 01101101b
db 11111111b
db 01101101b
db 00101001b
db 00100000b
db 01100000b
db 00100000b
db 00000100b
db 00000110b
db 11111111b
db 00000110b
db 00000100b
db 00001000b
db 00001000b
db 00101010b
db 00011100b
db 00001000b
db 00001000b
db 00011100b
db 00101010b
db 00001000b
db 00101100b
db 00101100b
db 00000100b
db 00000100b
db 00000100b
db 00001000b
db 00011100b
db 00001000b
db 00011100b
db 00000100b
db 00001100b
db 00011100b
db 00000100b
db 00010000b
db 00011000b
db 00011100b
db 00011000b
db 00010000b
db 01100000b
db 11111011b
db 11111011b
db 01100000b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 11100000b
db 11110000b
db 00000000b
db 11110000b
db 11100000b
db 00100100b
db 01111110b
db 00100100b
db 01111110b
db 00100100b
db 00100100b
db 01010010b
db 11111111b
db 01001010b
db 00100100b
db 01100010b
db 01100100b
db 00001000b
db 00010011b
db 00100011b
db 00110110b
db 01001001b
db 00110101b
db 00000010b
db 00000101b
db 00000000b
db 00001000b
db 11110000b
db 11100000b
db 00000000b


db 00011100b
db 00111110b
db 01000001b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 01000001b
db 00111110b
db 00011100b
db 00101010b
db 00011100b
db 01111111b
db 00011100b
db 00101010b
db 00001000b
db 00001000b
db 00111110b
db 00001000b
db 00001000b
db 00000000b
db 00000001b
db 00001110b
db 00001100b
db 00000000b
db 00001000b
db 00001000b
db 00001000b
db 00001000b
db 00001000b
db 00000011b
db 00000011b
db 00000000b
db 00000000b
db 00000000b
db 00000010b
db 00000100b
db 00001000b
db 00010000b
db 00100000b
db 00111110b
db 01000101b
db 01001001b
db 01010001b
db 00111110b
db 00000000b
db 00000000b
db 00000000b
db 00100000b
db 01111111b
db 01111111b
db 00000000b
db 01000111b
db 01001111b
db 01001001b
db 01111001b
db 01110001b
db 00000000b
db 01000001b
db 01001001b
db 01001001b
db 01111111b
db 00110110b
db 00000000b
db 00001100b
db 00010100b
db 00100100b
db 01111111b
db 01111111b
db 00000000b
db 01110001b
db 01010001b
db 01010001b
db 01011111b
db 00001110b
db 00000000b
db 00111110b
db 01111111b
db 01010001b
db 01011111b
db 00001110b
db 00000000b
db 01000111b
db 01001111b
db 01011000b
db 01110000b
db 01100000b
db 00000000b
db 00110110b
db 01111111b
db 01001001b
db 01111111b
db 00110110b
db 00000000b
db 00110000b
db 01111001b
db 01001001b
db 01111111b
db 00111110b
db 00000000b
db 00000000b
db 00011011b
db 00011011b
db 00000000b
db 00000000b
db 00000000b
db 01101110b
db 01101100b
db 00000000b


db 00001000b
db 00010100b
db 00100010b
db 01000001b
db 00000000b
db 00010100b
db 00010100b
db 00010100b
db 00010100b
db 00010100b
db 00000000b
db 01000001b
db 00100010b
db 00010100b
db 00001000b
db 01100000b
db 10000000b
db 10001101b
db 10010000b
db 01100000b
db 01111110b
db 10000001b
db 10011101b
db 10011101b
db 01111001b
db 00111111b
db 01111111b
db 01000001b
db 01111111b
db 00111111b
db 01111111b
db 01111111b
db 01001001b
db 01111111b
db 00110110b
db 00000000b
db 00111110b
db 01111111b
db 01000001b
db 01000001b
db 00000000b
db 01111111b
db 01111111b
db 01000001b
db 01111111b
db 00111110b
db 00000000b
db 01111111b
db 01111111b
db 01001001b
db 01001001b
db 01000001b
db 00000000b
db 01111111b
db 01111111b
db 01010000b
db 01010000b
db 01000000b
db 00000000b
db 00111110b
db 01111111b
db 01000001b
db 01001001b
db 00001111b
db 00000000b
db 01111111b
db 01111111b
db 00001000b
db 01111111b
db 01111111b
db 00000000b
db 00000000b
db 01111111b
db 01111111b
db 01111111b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 01111111b
db 00000001b
db 00000001b
db 01111111b
db 01111110b
db 00000000b
db 01111111b
db 00111111b
db 00010000b
db 00111111b
db 01111111b
db 00000000b
db 01111111b
db 00111111b
db 00011000b
db 00001100b
db 01111111b
db 00000000b
db 00111110b
db 01111111b
db 01000001b
db 01111111b
db 00111110b
db 00000000b

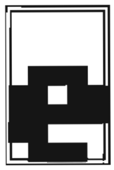

db 01111111b
db 01111111b
db 01001000b
db 01111000b
db 00110000b
db 00000000b
db 00111100b
db 01111110b
db 01000010b
db 01111111b
db 00111101b
db 00000000b
db 01111111b
db 01111111b
db 01000110b
db 01111111b
db 00111001b
db 00000000b
db 00110010b
db 01111001b
db 01001001b
db 01001111b
db 00100110b
db 00000000b
db 01000000b
db 01111111b
db 01111111b
db 01000000b
db 00000000b
db 00000000b
db 01111110b
db 01111111b
db 00000001b
db 01111111b
db 01111110b
db 00000000b
db 01111000b
db 00011110b
db 00000111b
db 00011110b
db 01111000b
db 00000000b
db 01111111b
db 00000010b
db 00000100b
db 00000010b
db 01111111b
db 00000000b
db 01100011b
db 00010100b
db 00001000b
db 00010100b
db 01100011b
db 00000000b
db 01110000b
db 01111000b
db 00001111b
db 01111000b
db 01110000b
db 00000000b
db 01000011b
db 01000111b
db 01001101b
db 01111001b
db 01110001b
db 00000000b
db 01111111b
db 01000001b
db 01000001b
db 00000000b
db 00000000b
db 00000010b
db 00000100b
db 00001000b
db 00010000b
db 00100000b
db 00000000b
db 00000000b
db 01000001b
db 01000001b
db 01111111b
db 00000000b
db 00010000b
db 00100000b
db 01000000b
db 00100000b
db 00010000b
db 00000000b
db 00000001b
db 00000001b
db 00000001b
db 00000001b
db 00000001b
db 00000000b
db 00000000b
db 01100000b
db 01110000b
db 00001000b
db 00000000b
db 00000010b
db 00010101b
db 00010101b
db 00011111b
db 00001111b
db 00000000b
db 01111111b
db 01111111b
db 00010001b
db 00011111b
db 00001110b
db 00000000b
db 00001110b
db 00011111b
db 00010001b
db 00010001b
db 00000000b
db 00000000b



```
0 0 0 0 0
0 0 0 1 1
0 0 0 1 1
0 1 1 1 1
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
0 1 1 1 1
```

```
db 0 0 0 0 1 1 1 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 0 0 0 1 b
db 0 1 1 1 1 1 1 1 b
db 0 1 1 1 1 1 1 1 b
```

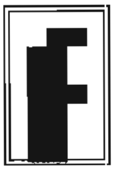
```
db 00001110b
db 00011111b
db 00010001b
db 01111111b
db 01111111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 1 0 1 1
1 1 1 1 1
1 1 0 0 0
0 1 1 1 0
```

```
db 0 0 0 0 1 1 1 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 0 1 0 1 b
db 0 0 0 1 1 1 0 1 b
db 0 0 0 0 1 1 0 0 b
```

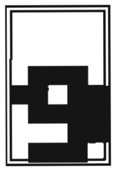
```
db 00001110b
db 00011111b
db 00010101b
db 00011101b
db 00001100b
db 00000000b
```



```
0 0 0 0 0
0 0 1 1 0
0 1 1 0 0
0 1 1 0 0
0 1 1 1 0
0 1 1 0 0
0 1 1 0 0
0 1 1 0 0
```

```
db 0 0 0 0 0 0 0 0 b
db 0 0 1 1 1 1 1 1 b
db 0 1 1 1 1 1 1 1 b
db 0 1 0 0 1 0 0 0 b
db 0 0 0 0 0 0 0 0 b
```

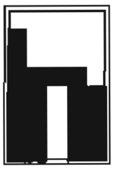
```
db 00000000b
db 00111111b
db 01111111b
db 01001000b
db 00000000b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 1 0 1 1
0 1 1 1 1
0 0 0 1 1
0 1 1 1 0
```

```
db 0 0 0 0 1 0 0 0 b
db 0 0 0 1 1 1 0 1 b
db 0 0 0 1 0 1 0 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 1 1 1 0 b
```

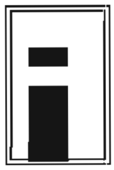
```
db 00001000b
db 00011101b
db 00010101b
db 00011111b
db 00001110b
db 00000000b
```



```
0 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 1 1 1 0
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
```

```
db 0 1 1 1 1 1 1 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 0 0 0 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 1 1 1 1 b
```

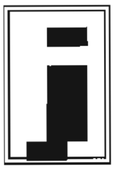
```
db 01111111b
db 00011111b
db 00010000b
db 00011111b
db 00001111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 1 1 0 0
0 0 0 0 0
0 1 1 0 0
0 1 1 0 0
0 1 1 0 0
0 1 1 0 0
```

```
db 0 0 0 0 0 0 0 0 b
db 0 0 1 0 1 1 1 1 b
db 0 0 1 0 1 1 1 1 b
db 0 0 0 0 0 0 0 0 b
db 0 0 0 0 0 0 0 0 b
```

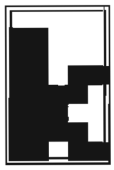
```
db 00000000b
db 00101111b
db 00101111b
db 00000000b
db 00000000b
db 00000000b
```



```
0 0 0 0 0
0 0 1 1 0
0 0 0 0 0
0 0 1 1 0
0 0 1 1 0
0 0 1 1 0
0 0 1 1 0
0 1 1 0 0
```

```
db 0 0 0 0 0 0 0 0 b
db 0 0 0 0 0 0 0 1 b
db 0 1 0 1 1 1 1 1 b
db 0 1 0 1 1 1 1 0 b
db 0 0 0 0 0 0 0 0 b
```

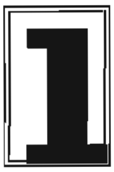
```
db 00000000b
db 00000001b
db 01011111b
db 01011110b
db 00000000b
db 00000000b
```



```
0 0 0 0 0
1 1 0 0 0
1 1 0 0 0
1 1 0 1 1
1 1 1 1 0
1 1 1 0 0
1 1 1 1 0
1 1 0 1 1
```

```
db 0 1 1 1 1 1 1 1 b
db 0 1 1 1 1 1 1 1 b
db 0 0 0 0 1 1 1 0 b
db 0 0 0 1 1 0 1 1 b
db 0 0 0 1 0 0 0 1 b
```

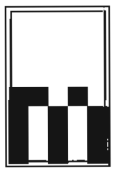
```
db 01111111b
db 01111111b
db 00001110b
db 00011011b
db 00010001b
db 00000000b
```



```
0 0 0 0 0
0 1 1 1 0
0 0 1 1 0
0 0 1 1 0
0 0 1 1 0
0 0 1 1 0
0 0 1 1 0
0 1 1 1 1
```

```
db 0 0 0 0 0 0 0 0 b
db 0 1 0 0 0 0 0 1 b
db 0 1 1 1 1 1 1 1 b
db 0 1 1 1 1 1 1 1 b
db 0 0 0 0 0 0 0 1 b
```

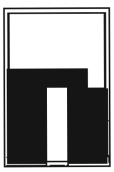
```
db 00000000b
db 01000001b
db 01111111b
db 01111111b
db 00000001b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 1 0
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
```

```
db 0 0 0 0 1 1 1 1 b
db 0 0 0 0 1 0 0 0 b
db 0 0 0 0 0 1 1 1 b
db 0 0 0 0 1 0 0 0 b
db 0 0 0 0 0 1 1 1 b
```

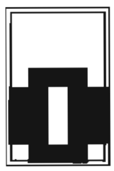
```
db 00001111b
db 00001000b
db 00000111b
db 00001000b
db 00000111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 1 1 0
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
```

```
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 0 0 0 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 1 1 1 1 b
```

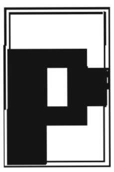
```
db 00011111b
db 00011111b
db 00010000b
db 00011111b
db 00001111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
0 1 1 1 0
```

```
db 0 0 0 0 1 1 1 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 0 0 0 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 1 1 1 0 b
```

```
db 00001110b
db 00011111b
db 00010001b
db 00011111b
db 00001110b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
1 1 1 1 0
1 1 0 1 1
1 1 0 1 1
1 1 1 1 0
1 1 0 0 0
1 1 0 0 0
```

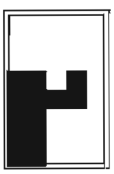
```
db 0 0 1 1 1 1 1 1 b
db 0 0 1 1 1 1 1 1 b
db 0 0 1 0 0 1 0 0 b
db 0 0 1 1 1 1 0 0 b
db 0 0 0 1 1 0 0 0 b
```

```
db 00111111b
db 00111111b
db 00100100b
db 00111100b
db 00011000b
db 00000000b
```

```
0 0 0 0 0
0 0 0 0 0
0 1 1 1 1
1 1 0 1 1
1 1 0 1 1
0 1 1 1 1
0 0 0 1 1
0 0 0 1 1
```

```
db 0 0 0 1 1 0 0 0 b
db 0 0 1 1 1 1 0 0 b
db 0 0 1 0 0 1 0 0 b
db 0 0 1 1 1 1 1 1 b
db 0 0 1 1 1 1 1 1 b
```

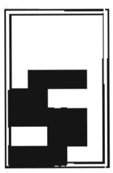
```
db 00011000b
db 00111100b
db 00100100b
db 00111111b
db 00111111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 1 0
1 1 1 1 0
1 1 0 0 0
1 1 0 0 0
1 1 0 0 0
```

```
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 1 0 0 0 b
db 0 0 0 1 1 0 0 0 b
db 0 0 0 0 0 0 0 0 b
```

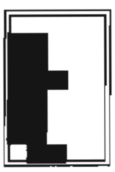
```
db 00011111b
db 00011111b
db 00001000b
db 00011000b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 1 1 0
1 1 0 0 0
1 1 1 1 0
0 0 1 1 0
1 1 1 0 0
```

```
db 0 0 0 0 1 1 0 1 b
db 0 0 0 1 1 1 0 1 b
db 0 0 0 1 0 1 1 1 b
db 0 0 0 1 0 1 1 0 b
db 0 0 0 0 0 0 0 0 b
```

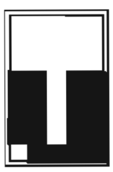
```
db 00001101b
db 00011101b
db 00010111b
db 00010110b
db 00000000b
```



```
0 0 0 0 0
1 1 0 0 0
1 1 0 0 0
1 1 1 0 0
1 1 0 0 0
1 1 0 0 0
1 1 0 0 0
0 1 1 0 0
```

```
db 0 1 1 1 1 1 1 0 b
db 0 1 1 1 1 1 1 1 b
db 0 0 0 1 0 0 0 1 b
db 0 0 0 0 0 0 0 0 b
db 0 0 0 0 0 0 0 0 b
```

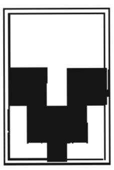
```
db 01111110b
db 01111111b
db 00010001b
db 00000000b
db 00000000b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
1 1 0 1 1
0 1 1 1 1
```

```
db 0 0 0 1 1 1 1 0 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 0 0 0 1 b
db 0 0 0 1 1 1 1 1 b
db 0 0 0 1 1 1 1 1 b
```

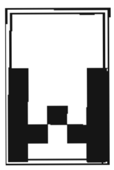
```
db 00011110b
db 00011111b
db 00000001b
db 00011111b
db 00011111b
db 00000000b
```



```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 1 1
1 1 0 1 1
0 1 1 1 0
0 1 1 1 0
0 0 1 0 0
```

```
db 0 0 0 1 1 0 0 0 b
db 0 0 0 1 1 1 1 0 b
db 0 0 0 0 0 1 1 1 b
db 0 0 0 1 1 1 1 0 b
db 0 0 0 1 1 0 0 0 b
```

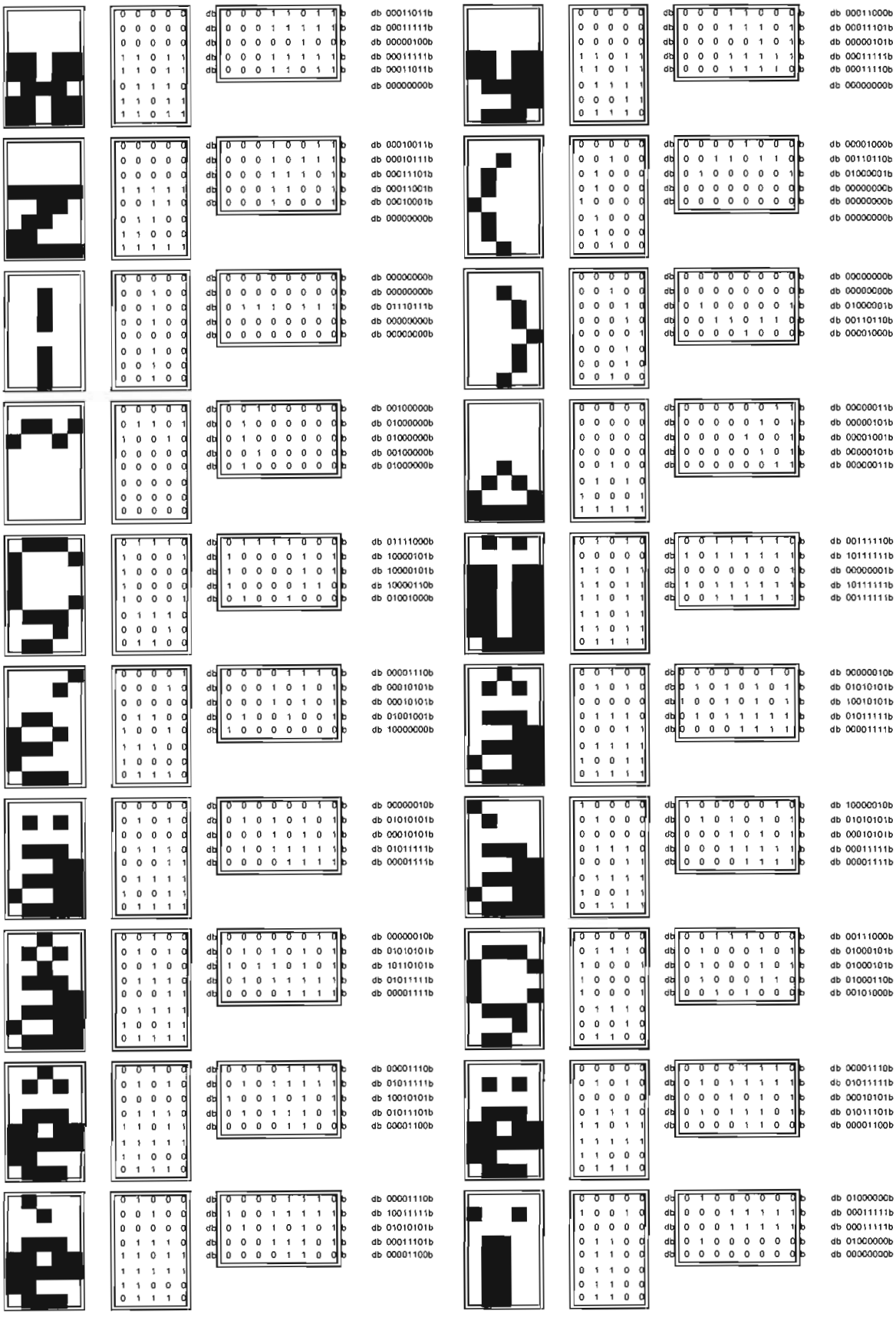
```
db 00011000b
db 00011110b
db 00000111b
db 00011110b
db 00011000b
db 00000000b
```

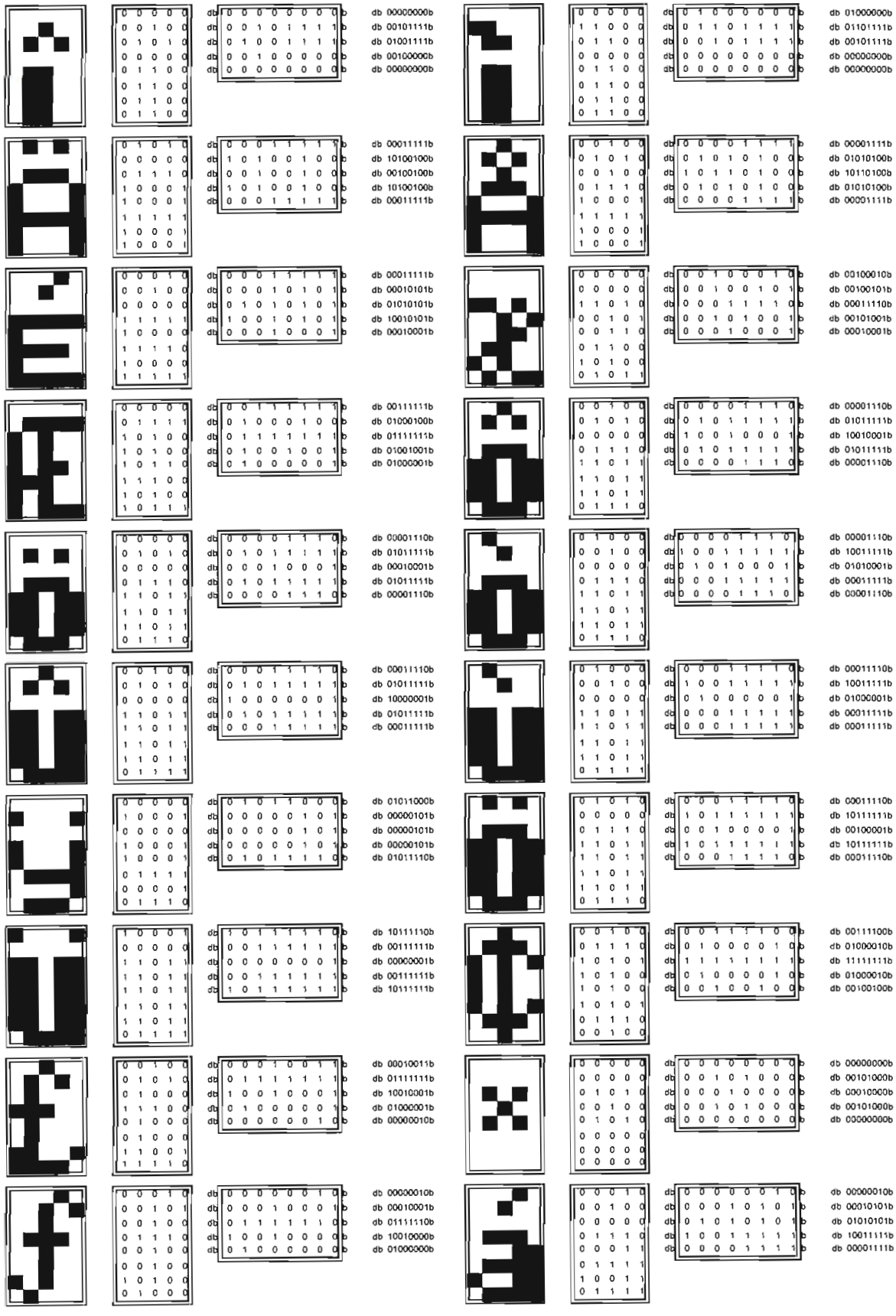


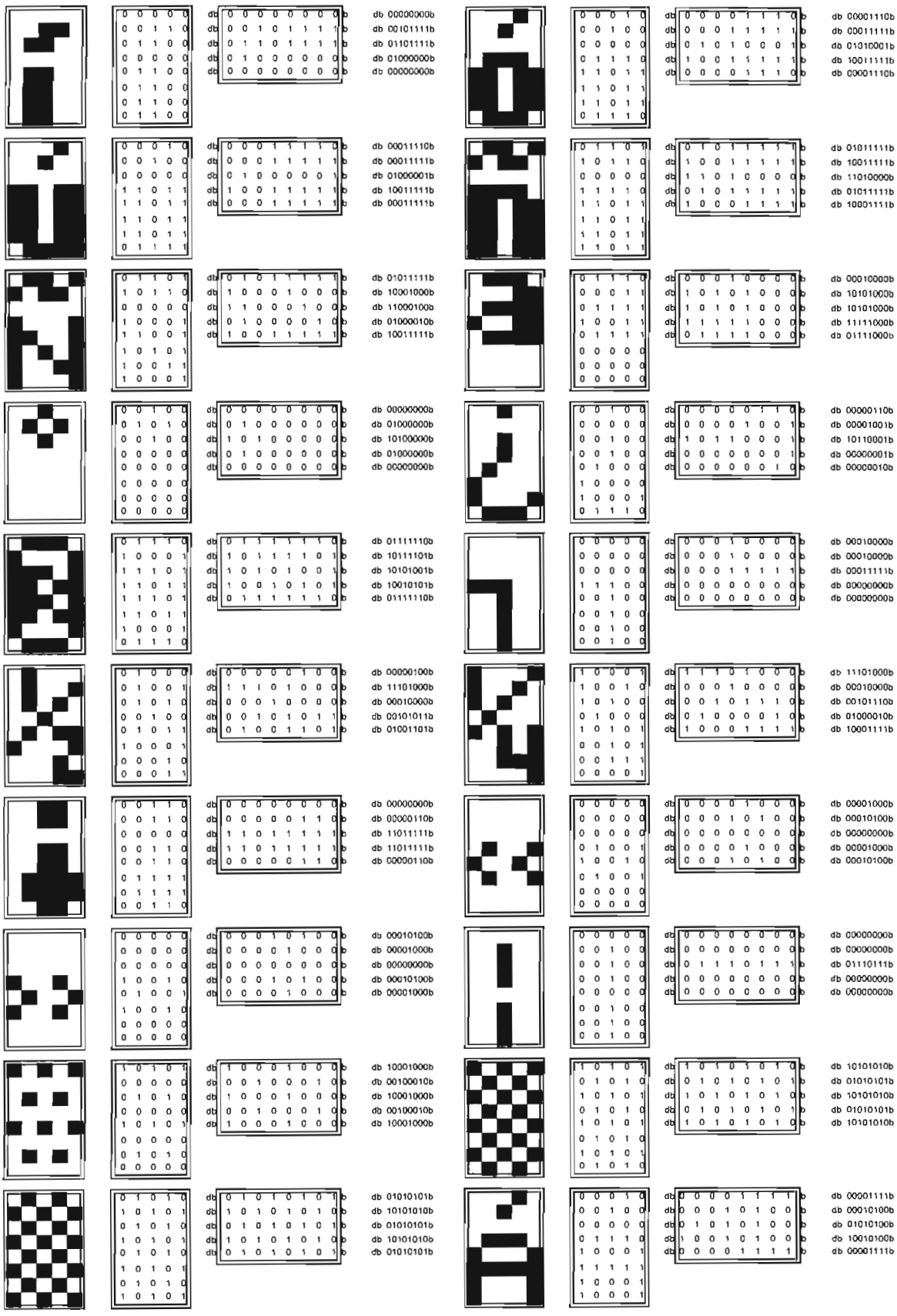
```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
1 0 0 0 1
1 0 0 0 1
1 0 1 0 1
1 1 0 1 1
1 0 0 0 1
```

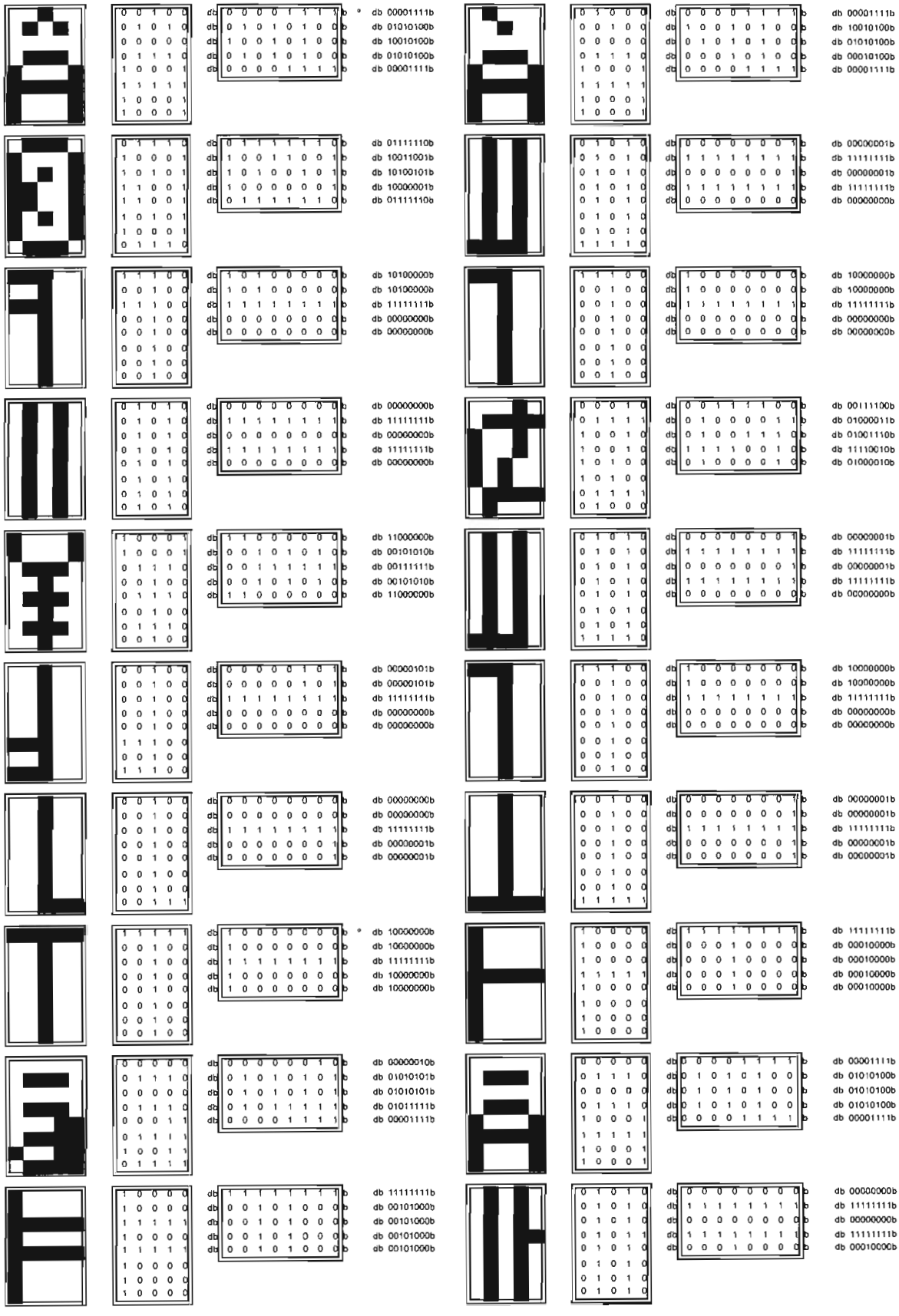
```
db 0 0 0 1 1 1 1 1 b
db 0 0 0 0 0 0 1 0 b
db 0 0 0 0 0 1 0 0 b
db 0 0 0 0 0 0 1 0 b
db 0 0 0 1 1 1 1 1 b
```

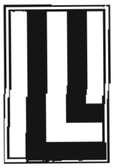
```
db 00011111b
db 00000010b
db 00000100b
db 00000010b
db 00011111b
db 00000000b
```


db 00011011b
db 00011111b
db 00000100b
db 00011111b
db 00011011b
db 00000000b
db 00011000b
db 00011101b
db 00000101b
db 00011111b
db 00011110b
db 00000000b
db 00010011b
db 00010111b
db 00011101b
db 00011001b
db 00010001b
db 00000000b
db 00001000b
db 00110110b
db 01000001b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 01110111b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 01000001b
db 00110110b
db 00001000b
db 00100000b
db 01000000b
db 01000000b
db 00100000b
db 01000000b
db 00000011b
db 00000101b
db 00001001b
db 00000101b
db 00000011b
db 01111000b
db 10000101b
db 10000101b
db 10000110b
db 01001000b
db 00111110b
db 10111111b
db 00000001b
db 10111111b
db 00111111b
db 00001110b
db 00010101b
db 00010101b
db 01001001b
db 10000000b
db 00000010b
db 01010101b
db 00010101b
db 01011111b
db 00001111b
db 00000010b
db 01010101b
db 10110101b
db 01011111b
db 00001111b
db 00001110b
db 01011111b
db 10010101b
db 01011101b
db 00001100b
db 00001110b
db 01011111b
db 00010101b
db 01011101b
db 00001100b
db 00111000b
db 01000101b
db 01000101b
db 01000110b
db 00101000b
db 00001110b
db 01011111b
db 00010101b
db 01011101b
db 00001100b
db 00001110b
db 10011111b
db 01010101b
db 00011101b
db 00001100b
db 01000000b
db 00011111b
db 00011111b
db 01000000b
db 00000000b



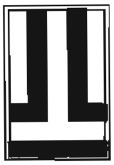

db 00000000b
db 00101111b
db 01101111b
db 01000000b
db 00000000b
db 00011110b
db 00011111b
db 01000001b
db 10011111b
db 00011111b
db 01011111b
db 10001000b
db 11000100b
db 01000010b
db 10011111b
db 00000000b
db 01000000b
db 10100000b
db 01000000b
db 00000000b
db 01111110b
db 10111101b
db 10101001b
db 10010101b
db 01111110b
db 00000100b
db 11101000b
db 00010000b
db 00101011b
db 01001101b
db 00000000b
db 00000110b
db 11011111b
db 11011111b
db 00000100b
db 00010100b
db 00001000b
db 00000000b
db 00010100b
db 00001000b
db 10001000b
db 00100010b
db 00100010b
db 10001000b
db 01010101b
db 10101010b
db 01010101b
db 10101010b
db 01010101b
db 00000110b
db 00001001b
db 10110001b
db 00000001b
db 00000010b
db 00010000b
db 00010000b
db 00011111b
db 00000000b
db 00000000b
db 11101000b
db 00010000b
db 00101110b
db 01000010b
db 10001111b
db 00001000b
db 00010100b
db 00000000b
db 00001000b
db 00010100b
db 00000000b
db 01110111b
db 00000000b
db 00000000b
db 10101010b
db 01010101b
db 10101010b
db 01010101b
db 10101010b
db 00001111b
db 00010100b
db 01010100b
db 10010100b
db 00001111b





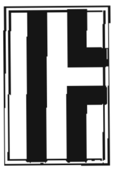
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	1
0	1	0	0	0
0	1	1	1	1

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 0 0 0 0 0 1 b    db 00000001b
db 1 1 1 1 1 1 0 1 b    db 11111101b
db 0 0 0 0 0 1 0 1 b    db 00000101b
```



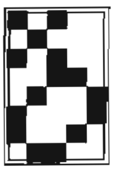
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
1	1	0	1	1
0	0	0	0	0
1	1	1	1	1

```
db 0 0 0 0 0 1 0 1 b    db 00000101b
db 1 1 1 1 1 1 0 1 b    db 11111101b
db 0 0 0 0 0 0 0 1 b    db 00000001b
db 1 1 1 1 1 1 0 1 b    db 11111101b
db 0 0 0 0 0 1 0 1 b    db 00000101b
```



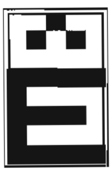
0	1	0	1	0
0	1	0	1	0
0	1	0	1	1
0	1	0	0	0
0	1	0	1	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 1 1 1 0 1 1 1 1 b    db 11101111b
db 0 0 1 0 1 0 0 0 b    db 00101000b
```



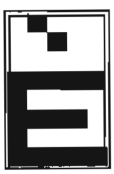
1	0	1	0	0
0	1	0	0	0
1	0	1	0	0
0	0	1	1	0
0	1	0	0	1
1	0	0	0	1
1	0	0	1	0
0	1	1	0	0

```
db 1 0 1 0 0 1 1 0 b    db 10100110b
db 0 1 0 0 1 0 0 1 b    db 01001001b
db 1 0 1 1 0 0 0 1 b    db 10110001b
db 0 0 0 1 0 0 1 0 b    db 00010010b
db 0 0 0 0 1 1 0 0 b    db 00001100b
```



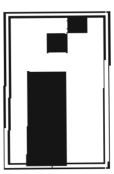
0	0	1	0	0
0	1	0	1	0
0	0	0	0	0
1	1	1	1	1
1	0	0	0	0
1	1	1	1	0
1	0	0	0	0
1	1	1	1	1

```
db 0 0 0 1 1 1 1 1 b    db 00011111b
db 0 1 0 1 0 1 0 1 b    db 01010101b
db 1 0 0 1 0 1 0 1 b    db 10010101b
db 0 1 0 1 0 1 0 1 b    db 01010101b
db 0 0 0 1 0 0 0 1 b    db 00010001b
```



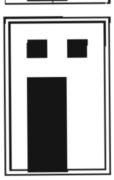
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0
1	1	1	1	1
1	0	0	0	0
1	1	1	1	0
1	0	0	0	0
1	1	1	1	1

```
db 0 0 0 1 1 1 1 1 b    db 00011111b
db 1 0 0 1 0 1 0 1 b    db 10010101b
db 0 1 0 1 0 1 0 1 b    db 01010101b
db 0 0 0 1 0 1 0 1 b    db 00010101b
db 0 0 0 1 0 0 0 1 b    db 00010001b
```



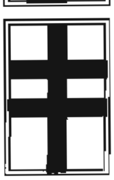
0	0	0	1	0
0	0	1	0	0
0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 0 0 1 1 1 1 1 b    db 00011111b
db 0 1 0 1 1 1 1 1 b    db 01011111b
db 1 0 0 0 0 0 0 0 b    db 10000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
```



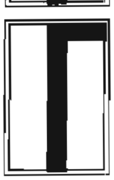
0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 1 0 1 1 1 1 1 b    db 01011111b
db 0 0 0 1 1 1 1 1 b    db 00011111b
db 0 1 0 0 0 0 0 0 b    db 01000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
```



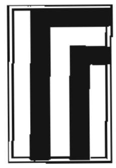
0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

```
db 0 0 1 0 1 0 0 0 b    db 00101000b
db 0 0 1 0 1 0 0 0 b    db 00101000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 1 0 1 0 0 0 b    db 00101000b
db 0 0 1 0 1 0 0 0 b    db 00101000b
```



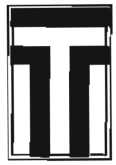
0	0	1	1	1
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 0 0 0 0 0 0 0 b    db 10000000b
db 1 0 0 0 0 0 0 0 b    db 10000000b
```



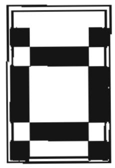
0	1	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 0 0 0 0 0 0 0 b    db 10000000b
db 1 0 1 1 1 1 1 1 b    db 10111111b
db 1 0 1 0 0 0 0 0 b    db 10100000b
```



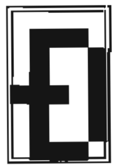
1	1	1	1	1
0	0	0	0	0
1	1	0	1	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

```
db 1 0 1 0 0 0 0 0 b    db 10100000b
db 1 0 1 1 1 1 1 1 b    db 10111111b
db 1 0 0 0 0 0 0 0 b    db 10000000b
db 1 0 1 1 1 1 1 1 b    db 10111111b
db 1 0 1 0 0 0 0 0 b    db 10100000b
```



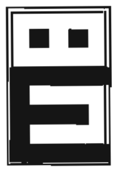
0	0	0	0	0
1	0	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0
1	0	0	0	1

```
db 0 1 0 1 1 1 0 1 b    db 01011101b
db 0 0 1 0 0 0 1 0 b    db 00100010b
db 0 0 1 0 0 0 1 0 b    db 00100010b
db 0 0 1 0 0 0 1 0 b    db 00100010b
db 0 1 0 1 1 1 0 1 b    db 01011101b
```



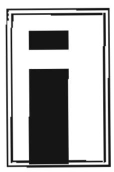
0	0	0	0	0
0	1	1	1	0
0	1	0	0	1
0	1	0	0	1
1	1	1	0	1
0	1	0	0	1
0	1	0	0	1
0	1	1	1	0

```
db 0 0 0 0 1 0 0 0 b    db 00001000b
db 0 1 1 1 1 1 1 1 b    db 01111111b
db 0 1 0 0 1 0 0 1 b    db 01001001b
db 0 1 0 0 0 0 0 1 b    db 01000001b
db 0 0 1 1 1 1 1 0 b    db 00111110b
```



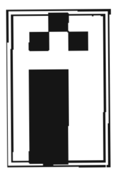
0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
1	1	1	1	1
1	0	0	0	0
1	1	1	1	0
1	0	0	0	0
1	1	1	1	1

```
db 0 0 0 1 1 1 1 1 b    db 00011111b
db 0 1 0 1 0 1 0 1 b    db 01010101b
db 0 0 0 1 0 1 0 1 b    db 00010101b
db 0 1 0 1 0 1 0 1 b    db 01010101b
db 0 0 0 1 0 0 0 1 b    db 00010001b
```



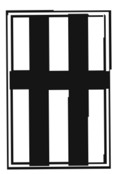
0	0	0	0	0
0	1	1	0	0
0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 1 0 1 1 1 1 1 b    db 01011111b
db 0 1 0 1 1 1 1 1 b    db 01011111b
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
```



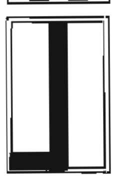
0	0	1	0	0
0	1	0	1	0
0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

```
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 1 0 1 1 1 1 1 b    db 01011111b
db 1 0 0 1 1 1 1 1 b    db 10011111b
db 0 1 0 0 0 0 0 0 b    db 01000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
```



0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
1	1	1	1	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

```
db 0 0 0 1 0 0 0 0 b    db 00010000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 0 1 0 0 0 0 b    db 00010000b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 0 1 0 0 0 0 b    db 00010000b
```

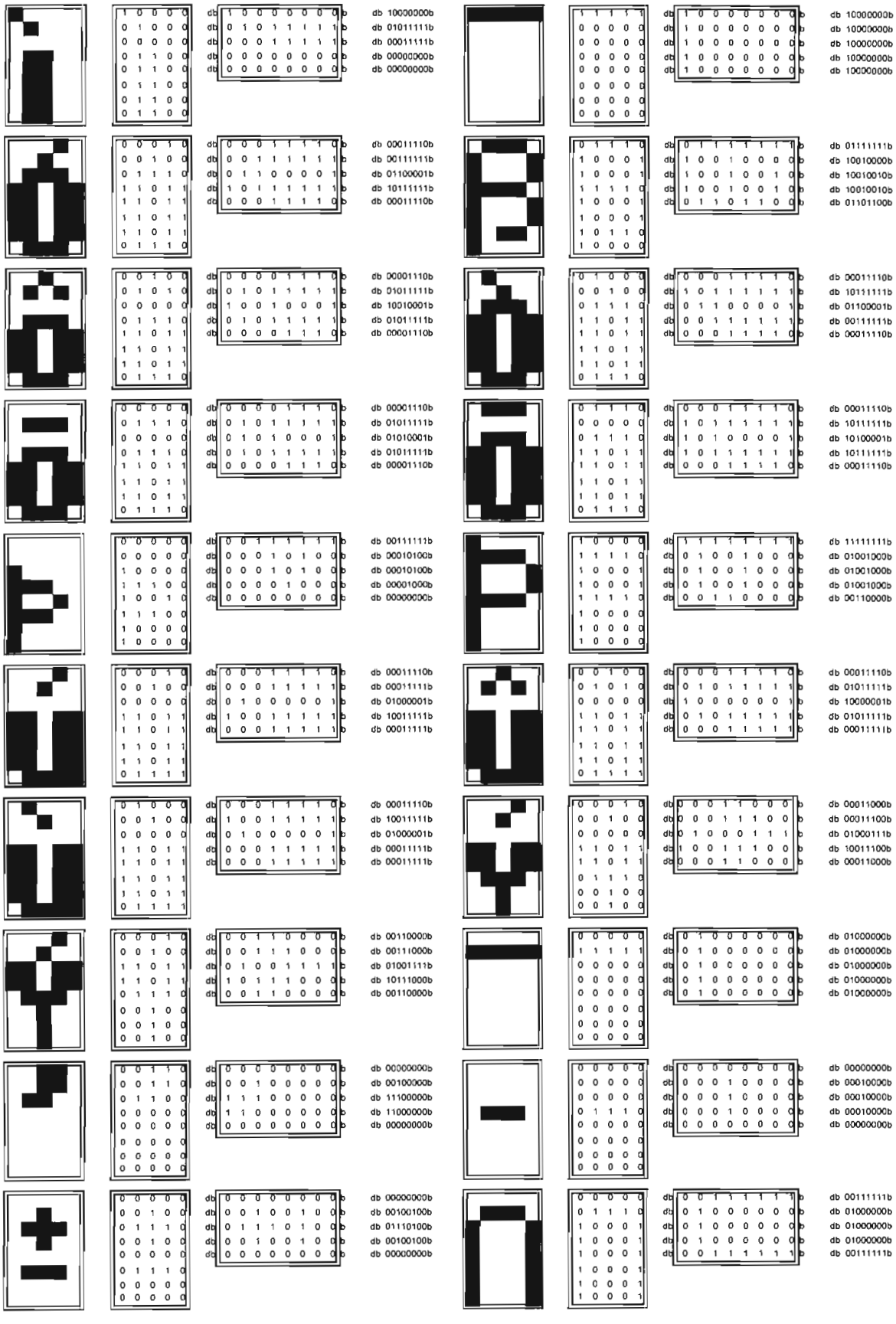


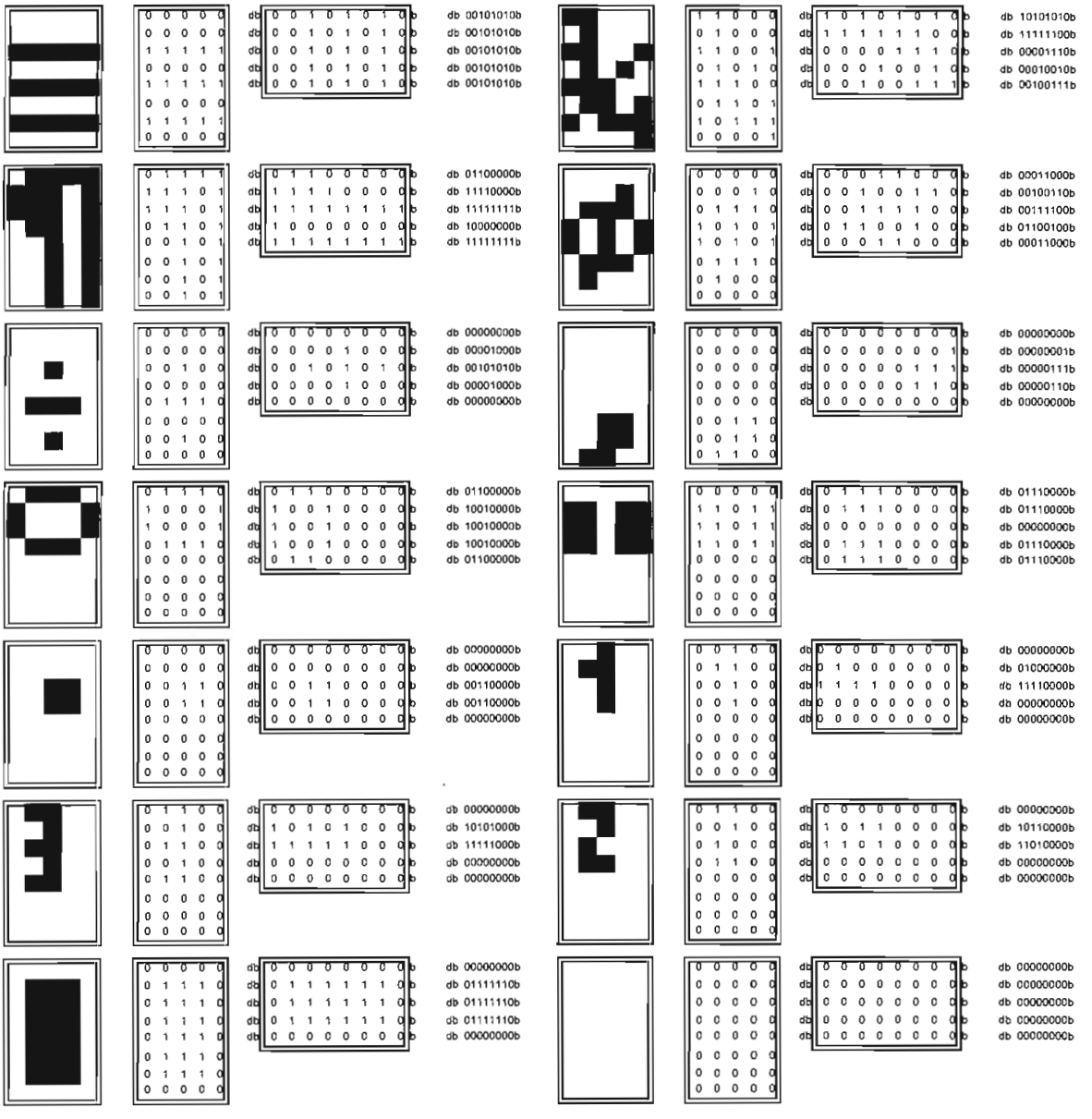
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
1	1	1	0	0

```
db 0 0 0 0 0 0 0 1 b    db 00000001b
db 0 0 0 0 0 0 0 1 b    db 00000001b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 0 0 0 0 0 0 0 0 b    db 00000000b
db 0 0 0 0 0 0 0 0 b    db 00000000b
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

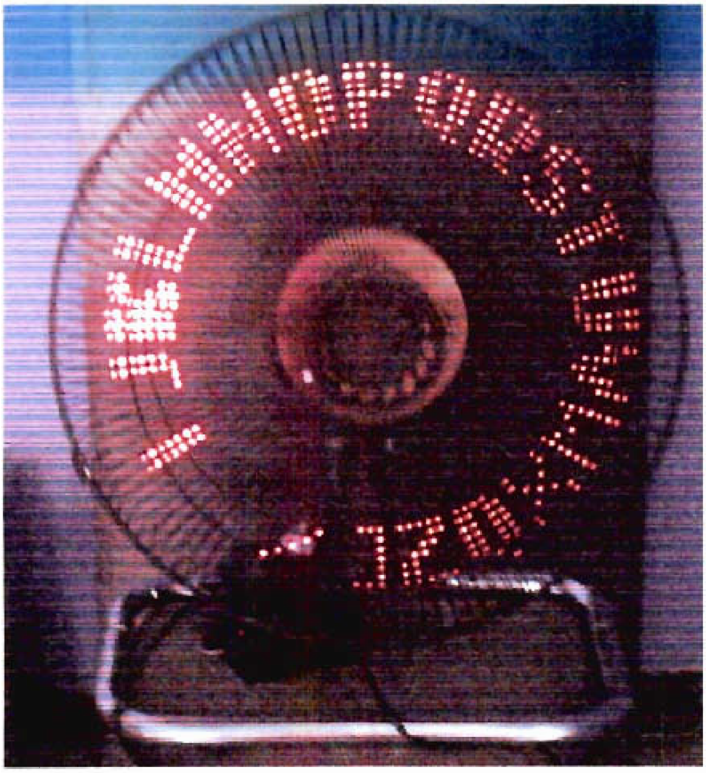
```
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 1 1 1 1 1 1 1 b    db 11111111b
db 1 1 1 1 1 1 1 1 b    db 11111111b
```

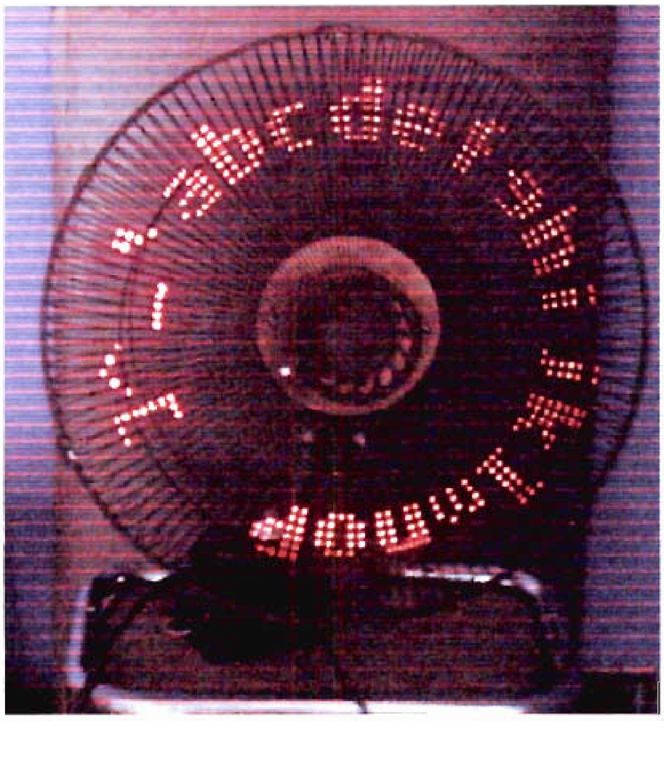


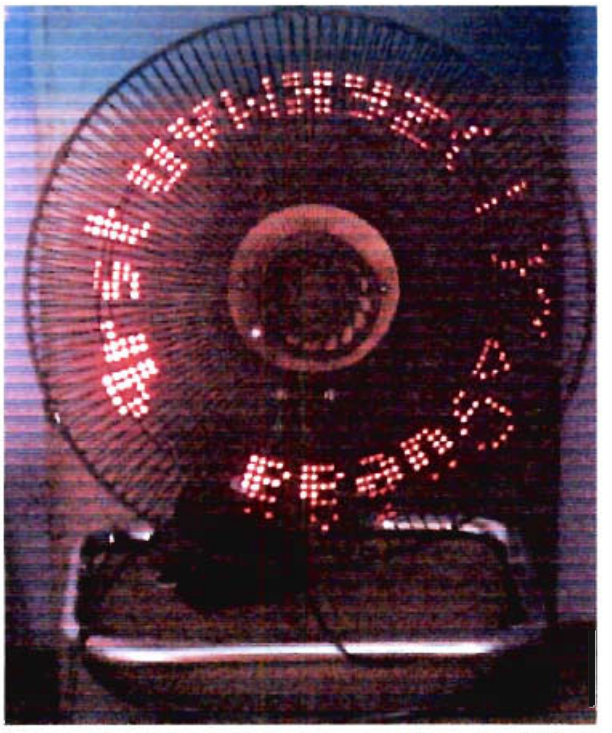

db 00101010b
db 00101010b
db 00101010b
db 00101010b
db 00101010b
db 10101010b
db 11111000b
db 00001110b
db 00010010b
db 00100111b
db 01100000b
db 11110000b
db 11111111b
db 10000000b
db 11111111b
db 00011000b
db 00100110b
db 00111100b
db 01100100b
db 00011000b
db 00000000b
db 00001000b
db 00101010b
db 00001000b
db 00000000b
db 00000000b
db 00000001b
db 00000111b
db 00000110b
db 00000000b
db 01100000b
db 10010000b
db 10010000b
db 10010000b
db 01100000b
db 01110000b
db 01110000b
db 00000000b
db 01110000b
db 01110000b
db 00000000b
db 00000000b
db 00110000b
db 00110000b
db 00000000b
db 00000000b
db 01000000b
db 11110000b
db 00000000b
db 00000000b
db 00000000b
db 10101000b
db 11111000b
db 00000000b
db 00000000b
db 10110000b
db 11010000b
db 00000000b
db 00000000b
db 00000000b
db 01111110b
db 01111110b
db 01111110b
db 00000000b
db 00000000b
db 00000000b
db 00000000b
db 00000000b

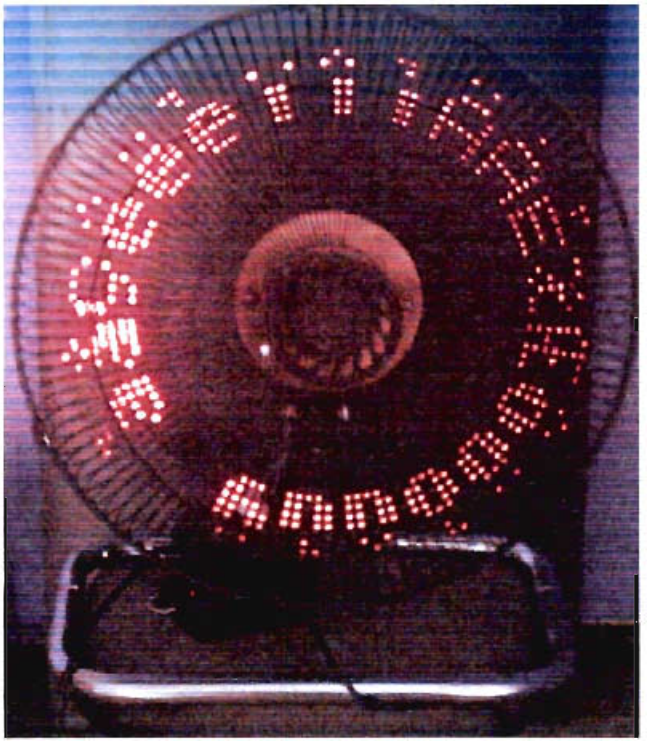
ANEXO C

CARACTERES ASCII MOSTRADOS EN EL GENERADOR DE CARACTERES A PARTIR DE UNA FILA DE LEDs GIRATORIA






Read


Amazon