

**UNIVERSIDAD AUTÓNOMA DE QUERÉTARO**

**CAMPUS SAN JUAN DEL RÍO**

**INGENIERÍA ELECTROMECÁNICA**

**DISEÑO E IMPLEMENTACIÓN DE INTERFAZ PARA EL  
CONTROL DE UNA FUENTE COREOGRÁFICA**

**Tesis**

**Como parte de los requisitos para obtener el título de  
Ingeniero Electromecánico**

**Presenta**

**Ernesto Carlos Capuano Zúñiga**

**San Juan del Río, Qro. a 7 de Agosto de 2009**

H73670

TS

621.39

C255d

# RESUMEN

La finalidad de esta investigación es presentar el desarrollo del control y el correspondiente software para una fuente hidráulica musical que sea eficiente, de buena calidad y con un relativo bajo costo. Para el desarrollo final del sistema se han utilizado herramientas e investigaciones sobre software, electrónica digital y mecánica de fluidos, sin embargo, el presente trabajo se enfoca específicamente al desarrollo del control para de la fuente, es decir, el desarrollo del software y el diseño e implementación del hardware. El desarrollo realizado incluye la elaboración de un software informático para comandar independientemente, acorde a una pista musical, un conjunto de 24 electroválvulas y 24 luces subacuáticas que actúan de manera síncrona con una pista musical. Esto se logra por medio de un conjunto de filtros pasa-bandas implementados en el software dedicado al procesamiento de audio, tales filtros están basados en herramientas matemáticas específicas de análisis de señales. La interfaz gráfica que se desarrolló es amigable, y permite una fácil configuración para la activación de las válvulas que liberan el chorro de agua, traduciéndose esto, en que es muy accesible al técnico durante la configuración y así mismo también para el usuario final durante la puesta en marcha y selección de pista musical. Una tarjeta electrónica controla independientemente cada una de las electroválvulas así como cada una de la luces de la fuente. La activación se realiza por medio de pulsos enviados por el programa. Estos pulsos se obtienen al sobrepasar un nivel preestablecido de la densidad del contenido de frecuencias de la melodía en turno. Los niveles son preestablecidos por el usuario con un previo ajuste a la reproducción. Se utilizó un algoritmo eficiente de la FFT (*Fast Fourier Transform* – Transformada Rápida de Fourier) para la obtención del contenido de frecuencias. Este algoritmo consume menor cantidad de recursos de cómputo aun cuando este tipo de operaciones son demasiado demandantes para cualquier computadora. La fase de desarrollo de hardware se dividió en dos etapas, la etapa de control y la de potencia. La etapa de control recibe los pulsos generados por el software y la etapa de control se encarga de activarlas y gestionar la activación de las válvulas.

# AGRADECIMIENTOS

Agradezco inmensamente a esas dos personas esenciales en mi vida, a mi madre y a mi hermano que sin las cuales no hubiera logrado ser lo que soy ahora y quienes han confiado ciegamente en mi voluntad para realizar esta investigación e incondicionalmente me han brindado todo su apoyo y amor. Gracias por su incansable amor.

A Mara, quien considero como mi segunda madre y quien me apoyó incondicionalmente en todos los pasos que he tomado durante mi carrera y mi vida personal.

A mi hermana Carolina quien me ha dado muchas buenas razones para ser mejor en mi vida y ha llenado de esperanza y paciencia mi trayectoria como ser humano.

A mi mejor amigo Fernando que a través de su gran amistad me ha enseñado que en la vida siempre hay momentos para estar feliz.

A mis sinodales y a mi director de tesis quienes me ayudaron a dirigir esta investigación por un camino ordenado y funcional.

# ÍNDICE

1- INTRODUCCIÓN .....	10
1.1 Descripción del problema .....	12
1.2 Objetivo General.....	13
1.2.1 Objetivos Particulares .....	13
1.3 Justificación e Hipótesis.....	14
1.4 Planteamiento general.....	16
1.4.1 El sistema en General.....	17
1.4.2 Componente Software del sistema.....	19
1.4.3 Componente Hardware del sistema .....	20
1.5 Antecedentes.....	21
1.5.1 Fuentes musicales y su evolución.....	22
2- REVISIÓN DE LITERATURA.....	26
2.1 Fundamentos teóricos .....	26
2.1.1 Naturaleza del Sonido .....	26
2.1.2 Sonido Digital .....	28
2.1.3 Análisis de señales Tiempo-Frecuencia.....	30
2.2 Procesamiento digital de señales .....	33
2.2.1 Muestreo digital.....	34
2.3 Justificación de la digitalización .....	39
2.3.1 Ventajas .....	40
2.3.2 Inconvenientes.....	41
2.4 Algoritmos de Procesamiento .....	41
2.4.1 Transformada Discreta de Fourier.....	42
2.4.2 Transformada Rápida de Fourier .....	43
2.4.3 El puerto Paralelo y su historia.....	44
3- DISEÑO E IMPLEMENTACIÓN DE HARDWARE.....	48
3.1 Características del puerto de comunicación ( <i>Parallel Port</i> ) .....	48
3.2 Tarjeta de Control .....	51
3.3 Tarjeta de Potencia.....	56
3.4 Diseño con PROTEL DXP .....	58

# ÍNDICE DE TABLAS

<b>Tabla</b>	<b>Pág.</b>
Tabla 1.1 Lista comparativa de precios para fuentes coreográficas .....	14
Tabla 2.1 Tipos de puertos y sus velocidades .....	46
Tabla 3.1 Tiempos de propagación por latch .....	53
Tabla 3.2 Características de la electro-válvula.....	59
Tabla 4.1 Funciones utilizadas para la captura de audio .....	68
Tabla 4.2 Algoritmo bit reverse .....	70
Tabla 4.3 Transformada de Fourier por métodos numéricos .....	71
Tabla 4.4 Metodología para el manejo de archivos WAV con API de Windows .....	78

Figura 4.3 Metodología para el desarrollo del software de control.....	66
Figura 4.4 Diagrama a bloques para la etapa de captura de audio.....	68
Figura 4.5 Etapa de Procesamiento.....	70
Figura 4.6 Diagrama a bloques de utilización de IO.DLL.....	73
Figura 4.7 Metodología de activación de válvulas y luces.....	74
Figura 4.8 Diagrama funcional del simulador de válvulas y luces.....	76
Figura 4.9 Metodología para la simulación.....	77
Figura 4.10 Metodología de lectura de panel de entrada del usuario final.....	80
Figura 5.1 Imagen de software capturando pista musical.....	83
Figura 5.2 Diagrama básico de plan de pruebas.....	84
Figura 5.3 Tubo de pruebas y boquillas para el plan de pruebas preliminar.....	85
Figura 5.4 Software capturando con saturación de frecuencias.....	86
Figura 5.5 Análisis de pista musical de género "Música Clásica".....	87
Figura 5.6 Análisis de pista musical de género "Rock Pesado – Heavy Metal".....	88
Figura 8.1 Explicación de funciones de interfaz - Software.....	96

## 1 - INTRODUCCIÓN

Hoy en día el uso de las fuentes de agua coreográficas (fuentes que emiten chorros de agua acordes a los tonos de una pista musical) es sinónimo de elegancia y de gran atractivo visual del lugar en el que se instalan. Tales lugares son principalmente hoteles y centros expositores de gran importancia. Sin embargo, su elevado costo y complejo diseño hacen que sean limitadas tanto en su construcción como en su implementación y mantenimiento. Uno de los objetivos de esta investigación, es el de involucrarse en todos los aspectos relacionados con las fuentes musicales coreográficas y así, desarrollar dos de los componentes más importantes de una fuente musical (software y hardware). Se debe cumplir con las exigencias del público en general, tratando de expedir porte y fineza con un toque de atracción para lugar que las hospeda. Todo lo anterior sin olvidar también los aspectos de seguridad para el usuario o espectador. Esto suena fácil de realizar, pero se necesitan realizar trabajos especializados de ingeniería de gran calidad y eficiencia para así llevar a cabo las pruebas necesarias e implementaciones correctas.

Cabe mencionar que, en la actualidad, son muchas las instituciones que desean este tipo de fuentes, sin embargo, son utilizadas únicamente en lugares específicos de gran importancia por el elevado costo tanto de diseño como de implementación.

En el presente trabajo se muestra información empleada, metodología, resultados y aportaciones del desarrollo de un proyecto de esta naturaleza, así, en el Capítulo 1 se mencionan las razones por las cuales surgió esta investigación así como los argumentos y justificación del porque es factible su realización. Se presentan también los logros a alcanzar y un esquema rápido de toda la estructuración de un sistema completo, de tal manera que, se puedan analizar claramente todos los componentes del proyecto



En el Capítulo 2, se mencionan los fundamentos teóricos necesarios para entender esta investigación. Entre los conceptos que abarca este tema están: Conceptos Básicos de Sonido, Sonido Digital o Digitalización de Audio, Análisis de Señales (Audio), procesamiento, Teoremas de Muestreo y fundamentos matemáticos específicos. Se menciona también los antecedentes históricos de las fuentes coreográficas en los que se detallan los procesos de construcción, armado y utilización. También se incluye la manera de cómo se utilizaban en el pasado, la evolución que ha sufrido con el paso de los años y los alcances que tienen hasta fechas recientes.

En el Capítulo 3 se detalla de manera específica la construcción de las tarjetas electrónicas, tanto para el control de activación de las electroválvulas como para el manejo de potencia. Se detalla el diseño, la construcción e implementación de cada elemento con el objetivo de entender plenamente el procedimiento. Durante el desarrollo se mencionan los paquetes y herramientas que se utilizaron para el diseño y creación de estos elementos de hardware. Además, se señalan los protocolos y reglas de comunicación entre la tarjeta de control y la PC (*Personal Computer* - Computadora Personal). Por otra parte, se hace referencia al diseño de un panel interactivo que permite el cambio de pista musical. Este panel, está orientado únicamente para el usuario final ya que no requiere de conocimientos de software especializados. Es importante comentar que dicho panel se ubicará en la parte exterior del tablero de control de la fuente, por lo que estará expuesto a uso rudo y humedad, por esta razón, se justifica una construcción con componentes no comerciales utilizados durante su diseño.

En el Capítulo 4 se detallan los pasos para el desarrollo del software de procesamiento y control de la fuente. Su característica principal es que tiene la capacidad de analizar las señales de audio analógicas y procesarlas para obtener la secuencia de activación de las electroválvulas. La señal de audio analizada se adquiere de la línea de entrada de la tarjeta de audio y se le aplica un muestreo de

44KHz en la modalidad estéreo. De esta forma, se genera continuamente una rutina de activación de las electroválvulas conectadas a la tarjeta de potencia y controladas por medio de la tarjeta de control. Se detalla también la operación de cada una de las funciones implementadas, tanto para el filtrado digital del audio como para la generación de las rutinas de activación de las electroválvulas. Y finalmente, se aborda el tema de los controladores para la comunicación entre la tarjeta de control con el puerto de comunicación (puerto paralelo), así como su estructura y metodología de uso.

El Capítulo 5 presenta los resultados sobre pruebas preliminares del sistema desarrollado así como la integración del software con el hardware. Se incluye la forma en que fueron acoplados uno con el otro, las condiciones de prueba de la interfaz y un análisis de resultados físicos obtenidos.

Finalmente, en el Capítulo 6 se exponen las conclusiones que se desprenden de esta investigación así como las perspectivas y recomendaciones, es decir, los logros obtenidos y las futuras modificaciones que pudieran hacerse tanto en software como hardware o incluso como sistema completo.

## **1.1 Descripción del problema**

En la actualidad no se tiene conocimiento de empresas nacionales que desarrollen fuentes coreográficas cuyo costo sea accesible para potenciales compradores, ya sean públicos o privados. Durante el desarrollo del parque Río-Arronte ubicado en la ciudad de San Juan del Río, se tuvo la visión de implementar una atracción de esta naturaleza y estratégicamente, se realizó un convenio entre la Universidad Autónoma de Querétaro (UAQ) y el municipio en turno para desarrollar e implementar una fuente de agua coreografiada con música. Esta fuente sería instalada en dicho parque, ya que la Fundación Río-Arronte se acercó a la UAQ para gestionar su diseño e instalación a través de la

Facultad de Ingeniería. De ésta forma, se ha pretendido desarrollar e innovar el diseño de una fuente musical coreográfica cuyo enfoque principal es que resulte atractiva y relativamente económica. Pensando también en una posible comercialización y expansión.

## 1.2 Objetivo General

Crear una interfaz para controlar una fuente musical coreográfica de fácil uso y que cumpla con las demandas básicas de seguridad, economía y calidad requeridas, todo ello basado en herramientas matemáticas, algoritmos de programación, y herramientas de diseño electrónico.

### 1.2.1 Objetivos Particulares

- Desarrollar una interfaz amigable en software que permita configurar la coreografía de la fuente en función de cada pista musical. La interfaz será desarrollada empleando Visual Basic.
- Programar algoritmos de análisis de señales para que, en función de la pista de audio seleccionada por el usuario, el software genere un patrón de pulsos para el control de las electroválvulas.
- Diseñar e implementar dos tarjetas electrónicas con apoyo de un software de diseño electrónico (Protel DXP). Este software de diseño servirá para crear las pistas eléctricas en las tarjetas electrónicas así como también para calcular el ancho de pista necesario para la tarjeta de potencia, ya que esta última gestiona una gran cantidad de corriente. Una tarjeta es para el control de las electroválvulas y de las luces rítmicas; la otra es de potencia que finalmente energizará dichos elementos.

- Integrar y realizar pruebas preliminares para verificar y poner en marcha el sistema desarrollado.

### 1.3 Justificación e Hipótesis

**Justificación.-** Debido a los elevados costos de desarrollo, instalación y mantenimiento de las fuentes musicales coreográficas (Tabla 1.1 Lista comparativa de precios para fuentes coreográficas), es importante diseñar una fuente que reúna de manera paralela el aspecto costo-calidad.

**Tabla 1.1 Lista comparativa de precios para fuentes coreográficas**

Compañía	Precio
Fuente Danzante Bernal	13,000,000 MX
Atlantic Fountains <sup>(a)</sup>	500,000 dls
Magical Sentosa	4,000,000 dls
Grand Heaven	500,000 dls
Exclusive Fountains <sup>(b)</sup>	700,000 dls

Una manera de lograr bajos costos y cumplir con los requerimientos mínimos de calidad es basar el diseño en buenas metodologías de procesamiento digital eficiente a fin de evitar el uso de una PC (*Personal-Computer*, Computadora Personal) de alto rendimiento y por ende de alto costo. Es importante también un buen diseño de las interfaces electrónicas entre la fuente coreográfica y la PC encargada del procesamiento. Para reducir costos es necesario también utilizar elementos electrónicos estrictamente seleccionados que no sacrifiquen calidad y que eviten costos posteriores por mantenimiento correctivo. La mayoría de las fuentes musicales comerciales contienen grupos de luces que solo pueden ser activados como grupo y no de forma individual. En este proyecto se emplea una tarjeta con 48 salidas de potencia cuya característica principal es que se puedan

activar de forma individual, vía software y acorde un filtrado digital. Del total de las 48 salidas, 24 tienen el objetivo de activar las electroválvulas y las restantes de activar luminarias subacuáticas. Es sumamente importante que la activación o emisión de pulsos se realice por separado, tanto en cada electroválvula como en cada luminaria, y así poder crear rutinas individuales para lograr un mejor efecto visual.

Por otra parte, en fuentes comerciales, la interfaz gráfica para la selección de las pistas por lo general es muy compleja de usar. Suele ser utilizada por un operario especializado, ya que los efectos o secuencias deben ser pre-programadas con anterioridad y con una inversión de tiempo excesivo. Esta programación se hace en intervalos de forma manual de tal manera que se abarque toda la pista musical. Lo anterior representa una desventaja para el sistema ya que se requiere demasiado tiempo para lograr realizar la coreografía y resulta un costo de operación excesivo. En la Figura 1.1 se muestra un software para la coreografía de las pistas musicales de la compañía **Musical Fountains**. Es evidente que el manejo de esta interfaz es complicado y que requiere de personal calificado y con conocimientos muy específicos del sistema para poder operarlo.

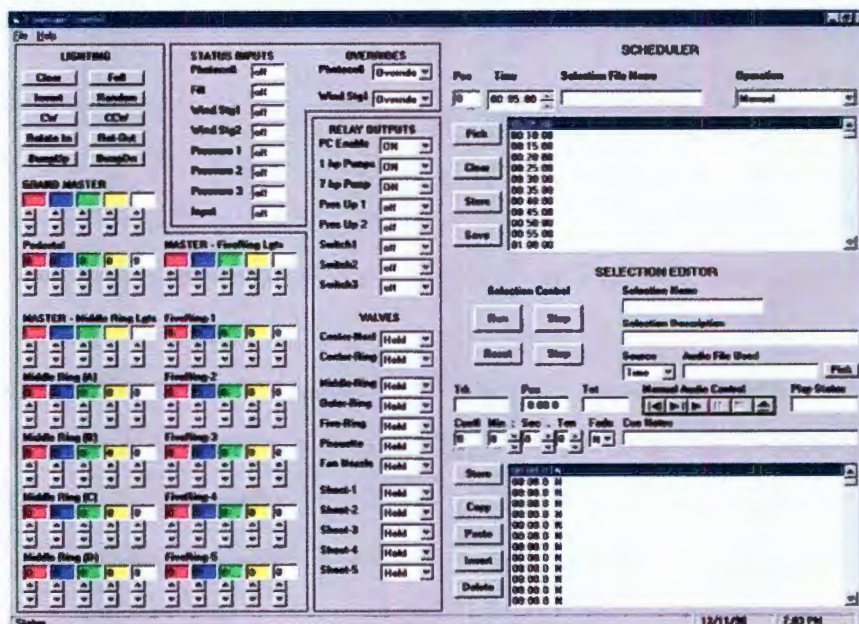


Figura 1.1 Software comercial para la coreografía de pistas musicales

En consideración de lo antes expresado, se pretende diseñar una interfaz amigable y accesible a cualquier usuario técnico con conocimientos básicos de computación. El usuario técnico que se encargará de configurar los parámetros primarios del sistema, solo tendrá que conocer lo básico en sistemas computacionales así como la estructura y ubicación de las electroválvulas (Conocimientos que también son estrictamente necesarios para los usuarios técnicos de las fuentes comerciales). El usuario técnico a cargo podrá seleccionar, reproducir y editar una coreografía para cada pista musical de manera fácil y efectiva, esto gracias a los algoritmos internos de análisis digital y a una amigable ubicación de los controles del software interfaz.

**Hipótesis.-** La fuente trabajará de manera correcta siempre y cuando contenga ritmos con tiempos moderados que no sobrepasen la frecuencia máxima de disparo de las válvulas (*Rock pesado, Heavy Metal* y estilos parecidos), el software debe ser configurado para seguir un patrón de activación una única vez, pero si es necesario, se podrá cargar una configuración diferente antes de reproducir un tipo de género diferente. El software tiene la capacidad de guardar varias configuraciones para poder ser cargadas después y reducir tiempo de configuración. El usuario técnico deberá configurar el patrón de activaciones acorde al género en turno. Por otro lado, el software tiene la capacidad de gestionar la activación de 24 válvulas y 24 luces a la vez, si así lo desea el usuario, pero deberá considerar una posible saturación en el sistema hidráulico al activar todas las válvulas instantáneamente.

#### 1.4 Planteamiento general

Por los alcances del objetivo principal de este proyecto, es importante definir el proceso de desarrollo de la interfaz así como sus propiedades, componentes, ventajas y desventajas de una aplicación específica de este tipo.

En el diseño de la interfaz, es importante conocer sus componentes como la comunicación, interacción y accesibilidad entre otros.

**Comunicación:** Define la interacción entre el usuario y la interfaz visual así como la que se establece entre el controlador y objeto a controlar. Esta comunicación debe ser clara y veloz, permitiendo una configuración rápida, así como cualquier modificación en general que se quiera llevar a cabo para poner en funcionamiento la fuente.

**Interacción:** Esta relación entre el usuario y la fuente a través de la interfaz, debe ser totalmente *eficaz* y *accesible* aún cuando la interfaz contiene parámetros no comunes que se deben ajustar, por ende, debe considerarse de forma tal que permita un diseño de interfaz que facilite el manejo de la misma.

#### 1.4.1 El sistema en General

La Figura 1.2 muestra de manera generalizada un sistema completo en el cual interviene la interfaz a desarrollar de esta investigación.

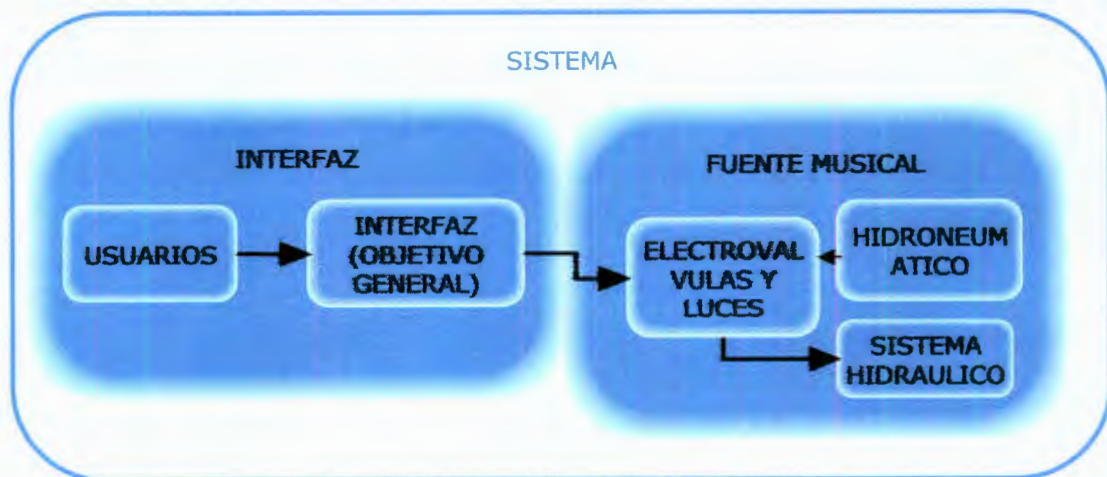


Figura 1.2 Diagrama general del sistema

Es importante notar que el bloque usuarios esta en plural, esto debido a que habrá más de un usuario interactuando con la fuente, cada uno de estos con un nivel de conocimientos diferente. Habrá específicamente dos tipos usuarios que manejarán la interfaz: el Técnico y el *Usuario Final* (Figura 1.3). El usuario *técnico* tendrá conocimientos sobresalientes pero no especiales sobre el manejo del software y hardware ya que se encargará de configurar el sistema de manera permanente. Esta pre-configuración realizada por el técnico dejará funcional la interfaz para que el *usuario final* sea quien la utilice configurando parámetros básicos como la reproducción, detención y adelanto u atraso de la pista musical. Existen puntos relevantes sobre los tipos de usuarios para este proyecto de los cuales se tendrá más detalle en los siguientes capítulos.

A diferencia de otras fuentes comerciales en las cuales se requiere de conocimientos especiales tanto de software como de hardware e incluso de hidráulica, esta fuente a desarrollar solo requiere de un usuario técnico con conocimientos básicos de software y conocimientos sobre la ubicación de las válvulas para poder configurarlas y facilitarle el trabajo al usuario final.

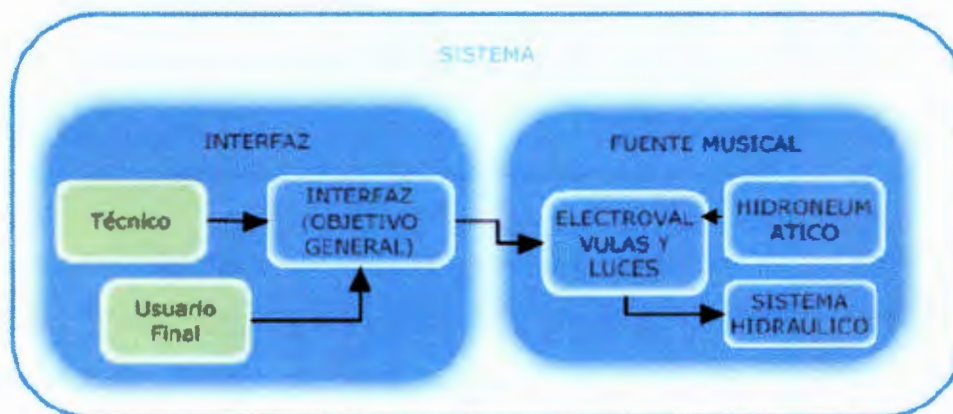


Figura 1.3 Tipos de usuarios para la interfaz



### 1.4.2 Componente Software del sistema

En la Figura 1.4 se muestra a detalle el bloque interfaz en cual intervienen dos componentes clave: el Software y el Hardware. El software será utilizado para capturar el audio de una tarjeta de sonido ubicada en la PC donde residirá este paquete de control; y al decir que capturará audio tenemos que pensar en señales análogas que serán transformadas a señales digitales, que se procesaran y discriminaran en función de la configuración definida por el usuario técnico. Una vez realizado lo anterior, estas señales discriminadas serán enviadas a una tarjeta de control que se encargarán de activar las electroválvulas instaladas en la fuente.

El Técnico configurará los niveles de densidad y de frecuencia, los cuales a su vez, activarán las electroválvulas; el Usuario Final solo indicará que pista musical será reproducida y analizada en línea (Figura 1.5).

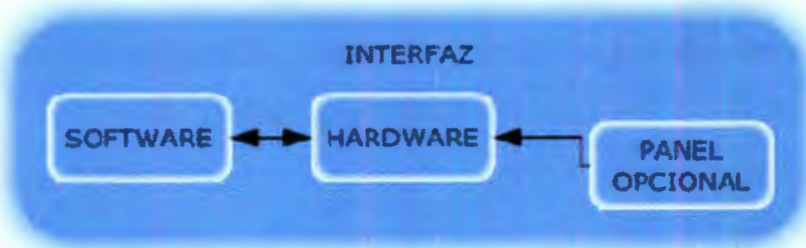


Figura 1.4 Diagrama detallado del bloque interfaz

La Figura 1.5 muestra un esquema completo del Software a desarrollar, el primer bloque es la captura de audio que se llevará a cabo específicamente por medio de funciones multimedia de Windows, teniendo capturada esta señal de audio la convertiremos a una señal digital para poder manipularla en la computadora. Después, debido a que esta señal digital está aún en el dominio del tiempo, se le aplicará la Transformada de Fourier para obtener su contenido frecuencial con el objetivo de separar las distintas frecuencias contenidas en las pista musical y restringirlas para el uso exclusivo de una válvula o luminaria específica del sistema.

Recuerde también que los parámetros sobre los cuales se establecerán los pulsos, que a su vez activarán las electroválvulas, serán los sobrepasos de los niveles de densidad de las frecuencias presentes en la música, y que estos niveles serán ajustados por el usuario técnico. Se muestra un bloque etiquetado como usuario técnico, ya que, es importante dejar en claro que solo el usuario técnico podrá mover estos parámetros de discriminación (ver Figura 1.5). Esos parámetros pasaran directamente al proceso de discriminación y causarán la activación de las válvulas de manera específica.

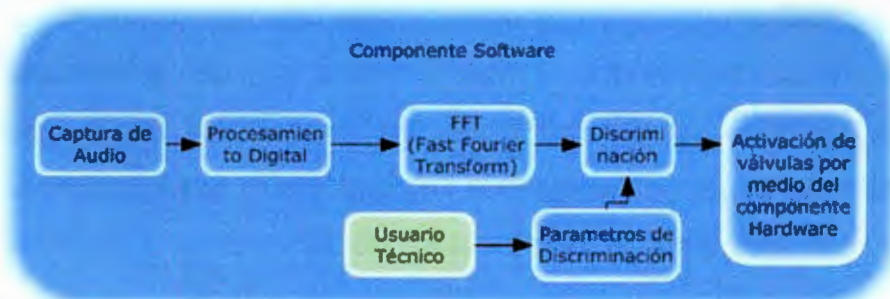


Figura 1.5 Componente Software a detalle

### 1.4.3 Componente Hardware del sistema

Otro de los componentes principales en esta investigación es el desarrollo del Hardware necesario para activar las electroválvulas en instantes precisos de la melodía en turno. En la Figura 1.6 se muestra la estructura del componente hardware.



Figura 1.6 Diagrama detallado de hardware

Los datos enviados por la etapa de software llegarán al componente de hardware por medio de la tarjeta de control. Esta tarjeta se encargará de interpretar los datos digitales para así enviarlos a la tarjeta de potencia. La tarjeta de control también tiene la finalidad de proteger la computadora de daños eléctricos que pudieran propiciarse en la etapa de hardware separando físicamente las señales de la PC por medio de acopladores ópticos. Por otro lado, la tarjeta de potencia recibirá los pulsos aislados de la PC y utilizarlos como patrón de disparo para una serie de 48 triacs (transistores de potencia). De éstos, 24 se emplean para activar las luces y 24 más para las válvulas. El objetivo principal de la tarjeta de potencia es, como su nombre lo indica, proveer de potencia necesaria para activar las luces y las electroválvulas, incluso de forma instantánea y paralela. Hay que tomar en cuenta que las lámparas y luces activadas al mismo tiempo consumen una gran cantidad de potencia.

### 1.5 Antecedentes

Las fuentes musicales han sido desarrolladas durante mucho tiempo atrás y han surgido debido a necesidades de lujo extremo en recintos públicos y propiedades personales. Una fuente musical es un tipo de fuente animada que crea un diseño estético y tridimensional logrado por medio de efectos de ondas de agua y luces que activadas en momentos estratégicos de tiempo expiden porte y elegancia al lugar en que se hospedan. Estos tiempos de activación dependen íntimamente de la información digital contenida en una pista musical, la cual en nuestro caso será una señal analógica capturada y muestreada por medio de la tarjeta de sonido de la PC del sistema, estos datos o muestras precisaran los tiempos de activación concretos para los chorros de agua, esto se verá más a fondo en el capítulo 3. También, uno de los capítulos avanzados de los cuales se hablará en los siguientes capítulos será la descripción de las actividades de la fuente como las de el espectro de frecuencia de las válvulas de las cuales de tiene

que enviar los datos y las señales a la tarjeta de control y seguido de esto a la de potencia.

### 1.5.1 Fuentes musicales y su evolución

**Fuente de Bodor.** El inventor Húngaro e ingeniero mecánico Peter Bodor construyó una fuente musical en un pueblo de Transilvania llamado Marosvásárhly (hoy en día Tírgu-Mures, Rumania) entre los años 1820 y 1822. Su fuente tenía un plano circular, con dos formaciones de estrella en los lados, un techo tipo domo soportado por pilares. El núcleo mecánico principal de su fuente fue una estructura hidráulica controlada por la fuerza del agua que hacía sonar unas populares campanas cada hora. En la parte superior había una estatua de un Neptuno dorada que giraba indicando las 24 horas. La fuente fue destruida en 1836 por una tormenta de nieve y nunca más fue restaurada. Después una copia idéntica de esta fuente fue construida en la isla Margit de Budapest en 1936 pero que no operaba con sistema hidráulico sino con un sistema eléctrico. Esta fue también destruida durante la Segunda Guerra Mundial pero restaurada en 1954 y nuevamente en 1997. Ahora es una atracción turística que toca música cada hora durante el día.

**Fuente Prismática** – El Mayor Robert W Speer y F.W. Darlington Un ingeniero de la estación ferrocarril interurbana de Denver dedicó la famosa fuente prismática a la ciudad de Park Lake. Esta fuente contaba con efectos de luz jamás antes vistos en público. Once columnas con coloridas luces brillantes emitían corrientes de agua acorde a los dramáticos patrones cambiantes. En lo alto de la torre norte de Park Pavilion un operador sentado en su escritorio movía palancas que ocasionaban las ondulaciones de las doce corrientes de agua así como también causaba el cambio de color en las luces de las columnas.

F.W. Darlington siempre dejó su firma en cada uno de sus diseños de fuentes musicales. Las boquillas con múltiples aros tipo canasta y que no fueron atribuidas

a él. El efecto "fan" un triple anillo de aspersores con múltiples boquillas y con ángulos en ahora un muy conocido efecto en las nuevas fuentes musicales.

**Fuente Garfield Park** - Después de la construcción de GreenHouses y el conservatorio de Indianápolis fue contratado Darlington para diseñar y construir las fuentes al este de Sunken Garden. Esas fuentes fueron las primeras en el país equipadas con elementos mecánicos que permitían el cambio de los chorros y desplegado de luces acordes a cada temporada del año. Durante el Memorial Day la fuente fue iluminada con luz roja, blanca y azul y en otros días de color dorado y blanco. Hoy en día estas fuentes siguen siendo populares y un centro de atracción para los visitantes. Esta fuente fue restaurada en 1997 y mejorada con un sistema de control de Atlantic Fountains Company.

**Fuente Pool of Industry** - Un ejemplo notable de una fuente musical coreografiada en directo era el Pool of Industry desarrollada en los años 1939 en Nueva Cork, donde tres operadores controlaron la fuente, servido de guía por un programa de papel que se leía bajo una ventana de vidrio como el rollo de papel de una pianola, que era mejor que controlar los efectos directamente como un rollo de piano, fue guiado con comandos que decían a los operadores cuándo empujar los botones y lanzar los interruptores. Esta fuente era más que sólo agua y luces, utilizaba 3 millones de watts en luces, una piscina gigantesca que contenía 1,400 boquillas de agua, 400 luces con un mecanismo que causaban que las luces cambiaran el color y el tiro de fuegos artificiales de más de 350 lanzadores crearon un espectáculo de noche sobre una balanza imponente. La música fue tocada en directo por la cinta de la exposición y transmitida por bocinas grandes instaladas en las áreas de la visualización. La función actualizada se exhibió en los recintos feriales en 1964 que carecían de estos efectos de color pero que usó tarjetas perforadas para la coreografía, tenía música en vivo, y utilizó el sistema entonces revolucionario de la luz que se filtraba (desarrollado por Bausch and Lomb para la fuente) que permitió que un lente oscuro de color y un

lente ligero de color causarían la misma luminosidad de la luz ahora. Fue junto a este proceso que 700,000 watts de luz causarían más de 3 mega candelas. Esta función también tenía luces solas con filtros de color deslizables múltiples para mezclar colores, y selecciones de boquillas que podían ser ajustadas, su dirección podía ser cambiada por actuadores hidráulicos o neumáticos.

En 1930 en Alemania, Otto y Przystawic concibieron la idea de crear una fuente que recrearía la belleza y la elegancia del ballet primero. Su primera visualización fue desarrollada en el restaurante de Resi en Berlín, visible a través de la serie de pasajes abovedados. Selecciones específicas de boquillas de agua crearon chorros de agua que no sólo aumentaban y caían sino que también controlaban el equipo por completo para girar alrededor de un círculo, imitando a bailarines legítimos y profesionales. Luces de color bañaron las formaciones de agua de la mudanza a color y eran el destello de agua. Los comensales estaban cautivados cuando el agua funcionó para recrear la música, controlado por un operador solo sentado en una consola de control. La ingeniería inteligente de Przystawic llevó las fuentes al próximo nivel.

Mientras viajaban por Berlín con sus fuentes imponentes, el empresario de espectáculos Harold Steinman vio la instalación de Resi y estaba encantado con la idea. Haciendo un contrato con Przystawic y fue Leon Leonidoff en Radio City Music Hall en la Ciudad de Nueva York. En el que por primera vez en la historia, Leonidoff reservó una función y Steinman realizó su presentación ante toda la sociedad estadounidense en enero de 1953. Cuatro semanas después, Dancing Waters tomó otro rumbo, siendo la primera función en ser nuevamente presentada en tan poco tiempo. Esta vez, Dancing Waters añadió su magia al programa especial de Pascua famoso de Radio City. El éxito de hacer funcionar a Waters en los EE.UU. incitaba que a Steinman los añadiera a su lista de funciones viajando, y así constituiría Dancing Waters, Inc. Rediseñado para ser portátil, las nuevas funciones podían ser desarmadas para el transporte y fueron proporcionadas con

palanganas de goma inflables para soportar el agua. Las funciones usaron la mejor ingeniería de Przystawic por ejemplo bombas de velocidad variable y válvulas complejas. Pasaron varios años y los motores de bomba de la fuente fueron conectados a resistores y a motores grandes, graduable por una hilera de palancas sobre la consola de control. Cambiando de lugar estas palancas, la cantidad del poder proporcionado al motor de cada bomba podía ser variada, que causó que las bombas se apresuraran o disminuyeran la velocidad y así cambiar la altura del agua. Además, la consola de control agrupó muchos de sus controles en interruptores solos, permitiendo que el operador jugara con muchos efectos inmediatamente con una mano. Las funciones de Przystawic eran únicas en lo que respecta a los efectos de agua. [www.wikipedia.com, Musical Fountains ,1997]

## 2 -REVISIÓN DE LITERATURA

Habiendo ya analizado previamente los componentes esenciales del sistema, se proseguirá con una breve comprensión de los fundamentos teóricos necesarios para poder proseguir.

### 2.1 Fundamentos teóricos

Durante el desarrollo de esta investigación, se han involucrado temas y conceptos teóricos que deben ser entendidos en su totalidad por lo que se verán de manera introductoria estos conceptos en este capítulo. Se analizará parte de lo relacionado con el sonido, sus propiedades y funciones, es importante recordar que la interfaz desarrollada se basará en ondas sonoras para activar efusiones de agua, intentando estar acorde a estas mismas. Otro tema a tratar será la captura de estas ondas sonoras así como su procesamiento en forma digital.

#### 2.1.1 Naturaleza del Sonido

El sonido son vibraciones de las partículas de aire que se propagan a través del aire, gracias a estas partículas del aire que se mueven, las vibraciones llegan a nuestros oídos. La audición en los seres humanos, ocurre siempre que una vibración tenga una frecuencia comprendida entre unos 15 y 20.000 hertz (unidad de medida de la frecuencia, es igual a la cantidad de ciclos por segundo de una señal), y su intensidad sea la suficiente para llegar al oído interno. Cuando las vibraciones pasan estos márgenes se habla de ultrasonidos y no son perceptibles al ser humano.

Las características del sonido se pueden medir y para ello se usa las unidades de hertzios (Hz) que miden la frecuencia de un sonido, esto es, cuantas veces vibra en un segundo, y los decibeles (db) que mide la intensidad en amplitud de una onda.



El oído y un micrófono incorporado a la tarjeta de sonido tienen un funcionamiento similar. Ambos transforman las vibraciones del aire en una señal eléctrica que puede ser comprendida y almacenada por sus respectivos cerebros. Esta señal puede ser guardada, manipulada o reproducida por los medios electrónicos adecuados. A continuación se muestran algunos ejemplos de ondas sonoras. [F. Alton Everest, 2001]

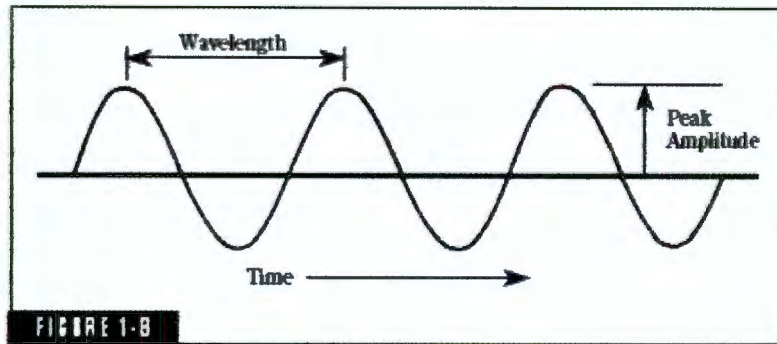


FIGURE 1-8

Figura 2.1 Características de una onda sonora

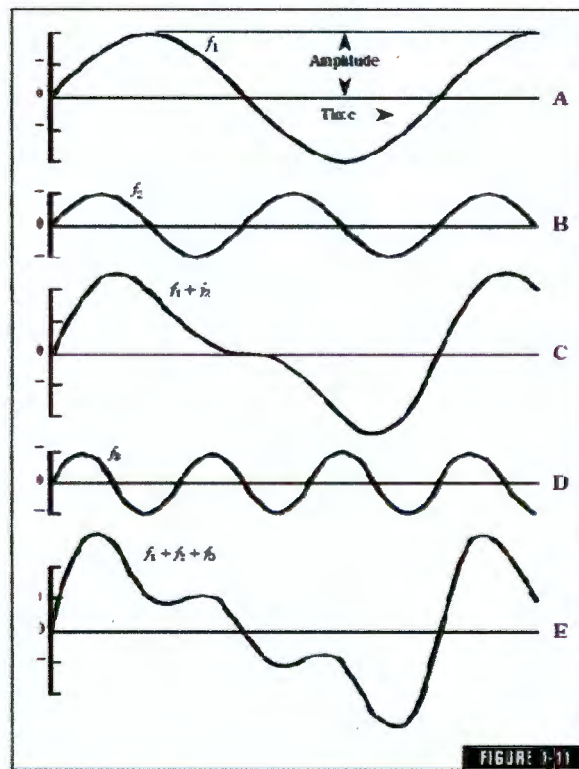


FIGURE 1-11

Figura 2.2 Muestra de una serie de ondas de sonido compuestas

### 2.1.2 Sonido Digital

La palabra digital indica la presencia de procesos numéricos para concretar un hecho (imagen, sonido, etc.), los sistemas de audio digital tienen circuitos eléctricos para guardar el registro de la música, en pocas palabras lo que hacen estos circuitos es grabar una larga cadena de números (digitalización o muestreo) con un dispositivo llamado convertidor análogo digital (ADC), que se encarga de monitorear la evolución de la onda y asignarle a cada momento un valor numérico, luego ese valor numérico es decodificado por un convertidor electrónico llamado “convertidor digital-análogo” (DAC – *Digital to Analog Converter*).

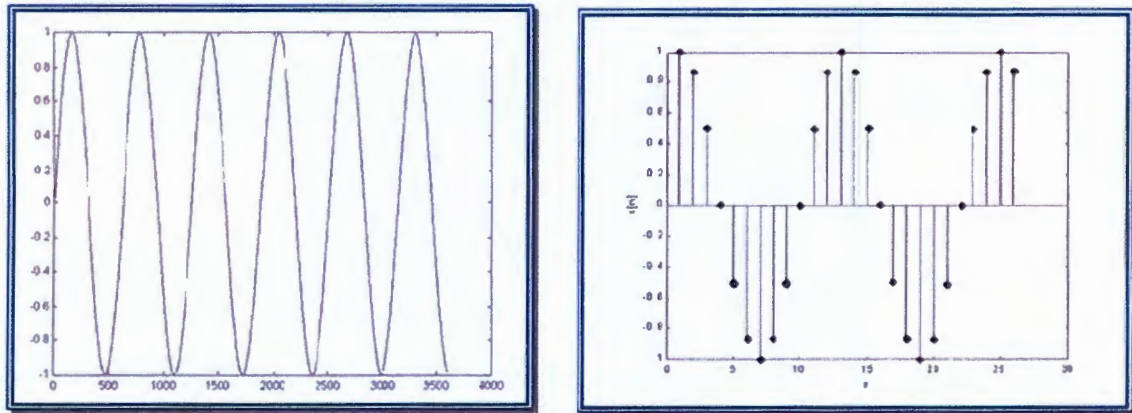


Figura 2.3 Muestreo de señales análogas (Paul M. Embree)

La calidad del sonido depende de la frecuencia del muestreo y a la resolución. Frecuencias de muestreo, se refiere al número de mediciones que se hacen por segundo. Cuanto mayor sea el numero de muestras mejor es la calidad del sonido, por ejemplo si la velocidad de muestreo es de una cada un segundo las variaciones del sonido que se produzcan en el intermedio no serán registradas. Según estudios, la frecuencia de muestreo debe ser el doble del sonido más alto que se pueda escuchar, como el oído humano puede escuchar aproximadamente hasta los 20.000 Hercios, la frecuencia óptima de muestreo será de 44,1 Khz.

(44.100 hercios), esta la frecuencia que se usa en los CD de música. Como los instrumentos o las voces humanas no pasan la frecuencia de los 10 Khz., con una frecuencia de muestreo de 32 Khz. es más que suficiente, al bajar mucho la frecuencia de muestreo el sonido se vuelve opaco o poco nítido pues se pierden las frecuencias agudas. [Paul M. Embree , 1995]

La resolución, el término hace referencia a la exactitud de las medidas de frecuencia. Se mide en bit, si la resolución es de 8 bit tenemos 256 niveles posibles ( $2^8=256$ ). Si se amplía a 16 bit el rango se extiende a 65.535 ( $2^{16}=65536$ ). Como referencia se puede decir que un disco compacto se graba a 44,1 Khz. y a una resolución de 16 bits.

Si se desea digitalizar 3 minutos de música a un muestreo de 44,1 Khz. y almacenando por cada muestra dos bytes (16 bits) se obtiene lo siguiente:

$$3 \text{ min} \cdot 60 \frac{\text{seg}}{\text{min}} \cdot 44100 \frac{\text{muestras}}{\text{seg}} \cdot 2 \frac{\text{bytes}}{\text{muestras}} = 15,876,000 \text{ bytes} \quad 2.1$$

El cálculo indica que para almacenar una canción de tres minutos con calidad profesional se necesitaran 16 MB aproximadamente. Es aquí donde surge el problema, pues aunque los discos duros u otros medios de almacenamiento han aumentado su capacidad, tener varias canciones significaría ocupar gran parte del disco de la computadora. Para solucionar este problema se han desarrollado formatos de archivo que permiten realizar grabaciones de sonido con muy buena calidad usando un método de compresión, el problema es que el sonido no puede ser editado para ser modificado a antojo del usuario.

### 2.1.3 Análisis de señales Tiempo-Frecuencia

Para el diseño de un interfaz de control por software para la fuente coreográfica, es necesario el análisis espectral de una señal para un correcto filtrado y activación de electroválvulas en tiempos específicos. Para ello, se deben utilizar fundamentos matemáticos que permitan obtener el contenido frecuencial de la música en instantes deseados e identificar el disparo de cada electroválvula con una frecuencia específica.

Los métodos del análisis tiempo-frecuencia comenzaron a emplearse hace unos 55 años y han sido aplicados con inmenso éxito en campos muy diferentes. Sin embargo, hoy en día seguimos sin contar con una completa y definitiva teoría de cómo varía un espectro en el tiempo. Se hace, por tanto, necesario desarrollar métodos que nos ayuden a comprender cómo el contenido espectral de una señal cambia en el tiempo y a relacionar esa variación con los mecanismos físicos que producen dicha señal.

Uno de los métodos más originales fue desarrollado a principio de los años cuarenta con el fin específico de comprender la naturaleza del habla humana. El método era simple y potente: dividir la señal en cortos intervalos de tiempo y analizar por transformada de Fourier cada intervalo (método hoy conocido como transformada corta de Fourier). Al método específico de dividir la señal en intervalos se le conoce como ventaneo (*windowing*). Entonces, para cada intervalo, obtenemos la densidad de energía espectral que nos indicará las frecuencias que existieron en ese intervalo temporal. La colección de todos esos espectros constituye una densidad tiempo-frecuencia comúnmente llamada espectrograma.

En 1946, un fascinante artículo de Gabor cristalizó la necesidad de describir una señal conjuntamente en tiempo y frecuencia. Gabor introdujo un gran número

de nuevos conceptos como los de señal analítica, principio de incertidumbre para señales, y diseñó lo que esencialmente es un espectrograma con una ventana gaussiana. Ville (1948) introdujo otra distribución conocida hoy como Distribución de Wigner-Ville. (La distribución de Wigner fue encontrada bastantes años antes (1932) y ha servido para tender un puente entre las mecánicas clásica y cuántica.) En los años posteriores se han encontrado numerosas distribuciones y métodos para desarrollarlas. La mayoría de ellas han surgido del uso efectivo de un formalismo general propuesto por L. Cohen hace unos 36 años y basado en una integral que depende de un núcleo: dependiendo del núcleo elegido, van surgiendo las distintas distribuciones en tiempo-frecuencia. [[www.wikipedia.com](http://www.wikipedia.com), 1997].

El análisis espectral se extendió a muchas otras situaciones más allá del estudio de la luz y otros fenómenos comunes, siendo la más importante el estudio de los fenómenos acústicos. Históricamente, la mayoría de las aplicaciones se realizaron en condiciones estacionarias, donde las características espectrales no varían en el tiempo. Sin embargo, lo que hoy conocemos como análisis espectral variable en el tiempo o análisis de señales en tiempo-frecuencia surge por mera necesidad. Existen señales en la naturaleza cuyo contenido espectral cambia en el tiempo. Quizás el primer ejemplo es el habla humano, donde las frecuencias van cambiando a medida que hablamos y, de hecho, es el cambio de frecuencias y su ordenación temporal el portador del mensaje.

Este tipo de señales son llamadas frecuentemente “no estacionarias” indicando que sus propiedades espectrales básicas cambian en el tiempo. El objetivo del análisis de señales en tiempo-frecuencia es describir la variación temporal del contenido espectral de éstas señales y ver cómo se relacionan las propiedades espectrales que varían en el tiempo con la fuente física que origina la señal y con el medio de propagación. Las ideas físicas y matemáticas necesarias

para llevar a cabo este objetivo, se han desarrollado durante décadas y continúan desarrollándose muy activamente hoy en día.

Los antecedentes históricos del desarrollo del análisis de señales en tiempo-frecuencia son numerosos. Sin embargo, la invención de la radio supuso un enorme empujón pues la modulación de amplitud y frecuencia son fenómenos inherentemente estacionarios. Esta necesidad de comprender tales situaciones condujo al desarrollo de nuevas ideas relacionadas con la naturaleza fundamental de las señales, como por ejemplo, el concepto de frecuencia instantánea.

La principal diferencia entre el análisis espectral estándar o análisis de Fourier y el análisis de señales en tiempo-frecuencia se puede explicar de una manera muy simple. El análisis de Fourier de una señal, sea estacionaria o no, nos dice qué frecuencias existen mientras dura esa señal. Sin embargo, no nos dice cuándo ocurren o aparecen esas frecuencias, esto es, no nos dice cómo evoluciona en el tiempo el espectro de la señal.

Un método para formalizar el análisis de señales en tiempo-frecuencia consiste en intentar generalizar el análisis de Fourier estándar. La idea básica consiste en extender el concepto de densidad espectral de energía a una función bidimensional que indique la intensidad por unidad de frecuencia y por unidad de tiempo. Tal función describiría entonces cómo el espectro evoluciona en el tiempo y sería llamada distribución, densidad ó representación tiempo-frecuencia. Durante los últimos 50 años se han propuesto numerosos métodos que en algún sentido han funcionado muy bien y que coinciden con nuestra idea intuitiva de cómo un espectro debería variar con el tiempo. [Stephan M. Bernsee, 1999]

## 2.2 Procesamiento digital de señales

El procesamiento digital de señales comienza con una señal digital la cual para las computadoras es una secuencia de valores digitales. La Figura 2.4 siguiente muestra un ejemplo claro de un sistema DSP (Procesamiento Digital de Señales – *Digital Signal Processing*) simple o una operación de procesamiento digital de señales. En la cual existe una secuencia de entrada  $x(n)$ , el operador  $\mathcal{O}\{\}$  y una secuencia de salida  $y(n)$ .

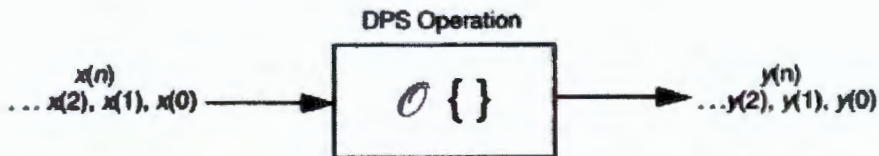


Figura 2.4 Operación DSP (Operación de Procesamiento Digital de Señales)

Un sistema completo DSP debe consistir de varias operaciones en la misma secuencia así como también operaciones a las secuencias resultantes. Debido a que las secuencias digitales son procesadas, todos los operadores en un DSP son operadores de tiempo discreto (lo opuesto a operadores en tiempo continuo utilizados comúnmente en los sistemas analógicos). Los operadores en tiempo discreto pueden ser clasificados ya sea variantes en el tiempo o invariantes en el tiempo y lineales o no lineales. La mayoría de los operadores descritos en este documento serán invariantes en el tiempo.

Los operadores son aplicados a las secuencia con el objetivo de obtener lo siguiente:

a) Extraer parámetros o características especiales de la secuencia (En nuestro caso trabajar únicamente con ciertas frecuencias obtenidas de la pista de audio

que es lo mismo que la secuencia de entrada  $x(n)$  donde  $n$  es el número de muestras almacenadas en el archivo de pista musical). [Paul M. Embree, 1995]

b) Producir una secuencia similar con características especiales mejoradas o eliminadas según sea el caso (En lo que concierne a esta investigación los datos serán analizados solo en la fase digital para generar una secuencia única de activación de las electroválvulas y después se codificarán exactamente igual para su salida a la tarjeta de sonido.)

c) Poder restaurar la secuencia a un estado anterior similar o modificado.

d) Poder codificar o comprimir la secuencia.

### 2.2.1 Muestreo digital

Para que una computadora digital pueda manipular una señal, esta señal debe ser muestreada con un intervalo específico. La Figura 2.5 muestra un ejemplo de una señal continua en el tiempo la cual ha sido muestreada con intervalos de  $T$  segundos. El conjunto resultante de números es llamado secuencia de la cual se habló anteriormente y a la cual se le aplicarán las operaciones específicas. Si la función en tiempo continua fuera  $x(t)$ , entonces las muestras serían  $x(nT)$  para  $n$ , siendo  $n$  un entero extendido en un rango finito de valores referidos al número de muestras realizadas [Vijay K. Madisetti, 1999].

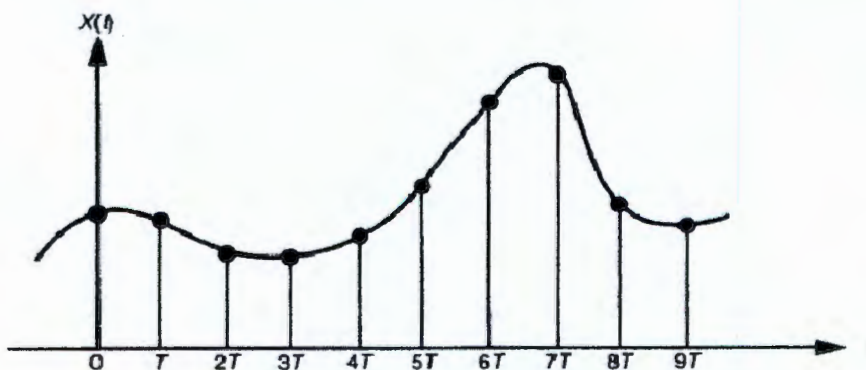


Figura 2.5 Teoría de muestreo simple



La definición de un muestreo apropiado es muy simple. Suponiendo que muestrea una señal continua de alguna manera, si puede reconstruir exactamente la señal análoga con las muestras obtenidas habrá logrado llevar a cabo un muestreo apropiado. Aún cuando los datos muestreados parezcan confusos o incompletos la información primordial o clave ha sido capturada si se dice que puede revertir el proceso (reconstrucción de la señal).

La Figura 2.6 muestra varias sinusoides antes y después de la digitalización. La línea continua representa la señal análoga entrando al ADC (Convertidor Análogo-Digital – *Analog to Digital Converter*) mientras que los cuadros pequeños representan la señal digital a la salida del ADC. En la Figura 2.4a la señal análoga es una valor de DC (Corriente Directa – *Direct Current*) constante, una onda cosenoidal de frecuencia cero. Por el hecho de que la señal análoga sea una serie de líneas rectas entre cada una de las muestras, toda la información necesaria para reconstruir la señal análoga está contenida en los datos digitales. Acorde a nuestra definición este es un muestreo apropiado.

La onda senoidal mostrada en la figura 2.6b tiene una frecuencia de 0.09 de la frecuencia de muestreo. Esto puede repercutir, por ejemplo, una onda senoidal a 90 ciclos por segundo siendo esta muestreada a 1000 muestras por segundo. Expresado de otra forma, hay 11.1 muestras tomadas en cada ciclo completo de la senoide. Esta situación es más complicada que el caso previo, porque la señal análoga no puede ser reconstruida simplemente dibujando líneas rectas entre estos puntos muestra. ¿Representan estas muestras una señal análoga propiamente?. La respuesta es sí, porque ninguna senoide o combinación de sinusoides producirá este patrón de muestras (con las restricciones razonables mencionadas anteriormente). Estas muestras corresponden a una sola señal análoga, después de haber sido muestreada, esta señal puede ser exactamente reconstruida. Nuevamente, es una instancia de un muestreo apropiado.

En la figura 2.6c, la situación se hace más difícil por el incremento de la frecuencia en la onda senoidal a 0.31 de la frecuencia de muestreo. Este resultado es solo 3.2 muestras por ciclo de la onda senoidal. Aquí las muestras están tan separadas que no parecen seguir la curva general de la señal análoga. Nuevamente la pregunta, ¿será esta una representación propia de una señal análoga?. Nuevamente, la respuesta es sí, y por la misma razón. Las muestras son una única representación de la señal análoga. Toda la información necesaria para reconstruir la forma de onda continua está contenida en los datos digitales del muestreo. La forma de lograr esto será comentada más adelante. Obviamente, esto es más sofisticado que solo dibujar líneas rectas entre estos puntos de datos. Esto puede sonar raro, pero es un muestreo apropiado acorde a nuestra definición.

En la figura d, la frecuencia de la señal análoga ha sido aumentada a 0.95 de la tasa de muestreo, con una mera cantidad de 1.05 muestras por ciclo. Lo cual indica que no es una representación correcta de un muestreo adecuado. Las muestras representan una onda senoidal diferente de una de las contenidas en la señal análoga. En particular, la onda senoidal original de frecuencia de 0.95 se mal interpreta como una onda senoidal de 0.05 en las señal digital. Este fenómeno de frecuencias cambiantes en las sinusoides durante el muestreo es llamado restitución (*aliasing*). Tal como un criminal puede asumir a un nombre o identidad diferente, (*alias*), la senoide asume otra frecuencia que no es la propia. Debido a que los datos digitales no están relacionados uno a uno a una señal análoga en particular, una reconstrucción de esta señal sería imposible. No hay nada en los datos muestreados que sugieran que la señal análoga original tiene una frecuencia de muestreo de 0.95 o de 0.05 la onda sinusoidal a ocultado su verdadera identidad completamente; el perfecto crimen ha sido cometido! Acorde a nuestra definición, esto es un ejemplo de un muestreo incorrecto.

El teorema del muestreo, frecuentemente es llamado teorema de Shannon, o teorema de muestreo de *Nyquist*. El teorema de muestreo indica que una señal continua puede ser propiamente muestreada, solo si ésta no contiene componentes frecuenciales por debajo de la mitad de la tasa de muestreo.

Por esta instancia, una tasa de muestreo de 2000 muestras por segundo requiere una señal análoga que esté compuesta por frecuencias por debajo de 1000 ciclos por segundo. Si existen frecuencias por debajo de este límite de la señal, estas se restituirán por frecuencias entre 0 y 1000 ciclos por segundo, combinándose estas con cualquier tipo de información que pueda encontrarse en la señal original.

Dos términos son extensamente usados cuando se habla del teorema de muestreo: la frecuencia de Nyquist y el muestreo de Nyquist. Desafortunadamente, su definición no ha sido estandarizada pero para entender esto, considere una señal análoga compuesta por frecuencias entre 3 khz. Para digitalizar correctamente esta señal esta debe ser muestreada a 6000 muestras por segundo ( 3khz ) o mayor. Suponiendo que nosotros elegimos un muestreo a 8000 muestras por segundo ( 8 khz ), permitiendo representar frecuencias entre 4 khz. En esta situación existen cuatro frecuencias: ( 1 ) la frecuencia mas alta en la señal, 3 khz; ( 2 ) el doble de la frecuencia, 6 khz; ( 3 ) la tasa de muestreo, 8 khz; y ( 4 ) la mitad de la tasa de muestreo, 4 khz. ¿Cual de esta cuatro es la frecuencia de Nyquist y cual el muestreo de Nyquist?. Esto depende de cómo se pregunte. Todos las posibles combinaciones son usadas. Afortunadamente, la mayoría de los autores son cuidadosos para definir cómo usar estos términos. En este libro ambos son definidos como la mitad de la tasa de muestreo.

La Figura 2.6d muestra como las frecuencia pueden ser cambiadas durante la restitución (*aliasing*). El punto clave a recordar es que la señal digital no puede contener frecuencias por debajo de la mitad de la tasa de muestreo (frecuencia de

Nyquist / tasa de muestreo de Nyquist). Cuando la frecuencia de una señal continua esta por arriba de la tasa de muestreo de Nyquist, la frecuencia de los datos muestreados es correcta y corresponde adecuadamente a la señal original a reconstruir. Por otro lado, cuando la frecuencia de la señal continua está por debajo de la tasa de muestreo de Nyquist, el fenómeno de restitución cambia la frecuencia a algo que no puede ser representado en los datos muestreados

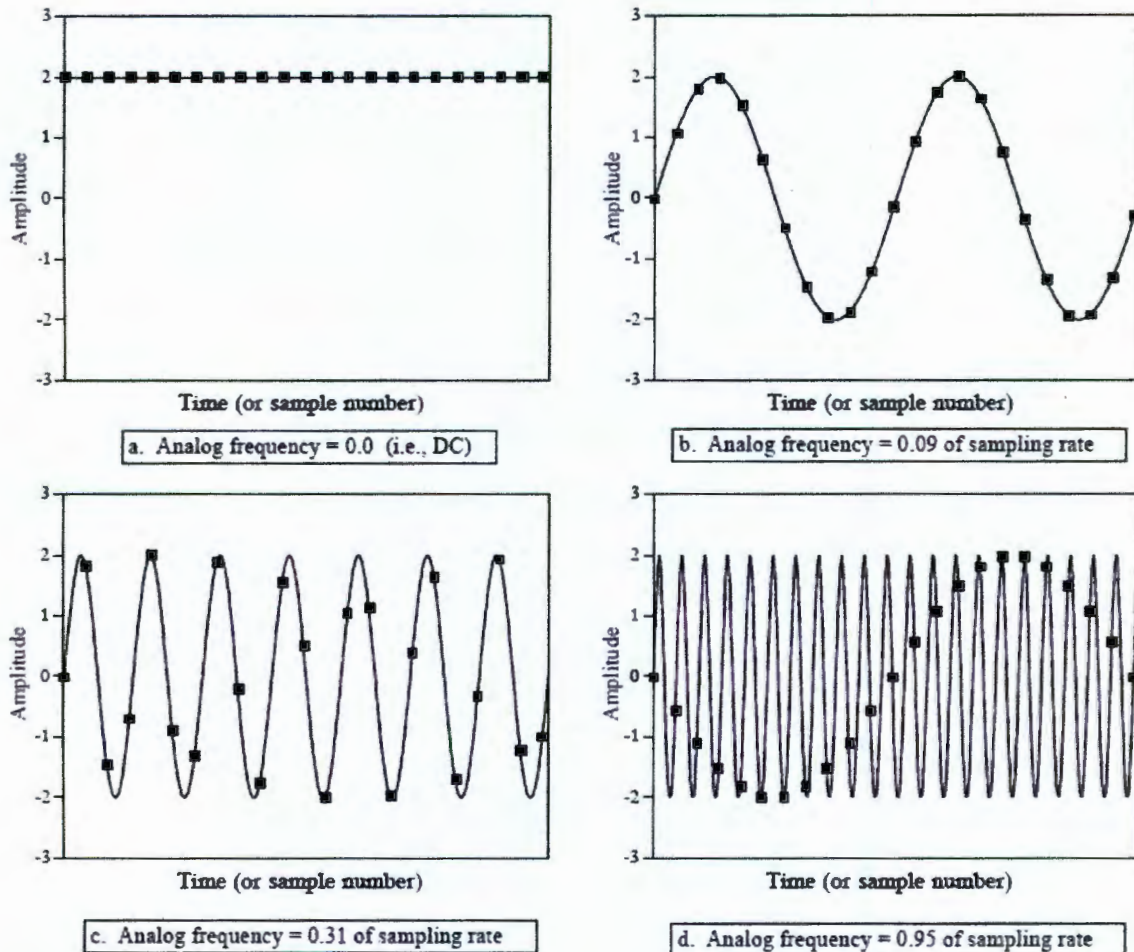


Figura 2.6 Muestreo a diferentes frecuencias a) Frecuencia de muestreo de 0 hz b) Frecuencia de muestreo de 0.09 o 9% del muestreo original, *aliasing* c) Frecuencia de muestreo de 0.31 o 31% también por debajo de la tasa de muestreo d) 95% de la tasa de muestreo, la señal si puede ser reconstruida nuevamente desde la señal original.

Una secuencia como representación de una señal continua en el tiempo tiene las siguientes características esenciales:

- La señal es muestreada. Esta tiene valores finitos solo en puntos discretos en tiempo.
- La señal está limitada a una longitud finita representando un intervalo de tiempo finito para la pista musical.
- La señal es cuantificada. Esta es limitada a intervalos discretos en amplitud, donde el tamaño y exactitud del intervalo dependerá de la cantidad de muestras que pueda proveer el conversor análogo- digital (ADC) así como también de la precisión aritmética (numero de bits) del procesador encargado del análisis digital de la señal [Vinay K. Ingle, 1997].

Para poder entender la naturaleza de los resultados que produce un procesamiento digital de señal, las características anteriores deben ser tomadas en cuenta así como también el teorema de muestreo siguiente que es la base de la manipulación de datos digitales.

### **2.3 Justificación de la digitalización**

Una señal digital no es audible, ya que requiere ser decodificada antes de su reproducción. Sin embargo, la codificación y posterior decodificación de una señal digital para su audición tiene varias ventajas, aunque también algunos inconvenientes.

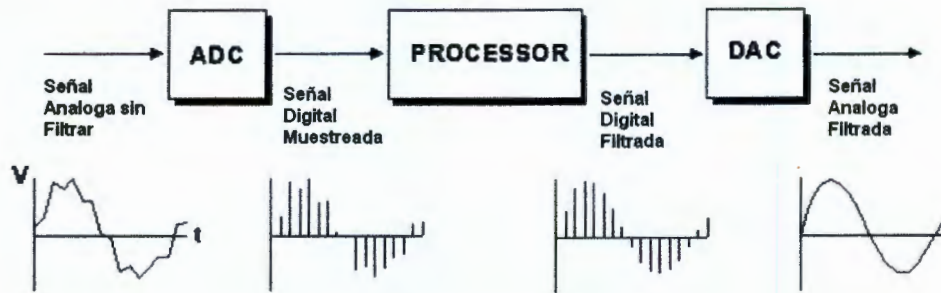


Figura 2.7 Proceso de muestreo de señales

### 2.3.1 Ventajas

La señal discreta (digital) es más fácil de transmitir, almacenar o manipular (en el caso del sonido: editar, comprimir, etc).

La señal digital es inmune al ruido. La señal digital es menos sensible que la analógica a las interferencias, etc.

Se puede tomar una muestra de sonido y cambiar cualquiera de sus parámetros para generar un sonido diferente sin tener que recrearlo en la realidad. (Las aplicaciones de esta ventaja en la generación de efectos especiales son infinitas).

La señal digital permite la múltiple generación infinita sin pérdidas de calidad. Esta ventaja sólo es aplicable a los formatos de disco óptico, la cinta magnética digital, aunque en menor medida que la analógica (que sólo soporta como mucho 4 o 5 generaciones), también va perdiendo información con la multi-generación [Alan V. Oppenheim, 1995].

Ante la pérdida de cierta cantidad de información, la señal digital puede ser reconstruida gracias a los sistemas de regeneración de señales (usados también para amplificarla, sin introducir distorsión). También cuenta, con sistemas de

detección y corrección de errores que, por ejemplo, permiten introducir el valor de una muestra dañada, obteniendo el valor medio de las muestras adyacentes (interpolación).

La señal digital puede ser enviada a casi cualquier punto del planeta en cualquier momento a un muy bajo costo a través de internet y a partir de aquí puede ser reenviada a su remitente o a algún otro destino. Esto sin que la señal sufra variaciones o alteraciones de calidad severas.

### **2.3.2 Inconvenientes**

Se necesita una conversión analógica-digital previa y una decodificación posterior, en el momento de la recepción.

Hay una pérdida inherente de información al convertir la información continua en discreta. Por mínimo e insignificante que resulte siempre hay un error de cuantificación que impide que la señal digital sea exactamente equivalente a la analógica que la originó.

La señal digital requiere mayor ancho de banda para ser transmitida que la analógica. Además, requiere una sincronización precisa entre los tiempos del reloj de transmisor, con respecto a los del receptor. Un desfase, por mínimo que sea, cambia por completo la señal. Si se utiliza compresión con pérdida, será imposible reconstruir la señal original. [Richard G. Lyons, 1948].

## **2.4 Algoritmos de Procesamiento**

Esta sección de la investigación se podría considerar como la más importante ya que sin los filtros pasa-bandas y procesos de adquisición de audio no se podrían activar las válvulas y luces acordes a la pista musical en turno, es

por eso que este capítulo trata por completo todos los aspectos necesarios para entender los procesos de procesamiento de la pista musical. Cabe mencionar también que el proceso de activación por medio de filtrado y selección de rangos de activación hace a esta investigación innovadora con respecto al diseño de otras fuentes comerciales existentes.

#### 2.4.1 Transformada Discreta de Fourier

En matemáticas, la **Transformada de Fourier discreta**, designada con frecuencia por la abreviatura DFT (Transformada Discreta de Fourier - *Discrete Fourier Transform*), y a la que en ocasiones se denomina Transformada de Fourier Finita, es una transformada de Fourier ampliamente empleada en tratamiento de señales y en campos afines para analizar las frecuencias presentes en una señal muestreada, resolver ecuaciones diferenciales parciales y realizar otras operaciones, como convoluciones. La transformada de Fourier discreta puede calcularse de modo muy eficiente mediante el algoritmo FFT (Transformada Rápida de Fourier - *Fast Fourier Transform*).

La secuencia de  $n$  números complejos  $x_0, \dots, x_{n-1}$  se transforma en la secuencia de  $n$  números complejos  $f_0, \dots, f_{n-1}$  mediante dicha transformada según la fórmula.

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \quad j = 0, \dots, n-1 \quad 2.2$$

Siendo  $e$  la base de los logaritmos naturales,  $i$  la unidad imaginaria ( $i^2 = -1$ ), y  $\pi$  el número pi. Esta transformada se nota con frecuencia mediante el símbolo  $\mathcal{F}$ , como en  $\mathbf{f} = \mathcal{F}(\mathbf{x})$  o en  $\mathcal{F}\mathbf{x}$ .

La transformada de Fourier discreta inversa (por sus siglas en inglés IDFT - *Inverse Fourier Transform*) se calcula, por otra parte, mediante:



$$x_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j e^{\frac{2\pi i}{n} jk} \quad k = 0, \dots, n-1 \quad 2.3$$

Nótese que el factor de normalización que multiplica a la transformada y a su inversa (en las fórmulas dadas, 1 y  $1/n$ ) y los signos de los exponentes son convencionales, y pueden diferir en otras presentaciones de la transformada de Fourier discreta. Lo importante es que la DFT y la IDFT tengan exponentes de signos contrarios y que el producto de sus factores de normalización sea  $1/n$ . Un factor de normalización de  $1/\sqrt{n}$  tanto para la transformada directa como para la inversa hace las transformaciones unitarias, lo que presenta ciertas ventajas teóricas, pero en la práctica suele ser más conveniente realizar la operación de escalado una única vez. [Albert Boggess, 1992]

#### 2.4.2 Transformada Rápida de Fourier

FFT es la abreviatura usual (Transformada Rápida de Fourier - *Fast Fourier Transform*) de un eficiente algoritmo que permite calcular la transformada de Fourier discreta (DFT) y su inversa. La FFT es de gran importancia en una amplia variedad de aplicaciones, desde el tratamiento digital de señales y filtrado digital en general a la resolución de ecuaciones diferenciales parciales o los algoritmos de multiplicación rápida de grandes enteros.

Sean  $x_0, \dots, x_{n-1}$  números complejos. La transformada discreta de Fourier (DFT, por sus siglas en inglés) se define como

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \quad j = 0, \dots, n-1. \quad 2.4$$

La evaluación directa de esa fórmula se requieren  $O(n^2)$  operaciones aritméticas. Mediante un algoritmo FFT se puede obtener un resultado similar con sólo  $O(n \log n)$  operaciones. En general, dichos algoritmos dependen de la factorización de  $n$  pero, al contrario de lo que frecuentemente se cree, existen FFT para cualquier  $n$ , incluso con  $n$  primo.

La idea que permite esta optimización es la descomposición de la transformada a tratar en otras más simples y éstas a su vez hasta llegar a transformadas de 2 elementos donde  $k$  puede tomar los valores 0 y 1. Una vez resueltas las transformadas más simples hay que agruparlas en otras de nivel superior que deben resolverse de nuevo y así sucesivamente hasta llegar al nivel más alto. Al final de este proceso, los resultados obtenidos deben reordenarse.

Dado que la transformada discreta de Fourier inversa es análoga a la transformada discreta de Fourier, con distinto signo en el exponente y un factor  $1/n$ , cualquier algoritmo FFT puede ser fácilmente adaptado para el cálculo de la transformada inversa [Albert Bogges, 1992].

### 2.4.3 El puerto Paralelo y su historia

En 1981, IBM (International Business Machines) introdujo la Computadora Personal (PC). El puerto paralelo (SPP, Puerto Paralelo Estándar - *Standart Parallel Port*) estaba incluido en el primer PC y se agregó a éste como una alternativa al bajo rendimiento del puerto serial, para utilizarlo como controlador de las impresoras de matriz de punto de alto desempeño. Este puerto tenía la capacidad de transmitir 8 bits de datos a la vez (del PC a la impresora), mientras

que el puerto serial lo hacía de uno en uno. En el momento que el puerto paralelo fue presentado, las impresoras de punto fueron el principal dispositivo externo que se conectó a éste. Al hacerse extensamente utilizado, el puerto paralelo llegó a ser la respuesta para conectar dispositivos más rápidos.

Después de este inicio, tres grandes grupos de problemas aparecieron a los desarrolladores y usuarios de este puerto: Primero, aunque éste había aumentado su velocidad considerablemente, no había cambio en la arquitectura o desempeño. La máxima velocidad de transferencia alcanzable estaba por los 150 kbytes /seg. y era extremadamente dependiente del software. Segundo, no había un estándar para la interface eléctrica. Esto causaba muchos problemas cuando se quería garantizar la operación en múltiples plataformas. Por último, la forma de diseño que le dieron, limitaba la distancia de los cables externos hasta un máximo de 1,8 metros. En 1991 hubo una reunión de fabricantes de modo que se pudiera desarrollar un nuevo estándar para el control inteligente de impresoras a través de una red. Estos fabricantes, donde estaban incluidos Lexmark, IBM, Texas Instruments y otros, formaron la *Network Printing Alliance* (NPA), como una respuesta a estas necesidades.

**Evolución del puerto paralelo** - Desde la introducción del PC al mercado, el puerto paralelo ha sufrido varias modificaciones para hacerlo más veloz. Ya que el puerto original era unidireccional, se creó el puerto bidireccional. El puerto bidireccional fue introducido con el PS/2 compatible. Este permite una comunicación de 8 bits en ambas direcciones. Algo interesante de notar es que el puerto original tenía la posibilidad de ser bidireccional realizando una conexión entre dos pines de un componente electrónico que venía incluido en éste. (Dicho de otro modo, el puerto original es bidireccional en diseño básico, pero no en el diseño operacional). Finalmente se han creado el *Enhanced Parallel Port* (EPP) y el *Extended Capability Port* (ECP). Estos dos últimos son al puerto estándar como el Pentium al 286, además de ser bidireccionales. [Jan Axelson, 2000]

Inicialmente el puerto paralelo se utilizó para la comunicación con impresoras. Actualmente se utiliza también para manejar otros periféricos como CD ROM, cintas de copia de respaldo, discos duros, tarjetas de red, protectores de copia, scanners, etc. Los tipos de puertos son:

#### RESUMEN DE TIPOS DE PUERTOS

- Puerto paralelo estándar (*Standart Parallel Port SPP*)
- Puerto Paralelo PS/2 (bidireccional)
- *Enhanced Parallel Port (EPP)* - Puerto paralelo mejorado
- *Extended Capability Port (ECP)* – Puerto con Capacidad Extendida

Y sus características están representadas en la siguiente tabla

**Tabla 2.1 Tipos de puertos y sus velocidades**

	<b>SPP</b>	<b>PS/2</b>	<b>EPP</b>	<b>ECP</b>
<b>Fecha de Introducción</b>	1981	1987	1994	1994
<b>Fabricante</b>	IBM	IBM	Intel, Xircom y Zenith Data Systems	Hewlett Packard y Microsoft
<b>Bidireccional</b>	No	Si	Si	Si
<b>DMA</b>	No	No	No	Si
<b>Velocidad</b>	150 Kbytes/seg.	150 Kbytes/seg.	2 Mbytes/seg.	2 Mbytes/seg.

### 3 - DISEÑO E IMPLEMENTACIÓN DE HARDWARE

Para la comunicación de la pantalla de selección de pista musical con la fuente musicalizada es necesaria una interfaz capaz de interpretar los pulsos que envía la PC por medio del puerto para poder activar las electroválvulas, para esto se ha dividido esta sección en dos etapas, la de control y la de potencia. Iniciaremos con una breve descripción del puerto paralelo con el que se lleva a cabo la comunicación de la PC con las tarjetas electrónicas.



Figura 3.1 Diagrama a bloques del desarrollo del hardware

#### 3.1 Características del puerto de comunicación (*Parallel Port*)

Durante el desarrollo de esta investigación se propuso el puerto paralelo como el puerto de comunicación primario entre la PC y las tarjetas que

comandaran la fuente musical por su fácil utilización y programación. Este puerto es muy utilizado por las impresoras y tiene conectores mecánicos como los que se muestran en la Figura 3.2, esto claro habiendo ciertos cambios para el conector hembra y para el conector macho respectivamente. [Jan Axelson, 2000]

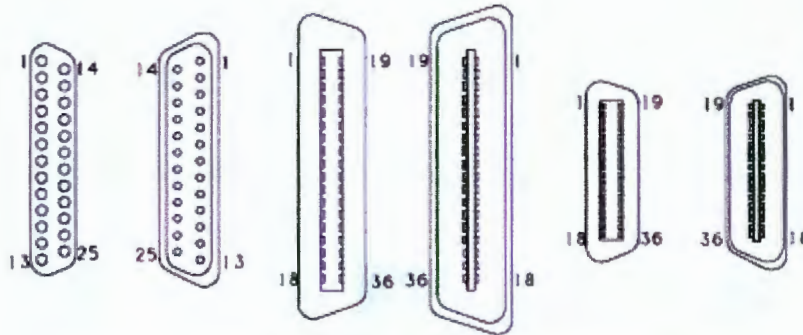


Figura 3.2 Configuración de pines del puerto paralelo

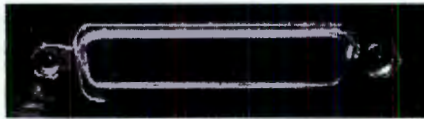


Figura 3.3 Vista frontal de un conector DB25

**Puerto de datos (Pin 2 al 9).** Su dirección comúnmente es la 0x378 (en varias computadoras esta dirección puede variar y ser 0x278 o 0x3BC), y es de sólo escritura, por este registro se enviarán los datos por el exterior de la PC y serán dirigidos a los latch conectados en cascada que a su vez activarán las electroválvulas en tiempos específicos por medio de la tarjeta de potencia durante la reproducción de la pista musical en turno. Cada latch es activado por medio del decoder de 3 a 8 líneas. En la Figura 3.3 se muestran más claramente los pines de este puerto.

**Puerto de estado (Pin 15, 13, 12, 10 y 11).** Su dirección comúnmente es la 0x379 (pero puede variar a 0x279 o 2BD) y es de solo lectura, por aquí se

enviaran señales eléctricas al ordenador, de este registro solo se utilizan los cinco bits de más peso, que son el bit 7, 6, 5, 4 y 3 teniendo en cuenta que el bit 7 funciona en modo invertido como se muestra en la Figura 3.3. Este puerto es utilizado para conectar el panel interactivo (Capítulo 4) así como 8 entradas más de uso auxiliar.

**Puerto de control (Pin 1, 14, 16 y 17).** Es el correspondiente a la dirección 0x37A (también 0x27A o 0x3BE), y es de escritura, es decir, podremos enviar señales eléctricas, según nuestras necesidades. De los 8 bits de este registro solo se utilizan los cuatro de menor peso o sea el 0, 1, 2 y 3, con un pequeño detalle, los bits 0, 1, y 3 están invertidos como se muestra en la Figura 3.4, los pines invertidos tienen un guiñón superior (en la Figura 3.4 se muestra direcciones en decimal y las antes explicadas en sistema hexadecimal).

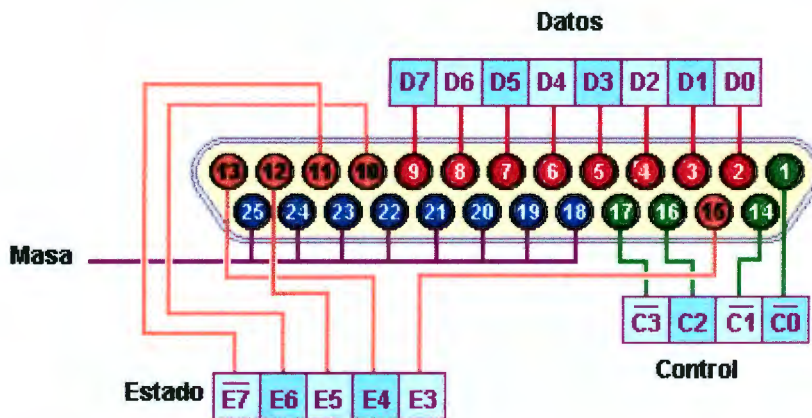


Figura 3.4 Diagrama explicativo del puerto paralelo

La tensión de trabajo del puerto es de 5 volts, por lo que necesitamos una fuente estabilizada o regulada de tensión, esto es importante tenerlo en cuenta, ya que estaremos enviando señales al puerto. Por otro lado, si bien se puede utilizar la PC para enviar señales al exterior sin necesidad de una fuente externa, pero es

recomendable utilizarla y así no se exige demasiado al puerto y se evitan problemas. [Jan Axelson, 2000]

### 3.2 Tarjeta de Control

La etapa de control es la que recibe los pulsos enviados por el software e interpretar cada uno de estos pulsos para así activar las electroválvulas. Los disparos son independientes en cada una de las electroválvulas al igual que la de las luces, esto se logra por medio de 6 latch (del inglés Seguro o Cerrojo, circuito eléctrico para retener datos y transmitirlos de una entrada a una salida por medio de un pulso específico restringido) octal tipo D modelo 74HC573 (ANEXO 1) conectados en cascada y disparados independientemente para que en cada uno de sus buses de datos maneje las señales independientemente. Los disparos se llevan a cabo por medio del puerto de datos del puerto paralelo y son accionados en tiempos concretos para una activación correcta de los elementos de potencia (electroválvulas y luces).

En el siguiente diagrama se muestra la conexión en cascada de los latch y de los pines de disparo para cada uno de estos.

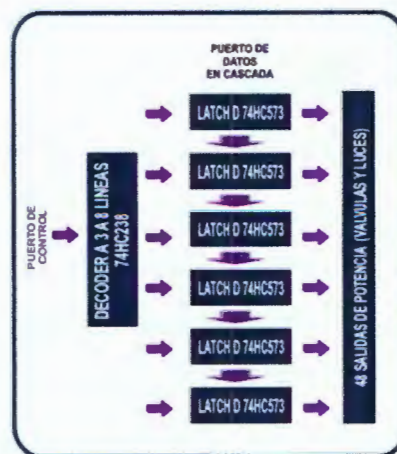


Figura 3.5 Diagrama a bloques de la tarjeta de control



Cada uno de los latch D refleja la configuración actual que existe en las entradas  $D_{0-7}$  hacia las salidas  $Q_{0-7}$  cada vez que hay un flanco de subida en el pin 11 (LE en inglés *Latch Enabled*). Todos los pines OE (*Output Enabled*) están conectados a tierra para activar así las salidas tri-estado, estas salidas son tri-estado (alta impedancia) por que tienen la capacidad de ser conectadas a un bus y manejar datos en cascada como la configuración utilizada en las entradas de los latch. Cada uno de los pines LE son activados por medio del *decoder* de 3 a 8 líneas esto para permitir la activación por separado de hasta 8 latch, por consiguiente tener hasta 64 salidas, en el caso de esta investigación solo se utilizaron 6 latch y por consiguiente 48 salidas independientes para la activación de cada electroválvula.

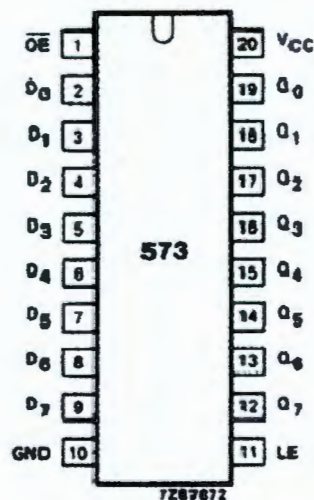


Figura 3.6 Especificación de pines del Latch Octal tipo D 74HC573

Debido a que se requieren 48 salidas individuales es necesario que el *decoder* envíe secuencialmente pulsos a cada pin LE y por consiguiente se tendrían que tomar en cuenta todos estos tiempos de activación de cada latch así como los tiempos de propagación de cada elemento (*decoder* y *latch*). El tiempo de propagación  $D_n$  a  $Q_n$  de este modelo de latch es de 150 ns (nanosegundos)

máximo como se muestra en la Tabla 2.1 (tiempo que tarda el latch en reflejar los datos de entrada D hacia las salidas Q) y el tiempo de propagación más importante que es el LE a  $Q_n$  es de 150 ns máximo (tiempo que tarda para reflejar los datos de entrada D hacia las salidas Q mediante un flanco de subida en el pin LE de este elemento).

Tabla 3.1 Tiempos de propagación por latch

SYMBOL	PARAMETER	$T_{amb}$ (°C)						UNIT	TEST CONDITIONS		
		74HC							$V_{CC}$ (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
$t_{PHL}/t_{PLH}$	propagation delay $D_n$ to $Q_n$		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
$t_{PHL}/t_{PLH}$	propagation delay LE to $Q_n$		50 18 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7
$t_{PZH}/t_{PZL}$	3-state output enable time $\overline{OE}$ to $Q_n$		44 16 13	140 28 24		175 35 30		210 42 36	ns	2.0 4.5 6.0	Fig.8
$t_{PHZ}/t_{PLZ}$	3-state output disable time $\overline{OE}$ to $Q_n$		55 20 16	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.8
$t_{THL}/t_{TLH}$	output transition time		14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig.6
$t_W$	enable pulse width HIGH	80 16 14	14 5 4		100 20 17		120 24 20		ns	2.0 4.5 6.0	Fig.7
$t_{su}$	set-up time $D_n$ to LE	50 10 9	11 4 3		65 13 11		75 15 13		ns	2.0 4.5 6.0	Fig.9
$t_h$	hold time $D_n$ to LE	5 5 5	3 1 1		5 5 5		5 5 5		ns	2.0 4.5 6.0	Fig.9

Teniendo ya los tiempos de propagación de cada latch solo queda por analizar, el tiempo del decodificador y así obtener el tiempo máximo de configuración de salidas de la tarjeta de control. El tiempo máximo de propagación  $A_n$  a  $Y_n$  del decodificador 74HC238 (ANEXO 2) es de 150 ns (tiempo que tarda en cambiar las salidas Y debido a un cambio en las entradas A) por lo que el tiempo

máximo promedio para configurar los 6 grupos de 8 bits de cada latch estaría dado por la siguiente relación:

$$T_{\max} = T_{PC} + T_D + T_{LE} + (6T_L) \quad 3.1$$

Donde:

$T_{PC}$  – Tiempo que le toma a la PC configurar el puerto de datos y el de control.

$T_D$  – Tiempo de propagación del decodificador

$T_L$  – Tiempo de propagación del latch.

$T_{LE}$  – Tiempo de propagación de LE (*Latch Enable*)

$$T_{\max} = (1000\text{ns}) + (150\text{ns}) + (45\text{ns}) + (6 \cdot 150\text{ns}) = 2050 \text{ ns} = 2 \text{ uS} \quad 3.2$$

Haciendo una aproximación para nuestra investigación y con los datos obtenidos anteriormente y considerando un tiempo máximo para  $T_{PC}$  de 1uS (un microsegundo) a la tarjeta de control le tomaría aproximadamente 2uS configurar las 48 salidas independientes, tiempo reducido que no afecta a nuestro sistema de activación de válvulas.

Lo anterior finalizaría en la construcción de una tarjeta que hospedara todos estos elementos de una forma eficaz y sencilla. Para esto se utilizó Protel y se diseñó un esquema eléctrico como el que se muestra en la Figura 3.6. El diseño se hizo a dos caras por la complejidad del circuito, pero se mantuvo la estética de la misma para no perder valores de calidad.

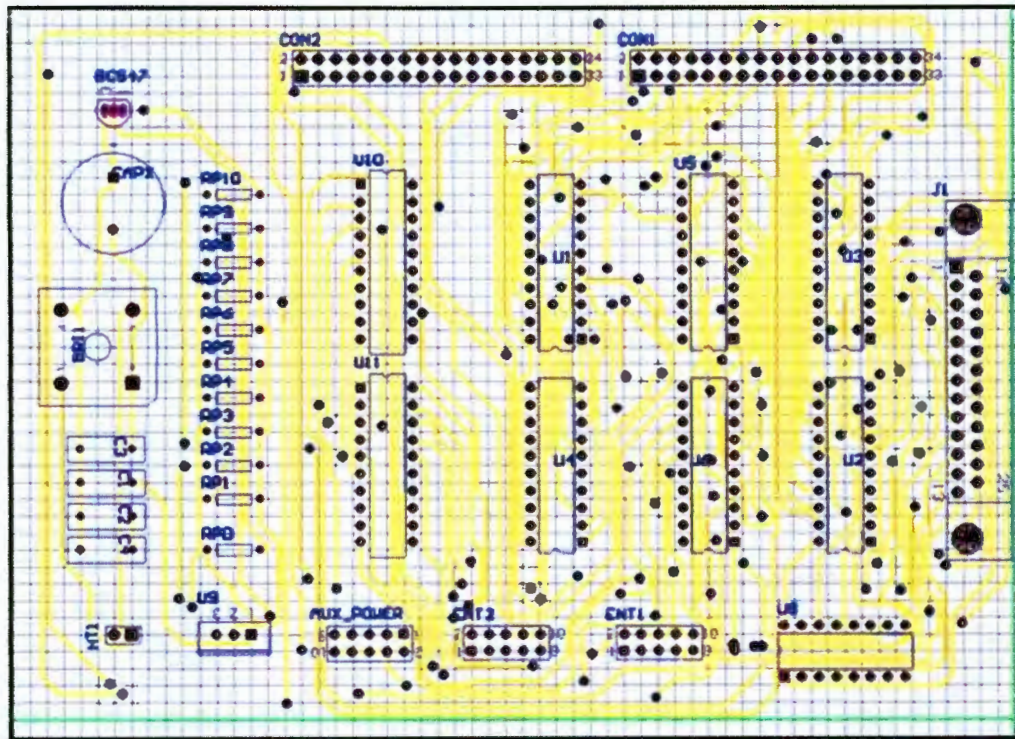


Figura 3.7 PCB de la tarjeta de control

Se puede notar considerablemente que las pistas y el auto ruteado son de gran calidad ya que deja pocos barrenos para manufactura *trough-hole* y están organizadas de la manera más compacta posible para minimizar los costos respecto al material. Esta última una de las razones por la que se eligió este software de diseño electrónico. El diseño final de la tarjeta se muestra en la Figura 2.7 este diseño ya está implementado en una tarjeta fenólica y está listo para ser utilizado en la fuente. En esta fase no se requiere un cálculo extenso sobre grosores de pistas ya que la corriente utilizada en esta tarjeta es muy reducida del orden de los 120 mA. (miliamperios) máximo en un caso en el que todos los elementos y sus salidas estén siendo utilizados a máxima carga.



**Figura 3.8** Manufactura final de la tarjeta de control

Es importante hacer notar que en el diseño se agregaron dos conectores tipo europeos de 64 pines para uso con cable plano, esto para una fácil conexión y también para evitar desgaste por conexión o desconexión continua de la tableta, ya que el cable plano soporta mucho esfuerzo por fatiga debido a su forma plana.

Otras de las razones por las que se utilizaron este tipo de conectores y cables es porque debido a la gran cantidad de pines que salen de la tarjeta se tendría que instalar canaletas para soportar esta gran cantidad de cables si se conectara con cables independientes, es por eso que se tomo en cuenta un conector que utilizará el menos espacio disponible así como también un cable con estas mismas características.

### **3.3 Tarjeta de Potencia**

La tarjeta de control consta de 48 salidas de potencia de las cuales 24 son para comandar las electroválvulas y las otras 24 para los disparos independientes de las luces, estas últimas cuentan con una capacidad de carga máxima de 4.5 A(amperes) ya que las luces son de 24 Volts – 1100 VCA. La tensión baja de este

tipo de luminarias es muy común en fuentes sub-acuáticas debido a factores de seguridad y posibles electrocuciones de los usuarios de las fuentes u obra civil donde será instalada la luminaria. Las salidas de potencia que comandan las electroválvulas solamente consumen 300 mA. cada una por lo que no fue necesario incrementar tanto el grosor de las pistas de la tarjeta de potencia en esta sección.

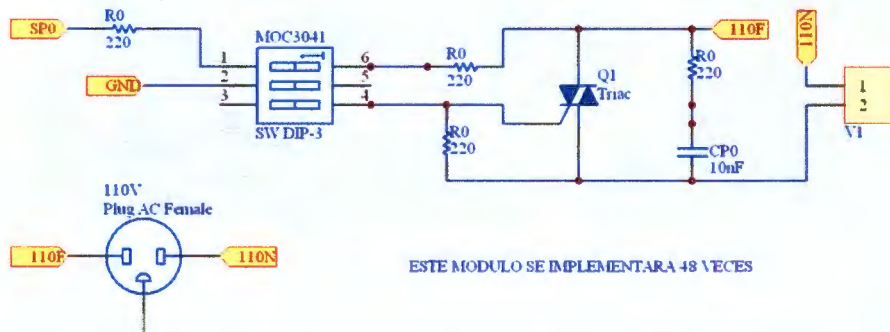


Figura 3.9 Esquema del modulo de potencia

En el esquema anterior se utilizó un Triacs modelo BT138 (ANEXO 5) por su capacidad de corriente máxima soportada a través de la MT1 y MT2 (Se pueden ver todas las características en la sección de apéndices en el área de hojas de datos). El factor frecuencia máxima de activación no es tan relevante en esta investigación debido a el valor estándar es suficiente para la activación correcta de las válvulas y luces.

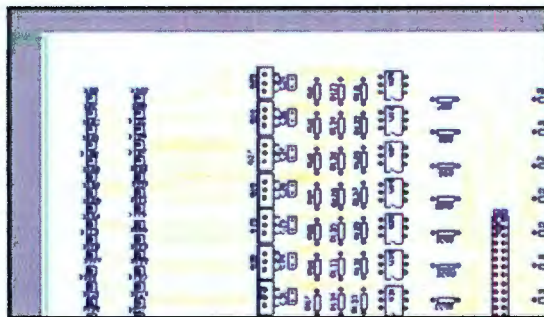


Figura 3.10 Segmento del PCB de todos los módulos de potencia

Los fototriacs conectados a la base de cada BT138 son para protección y aislamiento de la etapa de potencia con la etapa de control, esto para evitar cortos circuitos que puedan repercutir en la tarjeta madre de la PC encargada de enviar los disparos. Estos fototriacs tienen capacidad máxima de 1 amperio pero debido a que en esta investigación solo son usados para disparar la base de los Triacs BT138, solamente se usan aproximadamente 20 mA. de su capacidad máxima (ANEXO 6).

### 3.4 Diseño con PROTEL DXP

El programa utilizado para el desarrollo de las tarjetas fenólicas fue Protel DXP ya que gracias a su gran flexibilidad y escalabilidad para el desarrollo de proyectos electrónicos me permitió elaborar los diseños de las tarjetas electrónicas de una manera sencilla y estética. Se diferencia de varios programas de diseño electrónico por su gran potencia en el auto ruteado de las pistas y por si facilidad de diseño de esquemáticos, contando también con una extensa librería de componentes electrónicos.

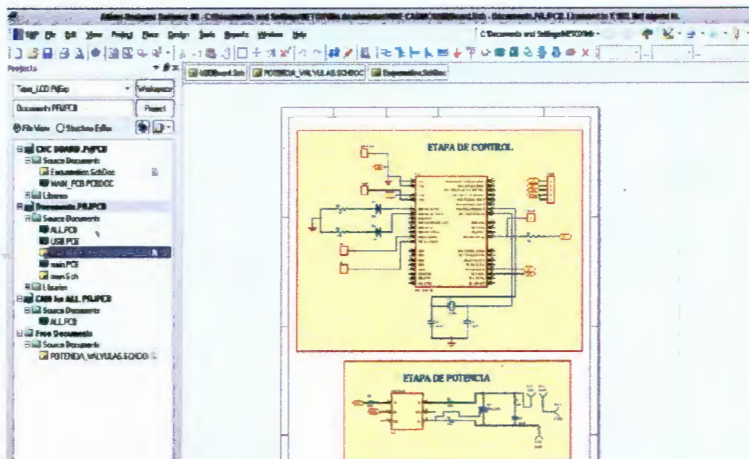


Figura 3.11 Ambiente de desarrollo en Protel

### 3.5 Características básicas de electroválvulas usadas

Las electroválvulas utilizadas durante el desarrollo de esta investigación son las electroválvulas de la compañía Parker (Figura 3.12). Se muestran las electroválvulas y sus características debido a que se requirió la frecuencia máxima de apertura como dato para generar límites de disparo desde software. Durante el desarrollo de esta investigación se hicieron varias pruebas a las electroválvulas logrando así la obtención de la frecuencia máxima de apertura.



**Figura 3.12 Electroválvula marca PARKER**

**Tabla 3.2 Características de la electro-válvula**

Tensión de activación de las bobinas	110 V ~ CA
Máxima Frecuencia de apertura y cierre	40 Hertz
Máxima presión en PSI	450 PSI
Dimensiones Físicas	½" in en la boquilla

También muestran las válvulas que se utilizarán durante la instalación de la fuente coreográfica a modo de comentario ya que en esta investigación no se requiere un estudio arduo sobre válvulas y tuberías, pero se mencionan como conocimiento general y para apoyo para entender la funcionalidad y la importancia de formar parte del proyecto.



## 4 - DESARROLLO DE SOFTWARE

Como parte esencial de nuestro sistema y también como parte de la interfaz a desarrollar en esta investigación está el componente de Software, que forma parte clave de lo innovador de esta investigación, debido a que es el encargado de adquirir el audio, procesarlo, interactuar con el usuario, filtrarlo, discriminarlo y por último enviar estos datos manipulados al componente de hardware para generar la representación física de esas señales de audio. Como se mencionó en capítulos anteriores existen programas comerciales especialmente diseñados para coreografiar el audio cada lapso de tiempo lo que nos haría pensar que podríamos comprar ese software y desarrollar únicamente el componente de hardware para usarlo en conjunto, pero no es así, como mencioné antes lo innovador es la manera en que este software procesa el audio ya que lo hace de forma inmediata por lo que se requiere menos tiempo de intervención de un técnico especializado para ajustar los parámetros de disparo de los chorros de agua. En este capítulo detallaremos la forma en que se desarrollo este paquete computacional y la manera en que debe ajustarse a un sistema de fuente o fuentes.

### 4.1 El Lenguaje Visual Basic

El lenguaje utilizado es Visual Basic ya que su fácil manejo y estructuración de código me permitió diseñar tanto la interfaz gráfica como el código de filtraje de una forma rápida y escalable. Es uno de los lenguajes más utilizados en la actualidad ya que permite integrar controles que facilitan la programación drásticamente así como el acceso directo con la API de Windows y otras herramientas que de igual manera facilitan la creación de software.

El sistema operativo empleado es Windows debido a que es un Sistema Operativo demasiado comercializado y debido a que la fuente debe ser lo más accesible y compatible. Este sistema tiene demasiadas debilidades respecto a

utilización de recursos físicos y virtuales pero aún así es aplicable a este sistema debido a que el código ha sido optimizado al máximo para evitar consumo de recursos excesivos.

Visual Basic es un lenguaje intérprete con la capacidad de controlar cada proceso o mensaje con los que Windows trabaja es por eso que ha permitido el enlace directo con las librerías multimedia de Windows y así controlar la reproducción de un archivo multimedia capturado con líneas de entrada y así manipular cada uno de sus datos generando espectrogramas que son leídos e interpretados con la finalidad de iniciar ciertos procesos en tiempos específicos del archivo de audio procesado.

Para el desarrollo del software que procesará las pistas musicales, se decidió emplear lenguaje Visual Basic por facilidad de uso y múltiples ventajas en cuanto a utilización de controles Activos (Active X). Tales controles permiten un fácil desarrollo de aplicaciones dirigidas a la reproducción de audio y al procesamiento digital. Otra de las grandes ventajas de este lenguaje intérprete es su potencial en cuanto a gráficos se refiere, el cual es necesario para cumplir con uno de los objetivos principales del sistema que es la creación de una interfaz amigable y de fácil operación. Se puede pensar que Visual Basic, por ser un lenguaje intérprete, es incapaz de dar un rendimiento elevado en aplicaciones desarrolladas con este lenguaje, pero contrario a esto, Visual Basic es capaz de comunicarse directamente con las funciones API (*Application Programming Interface* – Interfaz de Programación de Aplicación) de Windows y demás librerías DLL (*Dynamic Library Link* – Biblioteca de Enlace Dinámico) como lo haría un compilador más robusto como Visual C++. Así, es posible también manejar un procedimiento principal para el envío de mensajes a la cola de mensajes de Windows como lo hace Visual C++, logrando con ello, obtener los mismos resultados en aplicaciones específicas como ésta. De igual forma, es importante recalcar una enorme ventaja por cuanto a la relativa facilidad de creación estética de la interfaz de usuario, cabe decir, además, que si bien todo esto repercute

en los resultados obtenidos del desarrollo, el éxito dependerá en mayor parte de las habilidades del programador y en menor medida del lenguaje utilizado.

## 4.2 Interfaz Gráfica

La Interfaz gráfica siguiente está dirigida al únicamente al técnico de operación y servirá para tomar el control de la fuente coreográfica, esto es, ajustar los parámetros de disparo de la electroválvulas. Se explica a continuación su forma de uso y flexibilidad. Se anexa también una pequeña guía rápida de uso de cada control.



Figura 4.1 Ventana principal de la interfaz

I) **Tarjeta de Captura.**- La selección de la tarjeta de captura indicará a la interfaz la ubicación física de donde deberá obtener el audio a ser procesado.

II) **Línea de Entrada.**- El técnico podrá seleccionar la línea de entrada de audio, esto para tener más opciones de adquisición de audio. En la mayoría de los casos las

principales líneas de entrada serán las del micrófono y la línea de grabación principal de la computadora.

III) **Indicador de máxima amplitud.**- Estos indicadores solamente muestran los niveles máximos de audio que han sido obtenidos por medio de la tarjeta y línea de entrada seleccionada. Esta es útil para cuando se quiere llevar a cabo el ajuste adecuado del máximo volumen. La barras de volumen no deben de exceder los niveles máximos indicados en color rojo ya que indicarían saturación en la salida de audio y por ende una mala calidad de audio de salida, por el contrario los niveles deberán estar en color naranja o verde dependiendo del tipo de género que se este reproduciendo.

IV) **Control de volumen.**- Seguido de los pasos anteriores el usuario podrá seleccionar el porcentaje de volumen en la línea de entrada, esto para no exceder constantemente los límites últimos del ecualizador y así mantener las electroválvulas en niveles adecuados de disparo.

V) **Gráfico de frecuencias.**- Este gráfico muestra 24 barras de diferentes colores indicando la separación de varias frecuencias durante un análisis temporal del audio obtenido con la tarjeta de captura indicada. Estas frecuencias van desde 50 Hz hasta 2400 Hz, este rango de frecuencias no puede ser modificado por el técnico de operación debido a que es el único rango de frecuencias audibles para el ser humano.

VI) **Controles de configuración.**- Estos botones sirven para manipular la configuración registrada por un técnico de operación, ya que guarda o abre (botones etiquetados como guardar y abrir respectivamente ver Figura 4.1) un archivo de configuración anteriormente configurado e inicializa el patrón de disparo correspondiente. Es muy útil esta última operación ya que para configurar en tipo de género se requiere de un tiempo determinado y conocimientos específicos que deben

ser guardados para evitar pérdidas de tiempo con la reconfiguración de cada puesta en marcha del sistema.

Dentro de este conjunto de controles se encuentra un botón que iniciará o detendrá la captura de audio, esto después de haber configurado los parámetros de disparo pertinentes.

El control de modo cambia entre el modo de luces y válvulas, esto es, permite al usuario configurar el disparo independiente tanto de luces como de válvulas, cuando se presione este botón cambiará de forma automática la imagen que se encuentra en la parte superior derecha junto al control de simulación indicando con una válvulas que lo que el técnico está configurando en ese momento es el patrón de disparo pero únicamente para las válvulas si la imagen muestra una luz sub-acuática indicará entonces que el técnico está modificando el patrón de disparo de las luces.

VII) **Barras de ajuste.**- Estas barras de frecuencias estas colocadas en un orden específico iniciando con los sonidos graves, seguida de la parte media y por último las frecuencias altas de los agudos. Cuando se tengan seleccionadas las válvulas o luces a relacionar lo único que tendrá que hacer el usuario es seleccionar el porcentaje de la barra a relacionar y presionar el botón derecho del ratón en cada barra (llamadas también deslizables o slider). Seleccionando este porcentaje de activación le indicará el usuario al programa que cuando los valores de la señal de entrada en los rangos de esa frecuencia excedan el porcentaje indicado, activará las válvulas relacionadas con esa barra de frecuencia.

VIII) **Panel de Información.**- Este panel informa al usuario sobre algún cambio hecho durante la configuración del sistema. Le indicará al usuario de manera visual intermitente ya que aparecerá y desaparecerá durante algún proceso con los controles de usuario.

XI) **Imagen de modo.**- Indica el modo en el cual se encuentre en usuario este puede ser válvulas o luces. Las modificaciones que se lleven a cabo durante cada modo serán guardadas independientemente para cada modulo ya se válvulas o luces.

X) **Control de simulación.**- Esta ventana simulará la activación de cada válvula durante el funcionamiento del sistema. Se encenderán temporalmente de color azul unos pequeños círculos en este control indicando que las válvulas han sido activas y su posición indicará cual válvula fue la que se activó. Esto último se verá durante el funcionamiento del sistema pero cuando se encuentre en modo configuración y no haya sido iniciado le dará al usuario la posibilidad de seleccionar las válvulas o luces según sea el caso por medio de las válvulas dibujadas en el simulador, cada vez que seleccione una válvula o fuente se encenderá una luz de color verde que indicara que esta luz esta lista para ser relacionada con una barra de frecuencia especifica ubicada en la parte inferior de esta misma ventana.

Rango de Frecuencias de la barra de selección

1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	2	2	2	24
0	1	2	3	4	5	6	7	8	0	2	3	4	5	6	7	8	9	0	1	2	00
9	8	7	6	5	4	3	2	1	9	0	0	1	2	3	4	5	6	7	8	9	0.
.	.	.	.	.	.	.	.	.	0	0.	9.	8.	7.	6.	5.	4.	3.	2.	1	1.	2
2	3	4	5	6	7	8	9		1	2	3	4	5	6	7	8	9		1		

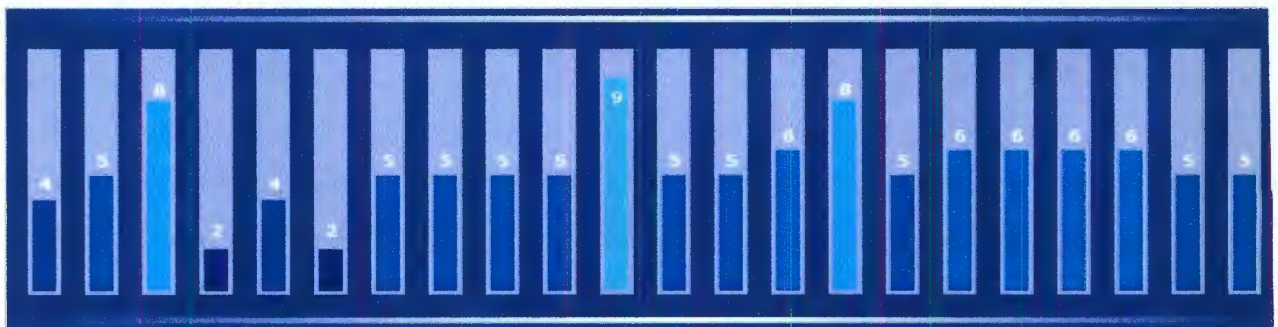


Figura 4.2 Barras selectoras de frecuencia (desde 50 Hz hasta 2.4 Khz)

XI) **Control de selección actual.**- Indica la cantidad de válvulas o luces seleccionadas durante el proceso de creación del patrón de disparo.

Las anteriores herramientas son las únicas herramientas que el técnico de operación deberá conocer para poder crear un patrón de disparo para cualquier tipo de género de música.

### 4.3 Metodología de desarrollo

Este componente del sistema se desarrolló en cuatro etapas diferentes con el objetivo de poder comprender mejor la metodología de elaboración. Estas etapas antes mencionadas son: Captura de Audio, Procesamiento, Activación y Visualización. Tal como se muestra en el siguiente diagrama

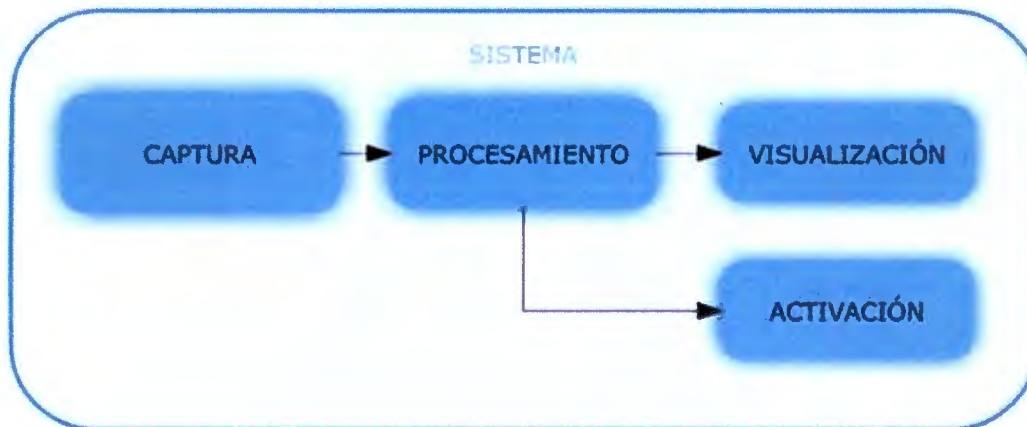


Figura 4.3 Metodología para el desarrollo del software de control

#### 4.3.1 CAPTURA

La parte más importante durante el desarrollo de software fue la implementación de una función capaz de calcular los valores de la transformada rápida de Fourier, y así poder conocer las señales que aparecen con más frecuencia durante el procesamiento de la señal. Fue de vital importancia que la cantidad de

cálculos realizados por la computadora fuera mínima para no forzar el procesador y optimizar el rendimiento de la computadora, así como tener una mejor respuesta en las electroválvulas y luces. De esta manera que el sistema no tendría que ser implantado en PC de extremo rendimiento y características especiales.

Para la lectura de los archivos de audio tipo WAV (Wave Audio Format) adquirido por la tarjeta de sonido y por medio de la línea de entrada, se requiere la estructura de archivos de audio de Windows predeterminada y a la cual se puede acceder mediante el uso de la API (*Application Programming Interface* – Interface de programación para aplicaciones) y utilizando las estructuras multimedia integradas de Windows.



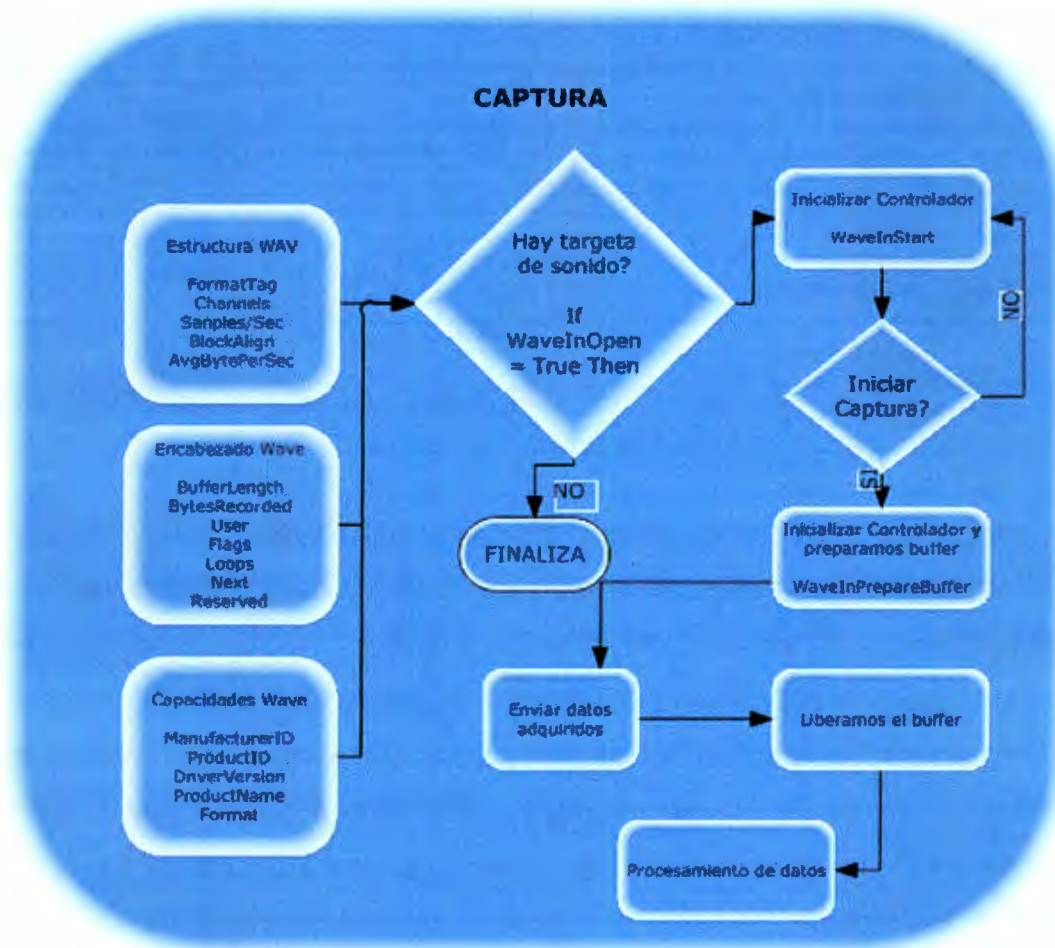


Figura 4.4 Diagrama a bloques para la etapa de captura de audio

Las siguientes son declaraciones de las librerías de la API para el manejo de pistas musicales con formato WAV, estas funciones permiten tanto la reproducción, pero , grabación como la administración de los datos internos de los archivos de audio. Es importante que el lector las analice para facilitar el entendimiento de la captura.

Tabla 4.1 Funciones utilizadas para la captura de audio

LIBRERÍA "WINMM"	
FUNCION	DESCRIPCION
<b>waveInAddBuffer</b>	Reserva memoria para el buffer encargado de gestionar el audio capturado
<b>waveInPrepareHeader</b>	Prepara el buffer para una entrada de audio tipo wav ( <i>Wave Format Audio</i> )
<b>waveInUnprepareHeader</b>	Libera el formato de tipo Wave del buffer

<b>waveInGetNumDevs</b>	Devuelve el número de tarjetas de sonido disponibles en la computadora
<b>waveInGetDevCaps</b>	Devuelve las características de cada tarjeta y sus métodos aceptados
<b>waveInOpen</b>	Abre o habilita un dispositivo de sonido para iniciar la captura de audio
<b>waveInClose</b>	Cierra o inhabilita un dispositivo de sonido y detiene la captura de audio
<b>waveInStart</b>	Inicia la captura de audio de la tarjeta de audio previamente seleccionada
<b>waveInReset</b>	Detiene temporalmente la captura de audio de la tarjeta seleccionada
<b>waveInStop</b>	Detiene por completo la captura de audio de la tarjeta actual seleccionada.

Con las funciones antes mencionadas se puede lograr obtener el audio que se esté reproduciendo en la tarjeta de sonido y a su vez se puede manipular libremente. Cabe mencionar que también se pueden emplear muchas más librerías para personalizar o agilizar el proceso de captura. A pesar de que la librería *winmm* es una librería de bajo nivel cuenta con funciones básicas que para nuestra investigación son perfectamente aplicables. Se recomendaría usar librerías de alto rendimiento si se decide modificar la metodología de captura a una de mayor velocidad. Se recomiendan las librerías DirectX por su gran flexibilidad y rendimiento.

Por otro lado, debido a que la captura se obtiene directamente de la tarjeta de sonido, se puede concluir que no es necesario conocer el formato de audio, ya que el software dedicado a la captura solo obtendrá muestras en intervalos de tiempo definidos (Ventaneo - Windowing), para este caso se realizará con una tasa de muestreo de 44.1 Khz.

#### 4.3.2 PROCESAMIENTO

En esta etapa se desarrolla el algoritmo más importante de esta investigación, el de procesamiento, debido a que de él dependen tanto la rutina de activación de las electroválvulas como el despliegado de las barras de frecuencias en la parte posterior

de la pantalla. Estas últimas relacionan cada una de las frecuencias a las cuales serán disparadas las electroválvulas. La siguiente figura ejemplifica el diseño interno de la etapa de procesamiento y la base del algoritmo.



Figura 4.5 Etapa de Procesamiento

Para la implementación de la función se toma en cuenta la función BIT REVERSE que invierte los bits de cada registro a procesar cambiando el bit de menor peso a la posición de mayor y viceversa. Para la implementación de esta función fue tomado en cuenta el libro *Numerical Recipes in C*, que muestra varias formas de la transformada de Fourier y se dan los créditos a Deeth por su mejora en rendimiento por cálculos innecesarios. En el siguiente fragmento de código podrá encontrar el código de la función BitReverse y se le colocó como código puro para que el lector pueda utilizarlo y hacer sus propias pruebas de software si así lo desea y le interesa el tema. [W. Press, 1988]

Tabla 4.2 Algoritmo bit reverse

```

Public Const AngleNumerator = 3.14159265358979 '6.2831852
Public Const NumSamples = 1024
Public Const NumBits = 10

Private ReversedBits(0 To NumSamples - 1) As Long

Sub DoReverse()

    Dim I As Long
    For I = LBound(ReversedBits) To UBound(ReversedBits)

```

```

    ReversedBits(l) = ReverseBits(l, NumBits)
Next
End Sub

Function ReverseBits(ByVal Index As Long, NumBits As Byte) As Long
    Dim l As Byte, Rev As Long

    For l = 0 To NumBits - 1
        Rev = (Rev * 2) Or (Index And 1)
        Index = Index \ 2
    Next

    ReverseBits = Rev
End Function

```

Seguido de la función BitReverse se encuentra la función primordial de nuestra investigación, la Transformada Rápida de Fourier que como ya se había comentado anteriormente sirve para analizar todas las frecuencias contenidas en una señal de audio. Esta función es también ampliamente usada en otro tipo de aplicaciones como imagen, biomedicina y video como muchas más.

**Tabla 4.3 Transformada de Fourier por métodos numéricos**

```

Sub FFTAudio(Realln() As Integer, RealOut() As Single)

    Static ImagOut(0 To NumSamples - 1) As Single

    Static l As Long, j As Long, k As Long, n As Long, BlockSize As Long, BlockEnd As Long
    Static DeltaAngle As Single, DeltaAr As Single
    Static Alpha As Single, Beta As Single
    Static TR As Single, Tl As Single, AR As Single, Al As Single

    For l = 0 To (NumSamples - 1)
        j = ReversedBits(l) 'I saved time here by pre-calculating all these values
        RealOut(j) = Realln(l)
        ImagOut(j) = 0 'Since this array is static, gotta make sure it's clear
        DoEvents
    Next

    BlockEnd = 1
    BlockSize = 2

    Do While BlockSize <= NumSamples
        DeltaAngle = AngleNumerator / BlockSize
        Alpha = Sin(0.5 * DeltaAngle)
        Alpha = 2! * Alpha * Alpha
        Beta = Sin(DeltaAngle)
    Loop

```

```
I = 0
Do While I < NumSamples
  AR = 1!
  AI = 0!

  j = I
  For n = 0 To BlockEnd - 1
    k = j + BlockEnd
    TR = AR * RealOut(k) - AI * ImagOut(k)
    TI = AI * RealOut(k) + AR * ImagOut(k)
    RealOut(k) = RealOut(j) - TR
    ImagOut(k) = ImagOut(j) - TI
    RealOut(j) = RealOut(j) + TR
    ImagOut(j) = ImagOut(j) + TI
    DeltaAr = Alpha * AR + Beta * AI
    AI = AI - (Alpha * AI - Beta * AR)
    AR = AR - DeltaAr
    j = j + 1
  DoEvents
Next

  I = I + BlockSize
  DoEvents
Loop

BlockEnd = BlockSize
BlockSize = BlockSize * 2
DoEvents
Loop
End Sub
```

Éste código también fue colocado como código explícito de VB con el objetivo de que pueda ser útil para aquellos lectores que deseen implementarla y mejorarla si es el caso. El algoritmo mostrado anteriormente fue desarrollado por **Ulli** y muchos otros autores, sin embargo se tuvo que modificar ligeramente a conveniencia para adecuarla a nuestro sistema. Cabe mencionar que el sistema no queda exento de muchas más mejoras por lo que sería agradable saber que hay lectores interesados que piensan mejorarla o desarrollar un proceso similar pero con mejor rendimiento. Hasta este punto se espera haya quedado definida parcialmente esta etapa así como sus partes esenciales.

### 4.3.3 ACTIVACIÓN

La comunicación del hardware con la PC se lleva a cabo por medio de una librería de enlace dinámico llamada IO.DLL (), la cual permite el uso del puerto paralelo en los sistema operativos más comunes como Windows, teniendo como ventaja la posibilidad de usarlo en sistemas NT como lo es XP. Permite la modificación de cada puerto ya sea el de datos, control o estado de una manera muy accesible y rápida.



Figura 4.6 Diagrama a bloques de utilización de IO.DLL

Las funciones principales que se utilizaron para enviar la matriz de disparo a la tarjeta de control son las que se muestran en el diagrama anterior, estas funciones tienen como parámetros básicos la dirección del puerto a utilizar y los valores a enviar en el caso de la función *portout*, ya que en las funciones de lectura solo se requiere la dirección del puerto y la función retornara el valor leído automáticamente. Es importante mencionar que esta librería funciona en el ambiente Windows XP lo que la hace más flexible ya que existen muchos problemas de permisos de usuarios cuando se quieren abrir el puerto paralelo en ambientes como XP, existen muchas más librerías libres pero no funcionan en XP, es por eso, que reiterando lo anterior hace de esta librería una de las librerías más potentes en este ámbito.

Se muestra en la siguiente figura un esquema detallado de la etapa de Activación.



Figura 4.7 Metodología de activación de válvulas y luces

Como se había mencionado anteriormente, se debe generar una librería de disparos para ser enviada a la tarjeta de control y sucesivamente a la tarjeta de potencia, esta subrutina o código de generación de la matriz de disparo se muestra a continuación. Hay que tomar en cuenta que este código es extremadamente lento y consume muchos recursos, y que dependerá completamente de un nuevo diseño para mejorar el rendimiento de la máquina que hospede este software, pero se coloca en la investigación para que pueda comprender la forma en que serán activadas las electroválvulas y la forma en que se discriminan las señales originales tomadas de la tarjeta de sonido.

Las matrices encargadas de almacenar los datos de referencia de las válvulas y luces son las matrices  $RefVal(24)$  y  $RefLuz(24)$  con 24 posiciones de memoria cada una, esto para, alcanzar el valor máximo de luces y válvulas que la fuente requiere.

En estas matrices se guardan las propiedades y las relaciones que tienen con las frecuencias adquiridas en turno, es decir, contienen los niveles tanto de amplitud como de frecuencias a los cuales dispararan sus válvulas correspondientes.

En la Figura 4.8 se aprecia de manera clara el diagrama de flujo con el que se registrará el software de control y con el cual podrá activar de manera adecuada las

válvulas y las luces del sistema. Como lo muestra la figura, el primer paso es comparar el modo o estado del sistema, esto es, si se encuentra en modo configuración o se encuentra capturando audio. Si no se encuentra capturando audio, el software deberá reconocer el modo de configuración en el que se encuentra, que podrían ser "Modo Válvulas" y "Modo Luces". Si se encuentra en el modo de configuración de válvulas el software espera a que el usuario elija una referencia para una o varias válvulas y después guarda el dato en una matriz de referencias que será usada durante la creación del patrón de disparos; por otro lado si fuese "Modo Luces" esperaría de igual manera al usuario para que fijara la referencia pero ahora de las luces para después guardar esa información en la matriz de referencia "RefLuz". Durante el desarrollo e implementación de esta función en Visual Basic se determinó que debía ser necesario mejorar la jerarquía total del software, esto debido a la prioridad requerida por cada proceso, era totalmente independiente y necesitaba ser priorizado con un espacio de memoria específico. Esto se logró por medio de funciones de windows (API – *Application Programming Interface*) destinadas al manejo de la prioridad y asignación de memoria del software, las cuales reducían o aumentaban la memoria del software interfaz sin dejar que otros paquetes actualmente ejecutados interfirieran con el rendimiento del paquete. Al proceso principal del paquete se le aplicaba una prioridad alta más no, prioridad de tiempo real para permitir que las funciones básicas de periféricos y ambiente de windows siguieran corriendo y no ocasionara una colisión con el núcleo de windows



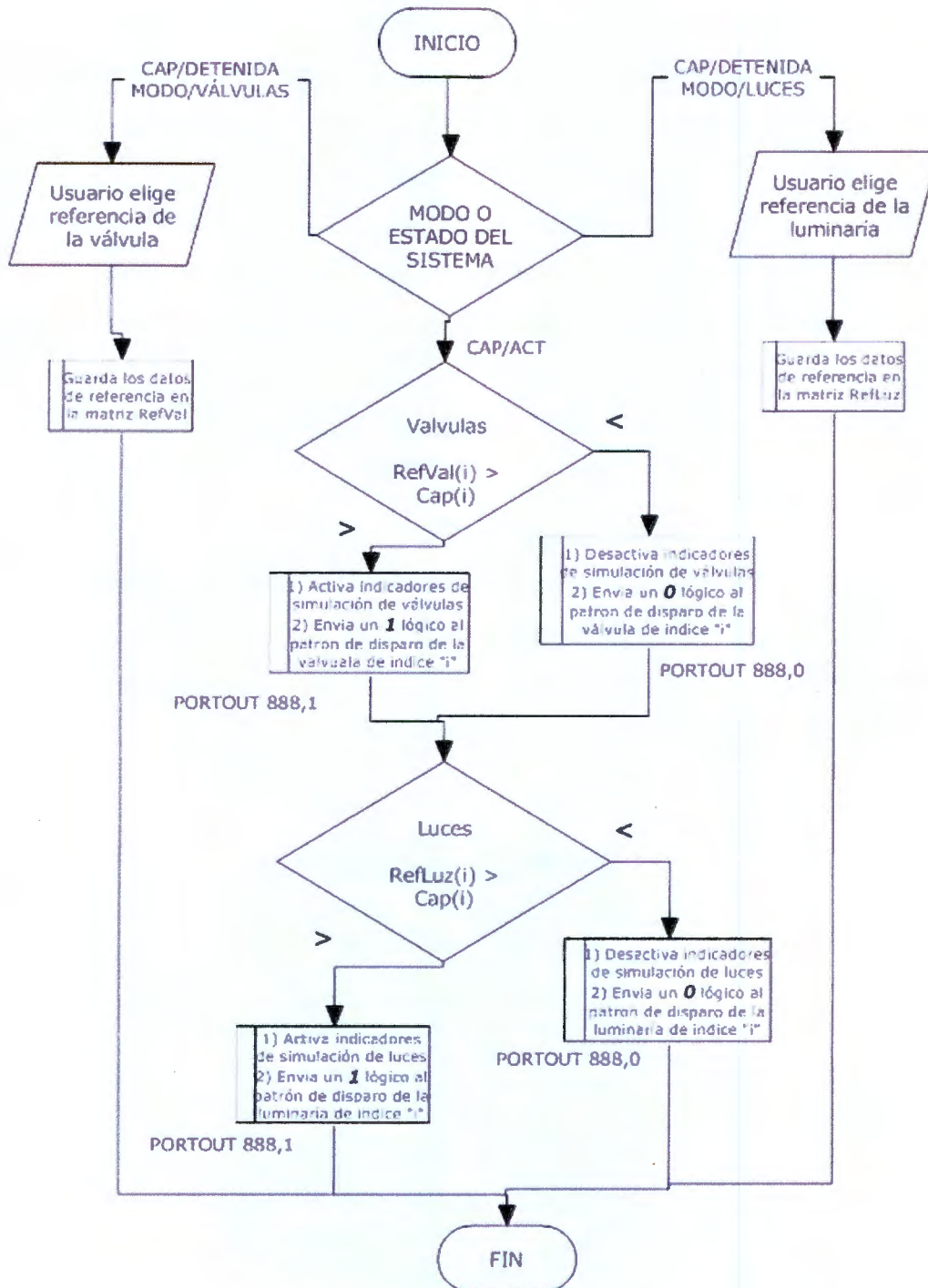


Figura 4.8 Diagrama funcional del simulador de válvulas y luces

Si se encontrará capturando audio, el proceso sería totalmente diferente, debido a que ahora se realizarán comparaciones con las matrices de referencia antes usadas

y las matrices generadas durante la captura de audio. Primero se comparan las matrices de referencia de válvulas y después de las luces; si la referencia fijada por el usuario ( $RefLuz(i)$ ) es mayor que la capturada, entonces, se tendrá que simular por software la activación de una válvula con índice  $i$  así como también se deberá activar de manera real la válvula con índice  $i$ , esto último se logrará creando una matriz de salida de activación denominada también en esta investigación "Patrón de Disparo". Cuando la matriz de activación esté lista se enviarán pulsos eléctricos hacia las válvulas tomando como referencia del "Patrón de Disparo" y usando funciones específicas de la librería IO.DLL como se muestra en el diagrama de flujo.

#### 4.3.4 VISUALIZACIÓN

En la siguiente figura se muestra la metodología de visualización en pantalla tanto del vúmetro (Indicador de máxima potencia de audio) y bandas de ecualización como el simulador de disparos (dibujo de la geometría de la fuente ubicado en la parte superior de la ventana principal).

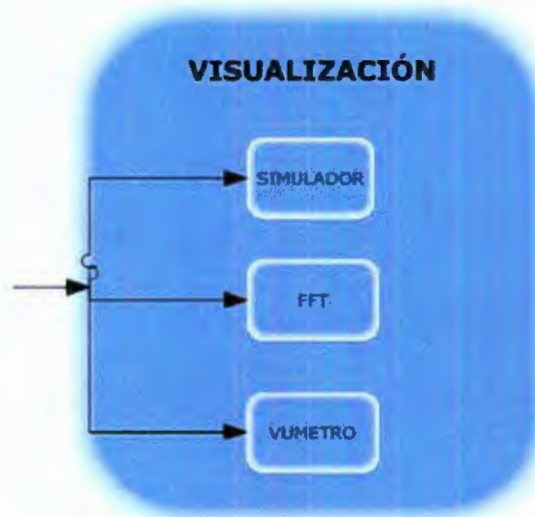


Figura 4.9 Metodología para la simulación

El siguiente código sirve para mostrar el simulador, vúmetro y barras de ecualización. Como se indica en el diagrama los datos son enviados desde la etapa de procesamiento y son utilizados en esta subrutina para dibujar las barras, vúmetro y simulador con funciones integradas de VB. Es importante mencionar que los gráficos se hacen haciendo referencia a la biblioteca de dibujos *GDI* de *API*.

**Tabla 4.4 Metodología para el manejo de archivos WAV con API de Windows**

Static X As Long	
Static archivowav As WaveHdr	//Declaración de Header WAV (PCM)
Static datosentrada(0 To Nummuestras - 1) As Integer	//Arreglo para captura de audio
Static datosalida(0 To Nummuestras - 1) As Single	//Arreglo para transformada de Fourier
Static NumCanales as Byte	
NumCanales=2	//Indica los canales usados
With ventana	//Eliminar redundancia de codigo
Do	//Inicia ciclo principal
archivowav.lpData = VarPtr(InData(0))	//Puntero para captura datos
archivowav.dwBufferLength = NumMuestras	//No. Muestras capturadas FFT 1023
Call waveInPrepareHeader(DevHandle, VarPtr(archivowav), Len(archivowav))	//Prepara buffer para captura
Call waveInAddBuffer(DevHandle, VarPtr(archivowav), Len(archivowav))	//Inicia la captura y envia los datos a lpData de la estructura archivowav
Do	//Ciclo anidado para espera de lectura
Loop Until ((archivowav.dwFlags And WHDR_DONE) = WHDR_DONE) Or DevHandle = 0	//Termina si acabó de enviar todo el buffer o si se desconecto la tarjeta de
	Captura o linea de entrada
If DevHandle = 0 Then Exit Do	//Si dispositivo no esta deshabilitado
Call waveInUnprepareHeader(DevHandle, VarPtr(archivowav), Len(archivowav))	//Libera el buffer para la siguiente //captura
Call FFT (datosentrada, datosalida)	//Los datos obtenidos envíalos a la FFT para obtener el espectro de frec.
'La función anterior envía las muestras de audio a la función FFT para obtener los datos de salida de la transformada de Fourier	
Call Barras(datosalida, NumMuestras, NumBarras)	//Actualiza las barras de Fourier que muestran las frecuencias de la pista
Call Vumetro (datoentrada, NumCanales)	// Actualiza las barras de volumen máximo mostradas en la parte inferior
Loop While DevHandle <> 0	//No termina ciclo mientras dispositivo de captura este disponible
End With	//termina rutina

En el código anterior se crea primeramente una estructura de tipo *WaveHeader* que contiene toda la información necesaria para el manejo de un archivo tipo *WAV PCM* de windows. Seguido de eso se prepara un buffer para capturar el audio proveniente del dispositivo de captura elegido por el usuario. Ese buffer es transmitido a un arreglo llamado *lpData* dentro de la estructura *waveheader* que es enviado a la función FFT (Transformada Rápida de Fourier). La función FFT nos regresará el espectro frecuencial de los datos capturados anteriormente para después ser desplegados a su vez en barras de frecuencias independientes (Gráfico de Fourier – Desplegado en la parte inferior izquierda del software interfaz). La penúltima línea de código nos indica que todo este proceso de captura y procesamiento para despliegue de gráficos básicos se llevará acabo mientras el dispositivo de audio permanezca activado y sin ningún error.

Las funciones usadas para este código de ejemplo, son de librerías de bajo rendimiento y por lo cual se sugiere aplicar funciones de mejor calidad u optimizar el código al máximo para evitar que la simulación y procesamiento de la señal de audio no intervenga en el rendimiento final durante la activación de las válvulas o luces, que es todo caso es lo más importante del sistema

#### 4.4 Interrupciones generadas por el panel de entrada

Durante los procesos de la pista musical el software de control de la fuente también ejecuta paralelamente una rutina que permite leer que pulsaciones son realizadas en el panel interactivo permitiendo así detener o iniciar nuevamente la captura del audio y por consiguiente la activación de las válvulas. Esta rutina realiza una lectura al puerto de estado con dirección 889 (0x379) y compara que botones en el panel han sido presionados, para así ejecutar la rutina correspondiente a la pulsación del panel. El pseudocódigo o metodología que permite capturar estas pulsaciones se muestra en la Figura 4.10. El código generado para la lectura de los botones fue colocado dentro de un control de tiempo, que leerá el estado de los

botones cada milisegundo. Este último proceso tiene una jerarquía baja dentro en el sistema, ya que, es más importante realizar los cálculos de procesamiento de audio y la generación del patrón de disparo de las válvulas que la lectura del panel.

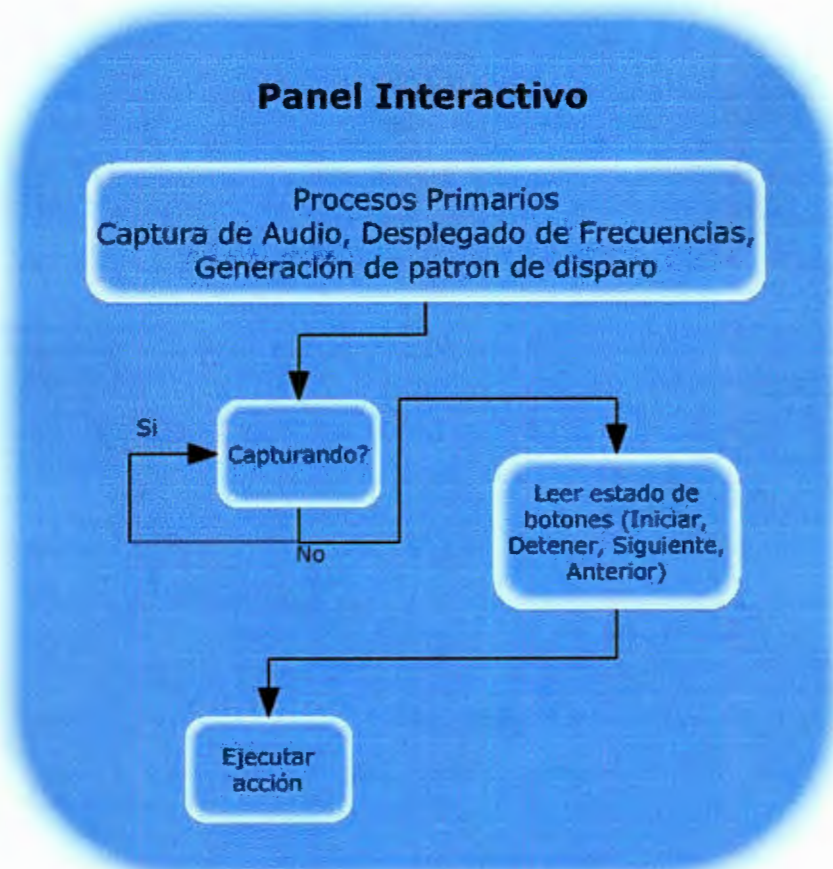


Figura 4.10 Metodología de lectura de panel de entrada del usuario final

## 5 - RESULTADOS

En este capítulo se desarrolló un esquema rápido que se ha presentado en capítulos anteriores y que describen los aspectos abordados a lo largo de la investigación realizada, los resultados obtenidos y su análisis. Todo ello se recoge en el compendio de temas que acompañan la investigación. Se construye aquí un hilo argumental que facilita la presentación del trabajo llevado a cabo, relacionándolo con los temas vistos, de manera que queda justificada nuevamente su unidad temática. La conexión entre el tema tratado en cada apartado y el desarrollo final se irá indicando, mediante referencias, a lo largo de todo el capítulo.

### 5.1 Caracterización del Software de control

El software final es capaz de controlar las válvulas en momentos adecuados sin ningún problema y crear una coreografía acuática adecuada y aceptable en la fuente. El software permite controlar el inicio y detención de captura de audio con el objetivo de permitir al técnico de operación, ajustar los parámetros que harán funcionar la fuente de manera correcta. El software desarrollado también fue capaz de guardar configuraciones preestablecidas con anterioridad y así poder ser cargadas nuevamente en cualquier momento, ahorrando así, tiempos innecesarios de reprogramación por parte del técnico y agilizando al mismo tiempo el proceso de puesta en marcha de la fuente.

Se desarrolló un software capaz de interactuar con el usuario de manera accesible y rápida. Aún cuando no se tengan conocimientos avanzados de informática el usuario final podrá manipular software, ya que solo tendrá que seleccionar la pista musical y presionar el botón que inicie la captura de audio (Ver videos de demostración en anexos).

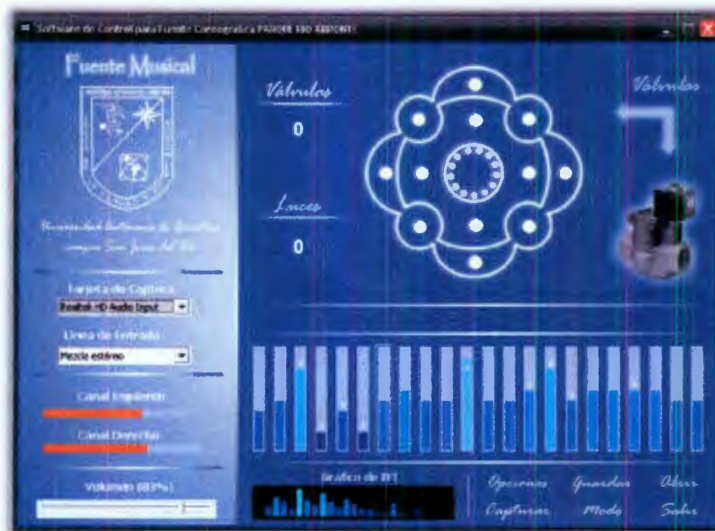


Figura 5.1 Imagen de software capturando pista musical

La imagen anterior muestra el objetivo principal de esta investigación, objetivo con el cual no se podría llevar a cabo la interacción de la fuente con la pista música y por el cual, evitando describir nuevamente las características esenciales de este paquete informático, se menciona el desempeño del mismo durante sus pruebas de validación.

El software fue capaz de procesar el audio capturado en la tarjeta de sonido de la computadora huésped, y sin ningún inconveniente, logró transformar estos datos a contenido frecuencial, con el objetivo de comparar ciertos niveles de amplitud dentro de varios rangos de frecuencia. El software también es capaz de desplegar esos datos frecuenciales en graficas de barras, que interpretan a su vez, los rangos de frecuencias en los cuales se descompone la pista musical. Cabe también, mencionar, que este paquete logra realizar la transformada rápida de Fourier en instantes de tiempo relativamente cortos, indicando que es capaz de procesar la información en tiempo real sin ningún inconveniente. Se descubrió una única desventaja respecto al procesamiento en tiempo real de la pista musical, esta es, que debido a retardos generados por el agua al caer o subir, se crea un asincronía cuando el software

manda la información en el tiempo correspondiente, si saber que el chorro de agua apenas está bajando, por lo que para corregir este problema, sería necesario implementar un retardo antes de activar la siguiente válvula.

## 5.2 Condiciones de las pruebas

Una parte esencial para el desarrollo de esta investigación ha sido también el plan de pruebas, realizado con el objetivo de conocer la factibilidad del proyecto. Se utilizó un hidroneumático, una bomba de medio caballo de potencia, un tubo de 2 pulgadas con 2 boquillas de prueba y la tubería adecuada, como se muestra en la Figura 5.2.

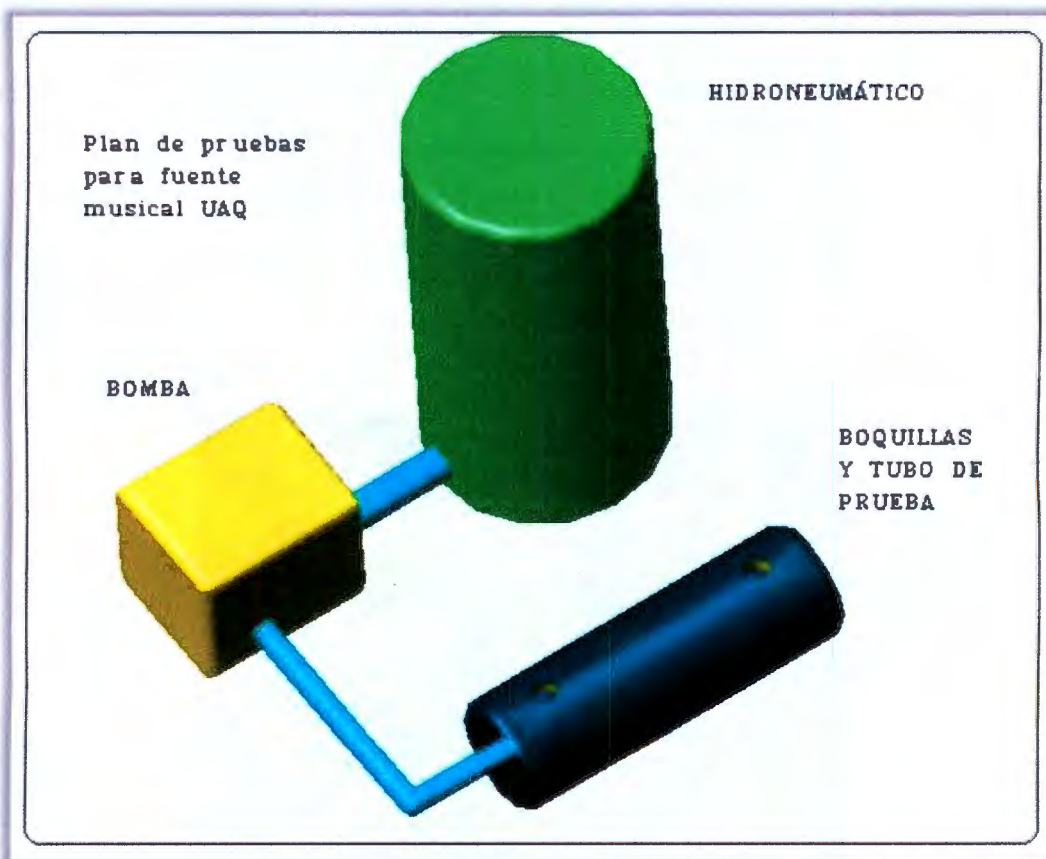


Figura 5.2 Diagrama básico de plan de pruebas



Durante las pruebas de campo, el hidroneumático utilizado suministró la cantidad de agua necesaria así como la presión para crear dos chorros de aguas de una altura aproximada de 3 metros, los cuales son adecuados y suficientes para el proyecto; por otro lado, la bomba también desempeñó un trabajo excelente debido a que aun cuando los chorros de agua se mantenían constantemente activos, la presión no disminuyó, y se mantuvo adecuadamente. Durante el desarrollo de las pruebas, se presentaron pequeñas fugas de agua las cuales vaciaron el tubo principal y ocasionaron un pequeño retardo al momento de ser activas, tomando en cuenta lo anterior se consideró el mantenimiento o la instalación adecuada para evitar este fenómeno cuando la fuente este en su etapa final y trabajando continuamente. Se colocaron dos boquillas diferentes para analizar el efecto del agua, una de las boquillas era de tipo aspersor y la otra de caudal máximo con diámetro de 2mm. De salida. Los dos efectos fueron validos y atractivos, por lo cual, se instalaran de los dos tipo en la fuente, la única diferencia es que la boquilla de caudal máximo genera una perdida más grande de presión que la de tipo aspersor, por esa razón, es conveniente combinar los tipos de boquillas para la fuente final. En la Figura 5.3 se muestra el tubo de pruebas así como las boquillas de diferente tipo, usadas durante las pruebas preliminares.



Figura 5.3 Tubo de pruebas y boquillas para el plan de pruebas preliminar

Es importante que durante la instalación de la tubería, se reduzcan al máximo las pérdidas por codos y diámetros extensos, así también, las fugas en acoplamientos y divisiones, este tipo de pérdidas durante las pruebas no fueron causales de falla debido a que solo se realizaron pruebas con una sola válvula y una sola rama de la tubería final, pero incrementando la cantidad de válvulas y tubería, es claro que algunas de las fallas antes mencionadas podrán repercutir de manera elevada al sistema completo.

Por otro lado, las pruebas de disparo realizadas desde la computadora fueron satisfactorias hasta cierto punto, esto debido a que, con la reproducción de pistas musicales con combinación de frecuencias características altas, bajas y medias, los disparos de las válvulas se mantenían por tiempo prolongados, generando un efecto de no sincronía con respecto a los chorros de agua y la pista musical, dicho de otra forma, cuando la pista contiene mucho efectos de bajo y voces continuas si pautas musicales adecuadas, las válvulas se mantienen siempre activas por que el patrón de disparo generado contiene frecuencias de disparo elevadas que sobrepasan la frecuencia máxima de apertura de las válvulas ocasionando que la válvula se quede siempre activada. Para entender el anterior fenómeno se muestra en la Figura 5.4, el software reproduciendo una pista musical de Rock Pesado (*Heavy Metal*), en la cual se puede apreciar claramente la saturación en la mayoría de los rangos de frecuencia.

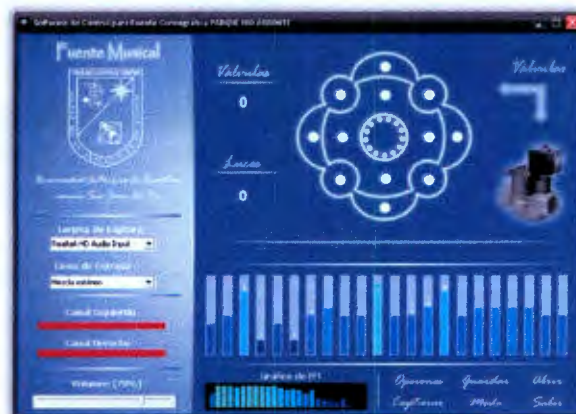


Figura 5.4 Software capturando con saturación de frecuencias

En la gráfica de Fourier situada en la parte inferior central se muestra la saturación de la mayoría de las bandas, así como también la saturación de niveles de audio (par de barras inferiores derechas). Por lo general, durante la reproducción de pistas musicales de este tipo de género, la saturación de frecuencias ocurre durante la mayor parte de la pista, y por ende, provoca que todas las válvulas permanezcan activadas durante un tiempo demasiado prolongado ocasionando que se genere esa no sincronía entre la pista y los chorros de agua.

Para entender un poco más este fenómeno se han obtenido las formas de onda de dos géneros musicales así como sus espectros frecuenciales, esto, con el objetivo de analizar profundamente la saturación de frecuencias en las pistas musicales. La primera gráfica en presentarse es del género "Música Clásica".

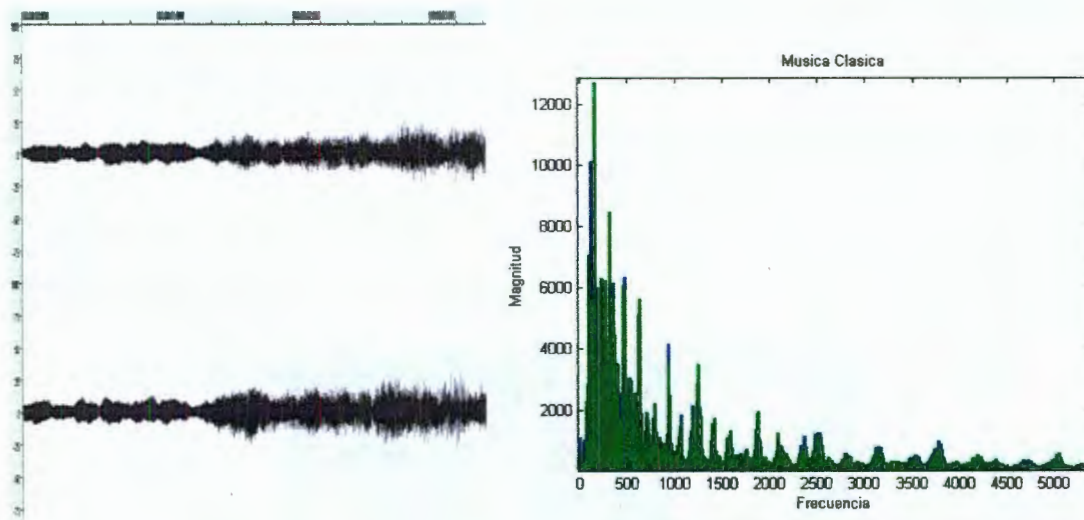
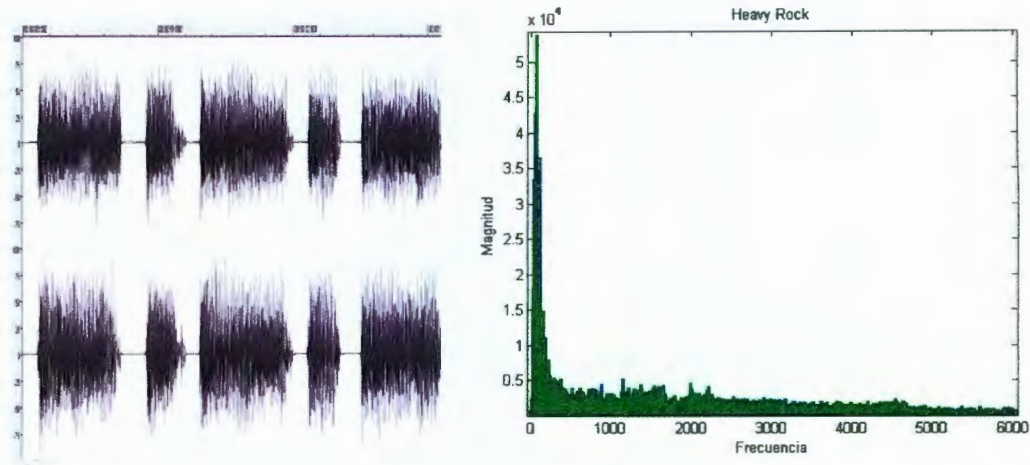


Figura 5.5 Análisis de pista musical de género "Música Clásica"

Seguido se presenta el análisis de una pista con alto contenido frecuencial de género "Rock Pesado - *Heavy Metal*", y capturado la misma cantidad de tiempo, por lo que el análisis de ambas pistas musicales se limita únicamente a los primeros 5

segundos. Es importante hacer la comparación en los mismos tiempos para resaltar las diferencias estructurales de cada pista musical sin confundirse con otros tiempos de cada pista.



**Figura 5.6 Análisis de pista musical de género “Rock Pesado – Heavy Metal”**

Las graficas que nos muestra la señal de las pistas musicales en el dominio del tiempo no nos muestran de manera clara la diferencia respecto a frecuencias entre cada pista musical sino que solamente nos permite apreciar un incremento alto en los niveles de la señal, sin embargo no es nuestro campo de análisis por lo que seguiremos con el análisis Fourier y sus espectros de frecuencias. En la primera grafica “Música Clásica” se aprecian varios picos de frecuencia separados por espacios proporcionales y con incrementos de magnitud logarítmicos, que nos indican un orden entre la aparición de cada frecuencia así como también un espacio diminuto pero suficiente de tiempo durante la aparición de cada frecuencias diferente. En cuanto a la segunda grafica de Fourier “Rock Pesado”, se puede apreciar un amontonamiento diminuto pero equivalente de todas las frecuencias en la parte inferior de la grafica, indicándonos así, que los cambios de una frecuencia a otra se presentan de una manera casi instantánea, provocándonos la permanente activación de las válvulas y el efecto de chorros de agua permanentes que no parecen sincronizarse con la pista musical.

Para evitar este efecto, es necesaria, la implementación de un retardo durante el procesamiento de cada tramo de audio analizado o mejor aún, el análisis completo de la pista musical antes de la reproducción final, esto, con el objeto de generar el patrón de disparo completo y aplicarle el arrastre de tiempo necesario (retardo o *shifting*) mucho antes de la reproducción. La mejora anterior evita que por falta de potencia de procesamiento o rendimiento de hardware, se presenten retardos inadecuados que causen fallas durante la activación de las válvulas, así como también, evita sobrepasar la frecuencia máxima de disparo de las válvulas corrigiendo el patrón de disparo y el orden las válvulas a activar.

## 6 - CONCLUSIONES Y PROSPECTIVAS

El principal logro obtenido a razón de esta investigación ha sido la creación de una interfaz por software que permite al usuario interactuar con una fuente musical hidráulica, y lograr interpretar una señal análoga en efusiones de agua acordes a esta señal, que en nuestro caso en específico, es una señal de audio. Este objetivo fue realizado satisfactoriamente y con resultados positivos.

Con el presente proyecto se ha conseguido también la instrumentación final de las tarjetas de control y de potencia de manera satisfactoria que permitirán la comunicación e interpretación de la pista musical en efusiones de agua, así mismo se ha logrado el desarrollo de un paquete computacional dedicado al procesamiento de una señal analógica adquirida por medio de la línea de entrada de una tarjeta de sonido cualquiera, adaptada a una PC.

Por medio de procedimientos matemáticos como la transformada de Fourier y otros mencionados en la investigación, enfocados claro, al procesamiento digital de audio, se ha conseguido obtener un espectrograma de cualquier pista musical que sea reproducida en el sistema, sin importar el formato ni el reproductor, y pudiéndose así, filtrar ciertos datos digitales para la selección de los parámetros que regirán la activación de las electroválvulas y luces en ciertos instantes específicos de la melodía en turno.

Se diseñó también, un prototipo funcional del panel interactivo que iniciará o detendrá la captura de la pista musical. Fue probado y se obtuvieron resultados satisfactorios en cuanto a desgaste y funcionalidad, esto referido, principalmente a la correcta activación de los sensores, traduciéndose en mantenimiento mínimo y reducción de costos.

Una ventaja obtenida de este sistema es la gran capacidad de interacción del usuario con el software, ya que al obtener la señal análoga directamente de la línea de entrada de la tarjeta de sonido (16 bits máx., incluso es posible la retroalimentación de la salida de audio de la PC a la línea de entrada) se podrá obtener cualquier reproducción dentro de la PC, evitando el problema de los formatos y tasas de muestreo. Incluso cualquier usuario puede utilizar su propio reproductor personal, conectarlo a la línea y generar su propia secuencia de disparo con la pista musical de su elección; esto después de haberse configurado los parámetros de filtrado de audio por el usuario técnico.

Como un sistema no perfecto esta por supuesto propenso a mejoras, mejoras que a través de esta investigación se pudieron analizar y compilar. Colocando al componente de software como uno de los pilares más importantes y objetivo principal de esta investigación la mayoría de las mejoras se le realizarían a este componente.

Una de las mejoras importantes sería la implementación de una clase más, que permitiera generar un retardo en el software y así compensar la inercia del agua al salir de las electroválvulas. Este proceso requerido es comúnmente llamado *Shifting* o arrastre en señales, que a su vez, forma parte de uno de los procesos más usados en los componentes DSP.

Otra de las mejoras sería la implementación de un nuevo modulo en el programa que lo hiciera más accesible, permitiendo al usuario, analizar por completo toda la pista musical antes de reproducirla, y así, evitar o compensar automáticamente los retardo del agua. Esta opción es muy aplicable a este diseño siempre y cuando el usuario tenga en cuenta que, no podrá esperar la máxima sincronía entre los chorros de agua y la pista musical, cuando quiera reproducir pistas con saturación de frecuencia o, dicho de otra manera, pistas con una elevada cantidad de ruido.

Como complemento de la anterior mejora, estaría por último la pre-coreografía de la pista musical. Esta funcionalidad es la más comercial, pero sin embargo, la más limitada, debido a que se requiere de gran esfuerzo para generar los parámetros de disparo de las válvulas, así como de conocimientos específicos en cuanto al software y el hardware de la fuente. Este procedimiento es complicado porque los parámetros de disparo son configurados cada cierto lapso de la pista musical, y por lo cual, requiere de un tiempo excesivo para su configuración sin dejar de mencionar, que limita de manera determinada la cantidad de géneros que se pueden reproducir en la fuente.

Pensando a futuro, también se pretende agregar una lista de reproducción y mejorar la manera de interactuar con los controles, haciendo que el usuario únicamente arrastre la válvula o luz hacia la barra de frecuencia destinada y mejore la sensibilidad de disparo de las electroválvulas y luces. Y a su vez se pueda optimizar el procesamiento de la señal, reduciendo la cantidad de memoria utilizada durante estas operaciones y modificando el código fuente para obtener mejores resultados, orientados a la reducción de tiempos de proceso.



## 7 - BIBLIOGRAFÍA

Alan V. Oppenheim, ***Signals and Systems***, Prentice Hall.

Albert Boggess & Francis J. Narcowich, ***A First Course in wavelets with Fourier Analysis***, Prentice Hall.

Enrique Palacios Municio, Fernando Remiro Domínguez, Lucas J. López Pérez, ***Microcontrolador PIC16F84 Desarrollo de Proyectos***, Primera Edición, 2004, AlfaOmega.

F. Alton Everest, ***Master Handbook Of Acoustics***, Fourth Edition, MacGraw-Hill, 2001

Jan Axelson, ***Parallel Port Complete***, Lakeview Research ,2000.

Murray R. Spiegel, 1974, ***Fourier Analysis with application to Boundary Value Problems***, McGrawHill.

Ogata Kat Suhiko, ***Ingeniería de Control Moderna***. 3ª Ed., 1998, Prentice Hall.

Paul M. Embree, 1995, ***C Algorithms for Real-Time DSP***, Prentice Hall.

Richard G. Lyons, 1998, ***Understanding Digital Signal Processing***, Prentice Hall.

Robert L. Boylestad, 2004, ***Introducción al Análisis de Circuitos***, Décima Edición, Prentice Hall.

Stephan M. Bernsee, ***The DFT "à Pied": Mastering The Fourier Transform in One Day*** , Septiembre 21, 1999, <http://www.dspdimension.com>.

Steven W. Smith, ***Digital Signal Processing***, 1981.

Vijay K. Madisetti-Douglas B. Williams, ***Digital Signal Processing HandBook*** , 1999, CRC Press.

Vinay K. Ingle, John G. Proakis, ***Digital Signal Processing using MatLab V. 4***, PWS Publishing Company.

W. Press, S. Teukolsky, W. Vetterling, B. Flannery, ***Numerical Recipes in C'***, Cambridge University Press, 1988/92.

[www.wikipidia.com](http://www.wikipidia.com), recopilación de antecedentes sobre Fuentes Musicales.

**8 -ANEXOS**

## 8.1 ANEXO 1 – MANUAL DE USUARIO DEL SOFTWARE

Este **GUIA RÁPIDA** le ayudará a reconocer rápidamente cada control incluido en el software de la fuente así como también lo guiará paso a paso para configurar los parámetros básicos para la coreografía de cualquier pista musical adquirida por medio de la PC del sistema. En la siguiente figura se muestra la ventana principal del software de control de la fuente y sus partes principales.

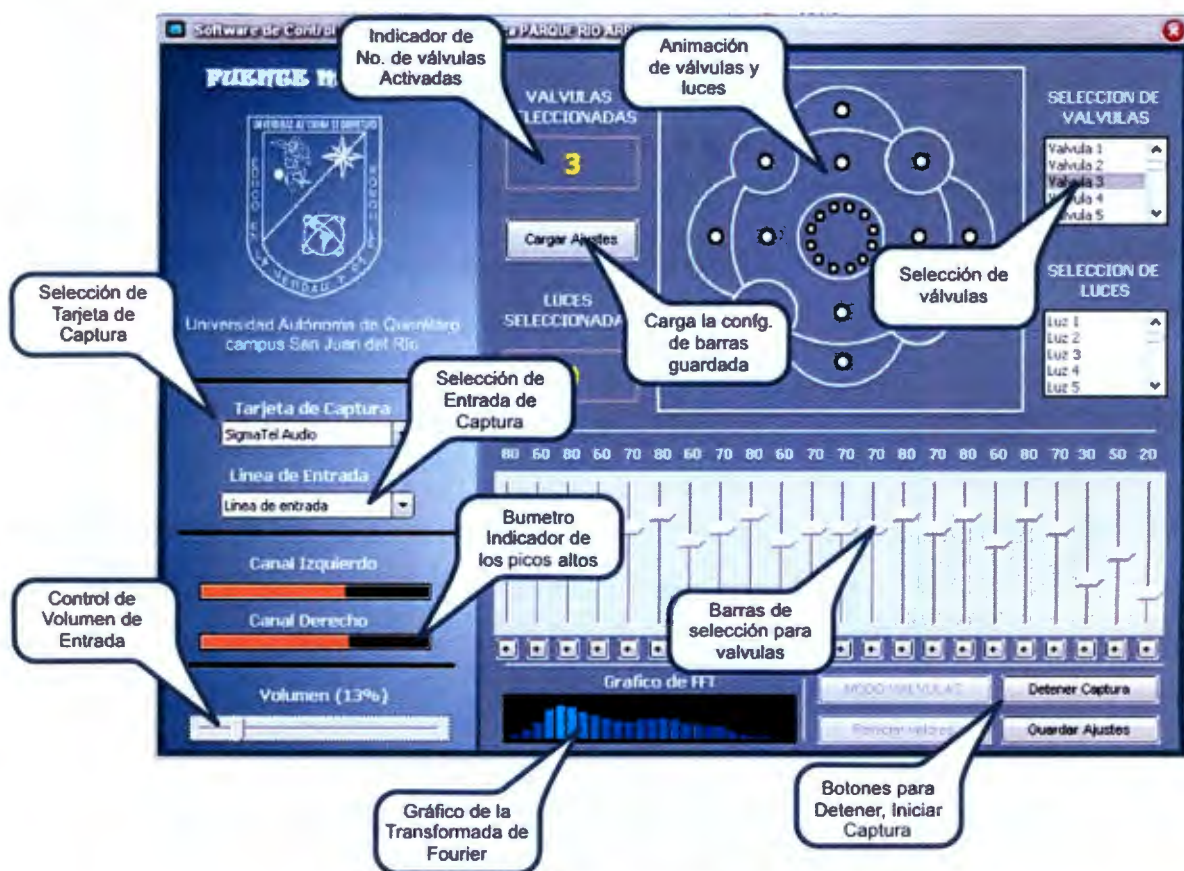


Figura 8.1 Explicación de funciones de interfaz - Software

**Selección de tarjeta de captura:** Permite seleccionar el recurso de hardware que se usará para capturar el audio que será procesado y enviado hacia las electroválvulas.

**Selección de estrada de Audio:** A diferencia del anterior componente este parámetro ajusta el componente de software que se encargará de capturar el audio este puede ser micrófono, mezcla estéreo o entrada principal.

**Control de Volumen de Entradas:** Este parámetro permite ajustar el volumen con el que se capturara el audio, este parámetro es muy importante y se debe considerar antes de capturar el audio ya que si lo ajustamos en niveles muy altos o erróneos los disparos de algunas electroválvulas puede nunca presentarse, esto debido a que si existe saturación en los canales de audio los valores de disparo siempre estarán al 100% y puede ocasionar que algunas electroválvulas ajustadas a niveles mas bajos que este nunca disparen, sin embargo se permite al usuario ajustar este parámetro debido a que algunas tarjetas de audio envían el audio muy atenuado por lo que en estos casos si es necesario aumentar al máximo el nivel máximo de volumen para captura.

**Bumetro:** Este indicador muestra el nivel de audio en dB que esta siendo capturado

**Gráfico de transformada de Fourier:** Muestra como su nombre lo dice el espectro en frecuencia del audio capturado por medio de la transformada de Fourier, este componente es muy útil para conocer los niveles de Graves, Medios y Agudos durante la captura.

**Botones para detención, inicio y pausa de captura:** Estos botones como su nombre lo indican inician la captura o la detienen según sea el caso.

**Barras de selección para válvulas:** estas barras sirven para ajustar el valor aproximado de los disparos de las electroválvulas, esto por medio de un porcentaje que indica el nivel mínimo de audio necesario para activar la electroválvula, cuando el nivel de audio sobrepase este nivel ajustado en estas barras la electroválvula asignada a este rango se disparará y de igual forma lo hará con las demás válvulas.

**Indicador de No. de válvulas seleccionadas:** Este indicador muestra las válvulas activas que serán ajustadas, no todas las válvulas se pueden ajustar al mismo tiempo y es por eso que se muestran solo las válvulas activas a las que se le ajustará el parámetro de disparo.

**Carga de configuración de barras:** Cuando se presiona este botón los parámetros modificados en las barras de niveles se cargan en las válvulas activas

**Selector de válvulas:** Este selector permite seleccionar un conjunto de válvulas con el objetivo de modificarles los parámetros de disparo con las barras de configuración antes mencionadas.

**Animación de válvulas y luces:** Consiste en una animación que muestra las válvulas y luces de forma interactiva, esto es, cuando enciende una electroválvula o luz físicamente en la fuente en esta animación aparecen de color rojo las válvulas activadas respectivas.

## 8.2 ANEXO 2 - HOJA DE DATOS 74HC573

## INTEGRATED CIRCUITS

## DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

**74HC/HCT573**

Octal D-type transparent latch;  
3-state

Product specification  
File under Integrated Circuits, IC06

December 1990

Philips  
Semiconductors



**PHILIPS**

## Octal D-type transparent latch; 3-state

## 74HC/HCT573

## FEATURES

- Inputs and outputs on opposite sides of package allowing easy interface with microprocessors
- Useful as input or output port for microprocessors/microcomputers
- 3-state non-inverting outputs for bus oriented applications
- Common 3-state output enable input
- Functionally identical to the "563" and "373"
- Output capability: bus driver
- $I_{CC}$  category: MSI

## GENERAL DESCRIPTION

The 74HC/HCT573 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT573 are octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state outputs for bus oriented applications.

A latch enable (LE) input and an output enable ( $\overline{OE}$ ) input are common to all latches.

The "573" consists of eight D-type transparent latches with 3-state true outputs. When LE is HIGH, data at

the  $D_n$  inputs enter the latches. In this condition the latches are transparent, i.e. a latch output will change state each time its corresponding D-input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE.

When  $\overline{OE}$  is LOW, the contents of the 8 latches are available at the outputs. When  $\overline{OE}$  is HIGH, the outputs go to the high impedance OFF-state. Operation of the  $\overline{OE}$  input does not affect the state of the latches.

The "573" is functionally identical to the "563" and "373", but the "563" has inverted outputs and the "373" has a different pin arrangement.

## QUICK REFERENCE DATA

GND = 0 V;  $T_{amb} = 25^\circ\text{C}$ ;  $t_r = t_f = 8\text{ ns}$

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
$t_{PHL}$ $t_{PLH}$	propagation delay	$C_L = 15\text{ pF}$ ; $V_{CC} = 5\text{ V}$			
	$D_n$ to $Q_n$		14	17	ns
	LE to $Q_n$		15	15	ns
$C_I$	input capacitance		3.5	3.5	pF
$C_{PD}$	power dissipation capacitance per latch	notes 1 and 2	26	26	pF

## Notes

1.  $C_{PD}$  is used to determine the dynamic power dissipation ( $P_D$  in  $\mu\text{W}$ ):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

$f_i$  = input frequency in MHz;  $f_o$  = output frequency in MHz

$\sum (C_L \times V_{CC}^2 \times f_o)$  = sum of outputs

$C_L$  = output load capacitance in pF;  $V_{CC}$  = supply voltage in V

2. For HC the condition is  $V_i = \text{GND to } V_{CC}$ ; for HCT the condition is  $V_i = \text{GND to } V_{CC} - 1.5\text{ V}$

## ORDERING INFORMATION

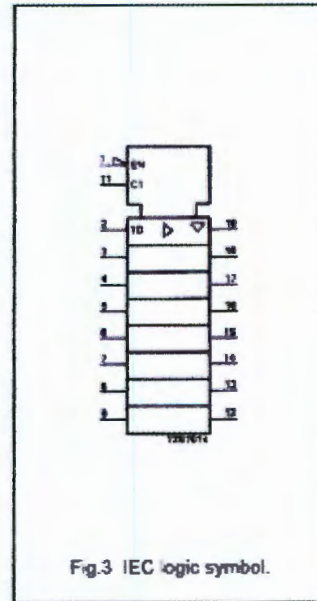
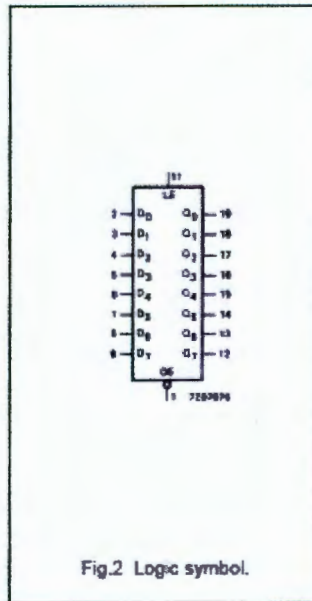
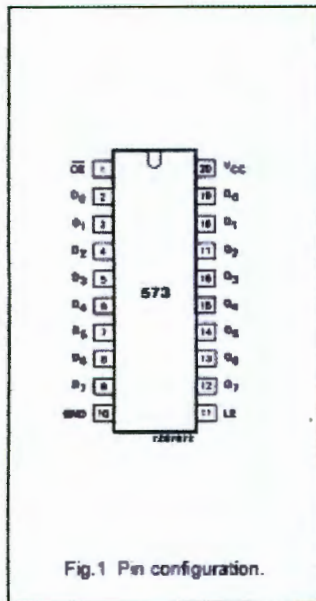
See "74HC/HCT/HCU/HCMOS Logic Package Information".

## Octal D-type transparent latch; 3-state

74HC/HCT573

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
2, 3, 4, 5, 6, 7, 8, 9	$D_0$ to $D_7$	data inputs
11	LE	latch enable input (active HIGH)
1	OE	3-state output enable input (active LOW)
10	GND	ground (0 V)
19, 18, 17, 16, 15, 14, 13, 12	$Q_0$ to $Q_7$	3-state latch outputs
20	$V_{CC}$	positive supply voltage





## Octal D-type transparent latch; 3-state

74HC/HCT573

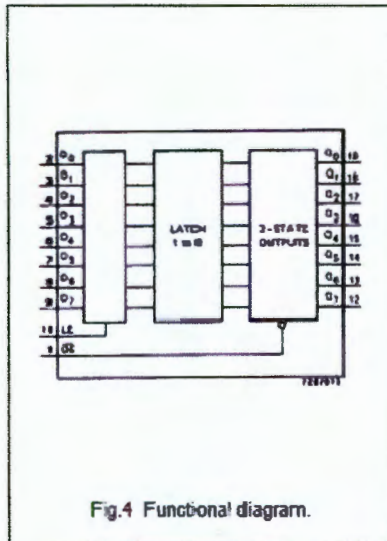


Fig.4 Functional diagram.

## FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS Q <sub>0</sub> to Q <sub>7</sub>
	$\overline{OE}$	LE	D <sub>N</sub>		
enable and read register (transparent mode)	L	H	L	L	L
	L	H	H	H	H
latch and read register	L	L	l	L	L
	L	L	h	H	H
latch register and disable outputs	H	L	l	L	Z
	H	L	h	H	Z

## Notes

- H = HIGH voltage level  
h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition  
L = LOW voltage level  
l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition  
Z = high impedance OFF-state

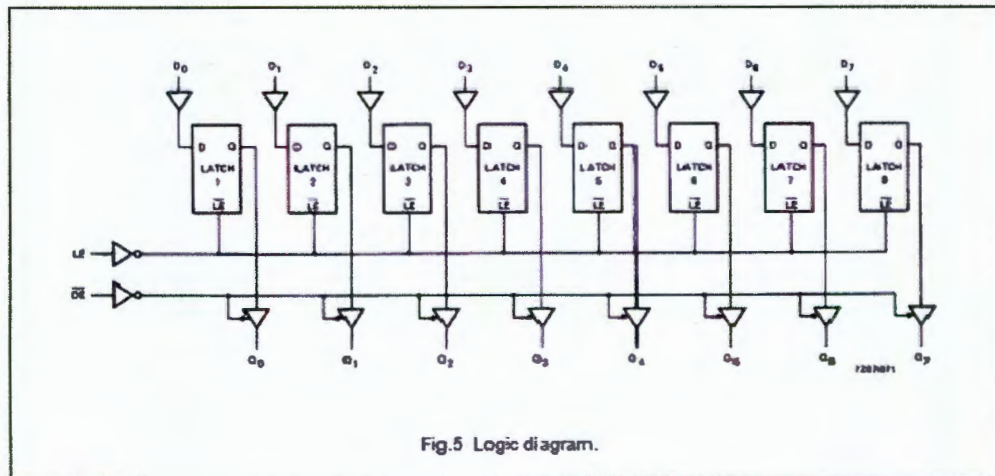


Fig.5 Logic diagram.

## Octal D-type transparent latch; 3-state

74HC/HCT573

## DC CHARACTERISTICS FOR 74HC

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: bus driver

 $I_{CC}$  category: MSI

## AC CHARACTERISTICS FOR 74HC

GND = 0 V;  $t_r = t_f = 6$  ns;  $C_L = 50$  pF

SYMBOL	PARAMETER	$T_{amb}$ (°C)						UNIT	TEST CONDITIONS		
		74HC							$V_{CC}$ (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
$t_{P-L} / t_{P-LH}$	propagation delay D <sub>n</sub> to Q <sub>n</sub>	47 17 14	150 30 28		100 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6	
$t_{P-L} / t_{P-LH}$	propagation delay LE to Q <sub>n</sub>	50 18 14	150 30 28		100 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7	
$t_{PZH} / t_{PZL}$	3-state output enable time OE to Q <sub>n</sub>	44 16 13	140 28 24		175 35 30		210 42 36	ns	2.0 4.5 6.0	Fig.8	
$t_{PZH} / t_{PZL}$	3-state output disable time OE to Q <sub>n</sub>	55 20 16	150 30 28		100 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.8	
$t_{THL} / t_{TLH}$	output transition time	14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig.6	
$t_N$	enable pulse width HIGH	80 18 14	14 5 4		100 20 17		120 24 20	ns	2.0 4.5 6.0	Fig.7	
$t_{SU}$	set-up time D <sub>n</sub> to LE	50 10 9	11 4 3		65 13 11		75 15 13	ns	2.0 4.5 6.0	Fig.9	
$t_H$	hold time D <sub>n</sub> to LE	5 5 5	3 1 1		5 5 5		5 5 5	ns	2.0 4.5 6.0	Fig.9	

## Octal D-type transparent latch; 3-state

74HC/HCT573

## DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: bus driver

I<sub>CC</sub> category: MSI

## Note to HCT types

The value of additional quiescent supply current ( $\Delta I_{CC}$ ) for a unit load of 1 is given in the family specifications. To determine  $\Delta I_{CC}$  per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
D <sub>n</sub>	0.35
LE	0.65
$\overline{OE}$	1.25

## AC CHARACTERISTICS FOR 74HCT

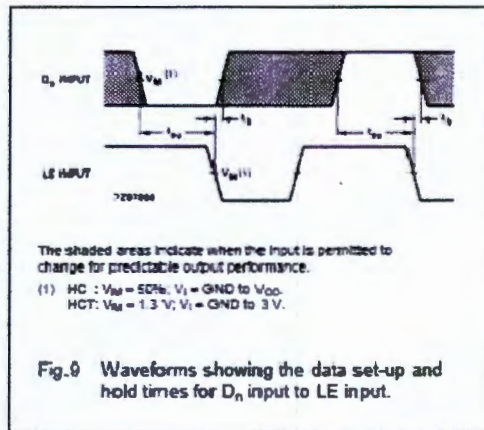
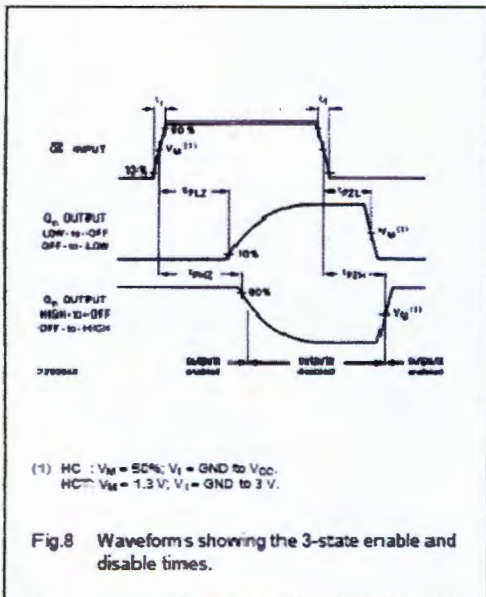
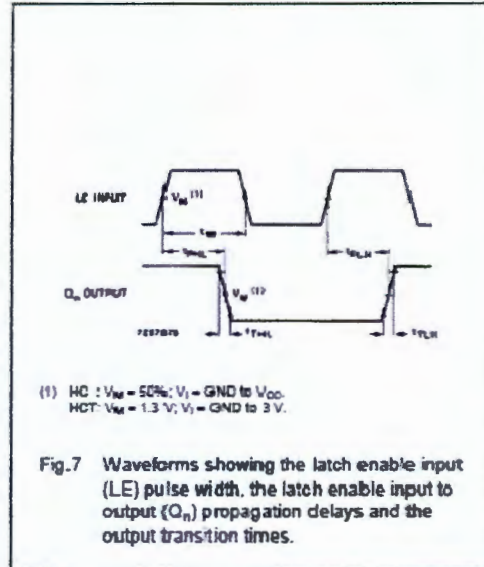
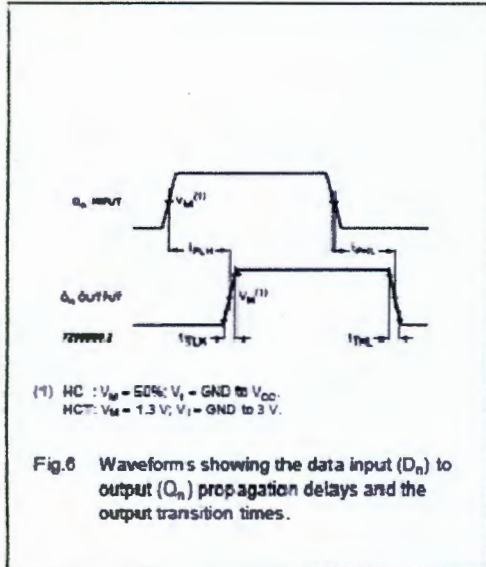
GND = 0 V; t<sub>r</sub> = t<sub>f</sub> = 6 ns; C<sub>L</sub> = 50 pF

SYMBOL	PARAMETER	T <sub>amb</sub> (°C)						UNIT	TEST CONDITIONS		
		74HCT							V <sub>CC</sub> (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t <sub>PHL</sub> / t <sub>PLH</sub>	propagation delay D <sub>n</sub> to Q <sub>n</sub>		20	35		44		53	ns	4.5	Fig.6
t <sub>PHL</sub> / t <sub>PLH</sub>	propagation delay LE to Q <sub>n</sub>		18	35		44		53	ns	4.5	Fig.7
t <sub>ZHL</sub> / t <sub>ZL</sub>	3-state output enable time $\overline{OE}$ to Q <sub>n</sub>		17	30		38		45	ns	4.5	Fig.8
t <sub>PHZ</sub> / t <sub>PLZ</sub>	3-state output disable time $\overline{OE}$ to Q <sub>n</sub>		18	30		38		45	ns	4.5	Fig.8
t <sub>THL</sub> / t <sub>TLH</sub>	output transition time		5	12		15		18	ns	4.5	Fig.8
t <sub>w</sub>	enable pulse width HIGH	16	5		20		24		ns	4.5	Fig.7
t <sub>su</sub>	set-up time D <sub>n</sub> to LE	13	7		16		20		ns	4.5	Fig.9
t <sub>h</sub>	hold time D <sub>n</sub> to LE	9	4		11		14		ns	4.5	Fig.9

## Octal D-type transparent latch; 3-state

74HC/HCT573

## AC WAVEFORMS



## PACKAGE OUTLINES

See "74HC/HCT/HCU/HCMOS Logic Package Outlines".

## 8.3 ANEXO 2 – HOJA DE DATOS 74HC238

## INTEGRATED CIRCUITS

## DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

## 74HC/HCT238

### 3-to-8 line decoder/demultiplexer

Product specification  
File under Integrated Circuits, IC06

December 1990

Philips  
Semiconductors



# PHILIPS

## 3-to-8 line decoder/demultiplexer

## 74HC/HCT238

## FEATURES

- Demultiplexing capability
- Multiple input enable for easy expansion
- Ideal for memory chip select decoding
- Active HIGH mutually exclusive outputs
- Output capability: standard
- I<sub>CC</sub> category: MSI

## GENERAL DESCRIPTION

The 74HC/HCT238 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT238 decoders accept three binary weighted address inputs (A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub>) and when enabled,

provide 8 mutually exclusive active HIGH outputs (Y<sub>0</sub> to Y<sub>7</sub>).

The "238" features three enable inputs: two active LOW ( $\bar{E}_1$  and  $\bar{E}_2$ ) and one active HIGH (E<sub>3</sub>). Every output will be LOW unless  $\bar{E}_1$  and  $\bar{E}_2$  are LOW and E<sub>3</sub> is HIGH.

This multiple enable function allows easy parallel expansion of the "238" to a 1-of-32 (5 lines to 32 lines) decoder with just four "238" ICs and one inverter.

The "238" can be used as an eight output demultiplexer by using one of the active LOW enable inputs as the data input and the remaining enable inputs as strobes. Unused enable inputs must be permanently tied to their appropriate active HIGH or LOW state.

The "238" is identical to the "138" but has non-inverting outputs.

## QUICK REFERENCE DATA

GND = 0 V; T<sub>amb</sub> = 25 °C; t<sub>r</sub> = t<sub>f</sub> = 8 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t <sub>PHL</sub> , t <sub>PLH</sub>	propagation delay	C <sub>L</sub> = 15 pF; V <sub>CC</sub> = 5 V			
	A <sub>n</sub> to Y <sub>n</sub>		14	18	ns
	E <sub>3</sub> to Y <sub>n</sub>		16	20	ns
	$\bar{E}_n$ to Y <sub>n</sub>		17	21	ns
C <sub>I</sub>	input capacitance		3.5	3.5	pF
C <sub>PD</sub>	power dissipation capacitance per package	notes 1 and 2	72	76	pF

## Notes

1. C<sub>PD</sub> is used to determine the dynamic power dissipation (P<sub>D</sub> in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f<sub>i</sub> = input frequency in MHz

f<sub>o</sub> = output frequency in MHz

∑ (C<sub>L</sub> × V<sub>CC</sub><sup>2</sup> × f<sub>o</sub>) = sum of outputs

C<sub>L</sub> = output load capacitance in pF

V<sub>CC</sub> = supply voltage in V

2. For HC the condition is V<sub>I</sub> = GND to V<sub>CC</sub>  
For HCT the condition is V<sub>I</sub> = GND to V<sub>CC</sub> - 1.5 V

## ORDERING INFORMATION

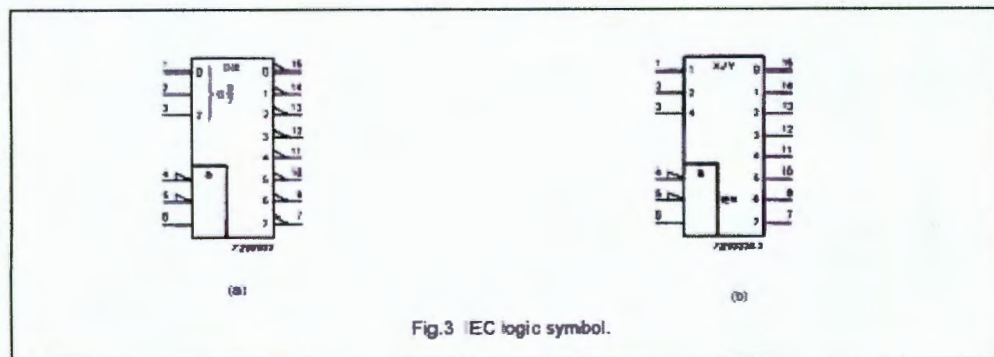
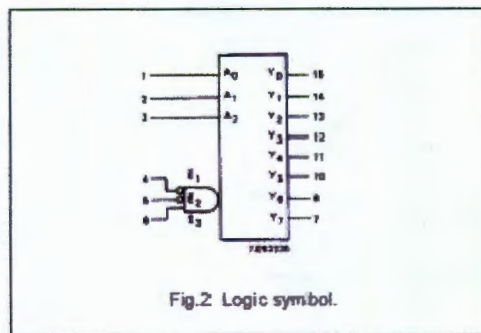
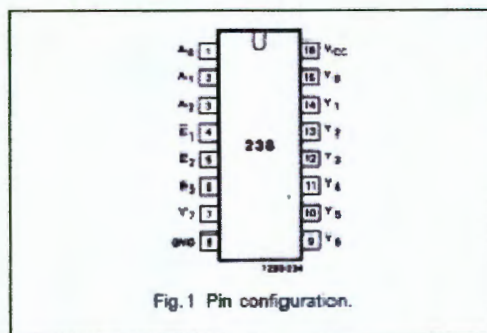
See "74HC/HCT/HCU/HCMOS Logic Package Information".

## 3-to-8 line decoder/demultiplexer

74HC/HCT238

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 2, 3	$A_0$ to $A_2$	address inputs
4, 5	$\bar{E}_1, \bar{E}_2$	enable inputs (active LOW)
6	$E_3$	enable input (active HIGH)
8	GND	ground (0 V)
15, 14, 13, 12, 11, 10, 9, 7	$Y_0$ to $Y_7$	outputs (active HIGH)
16	$V_{CC}$	positive supply voltage



3-to-8 line decoder/demultiplexer

74HC/HCT238

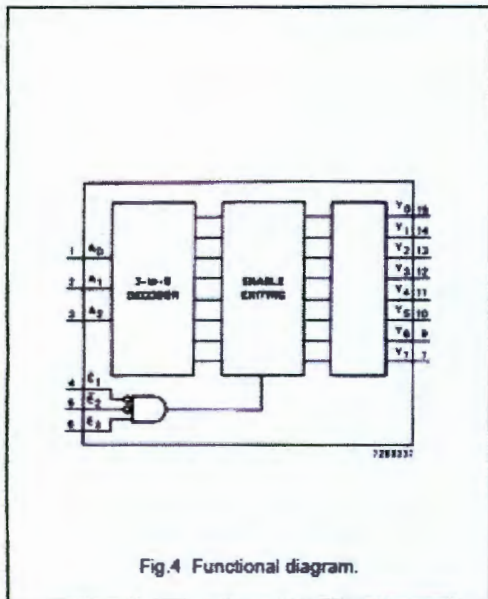


Fig.4 Functional diagram.

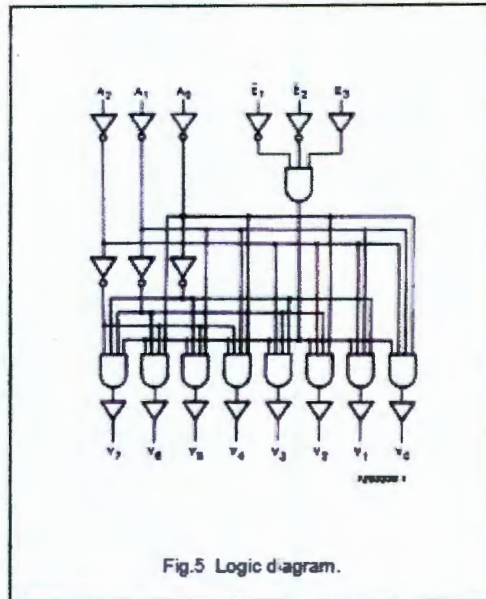


Fig.5 Logic diagram.

FUNCTION TABLE

INPUTS						OUTPUTS							
E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>
H	X	X	X	X	X	L	L	L	L	L	L	L	L
X	H	X	X	X	X	L	L	L	L	L	L	L	L
X	X	L	X	X	X	L	L	L	L	L	L	L	L
L	L	H	L	L	L	H	L	L	L	L	L	L	L
L	L	H	H	L	L	L	H	L	L	L	L	L	L
L	L	H	L	H	L	L	L	H	L	L	L	L	L
L	L	H	H	H	L	L	L	L	H	L	L	L	L
L	L	H	L	L	H	L	L	L	L	H	L	L	L
L	L	H	L	H	H	L	L	L	L	L	H	L	L
L	L	H	H	H	H	L	L	L	L	L	L	H	L
L	L	H	H	H	H	L	L	L	L	L	L	L	H

Note

- 1. H = HIGH voltage level
- L = LOW voltage level
- X = don't care



## 3-to-8 line decoder/demultiplexer

74HC/HCT238

## DC CHARACTERISTICS FOR 74HC

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: standard

I<sub>CC</sub> category: MSI

## AC CHARACTERISTICS FOR 74HC

GND = 0 V; t<sub>r</sub> = t<sub>f</sub> = 8 ns; C<sub>L</sub> = 50 pF

SYMBOL	PARAMETER	T <sub>amb</sub> (°C)						UNIT	TEST CONDITIONS		
		74HC							V <sub>CC</sub> (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t <sub>PHL</sub> /t <sub>PLH</sub>	propagation delay A <sub>n</sub> to Y <sub>n</sub>		47 17 14	150 30 28		100 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t <sub>PHL</sub> /t <sub>PLH</sub>	propagation delay E <sub>3</sub> to Y <sub>n</sub>		52 19 15	160 32 27		200 40 34		240 48 41	ns	2.0 4.5 6.0	Fig.6
t <sub>PHL</sub> /t <sub>PLH</sub>	propagation delay E <sub>n</sub> to Y <sub>n</sub>		50 18 14	155 31 26		105 30 33		235 47 40	ns	2.0 4.5 6.0	Fig.7
t <sub>THL</sub> /t <sub>TLH</sub>	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Figs 6 and 7

## 3-to-8 line decoder/demultiplexer

## 74HC/HCT238

## DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: standard

$I_{CC}$  category: MSI

## Note to HCT types

The value of additional quiescent supply current ( $\Delta I_{CC}$ ) for a unit load of 1 is given in the family specifications. To determine  $\Delta I_{CC}$  per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
$A_n$	0.70
$\bar{E}_n$	0.40
$E_3$	1.45

## AC CHARACTERISTICS FOR 74HCT

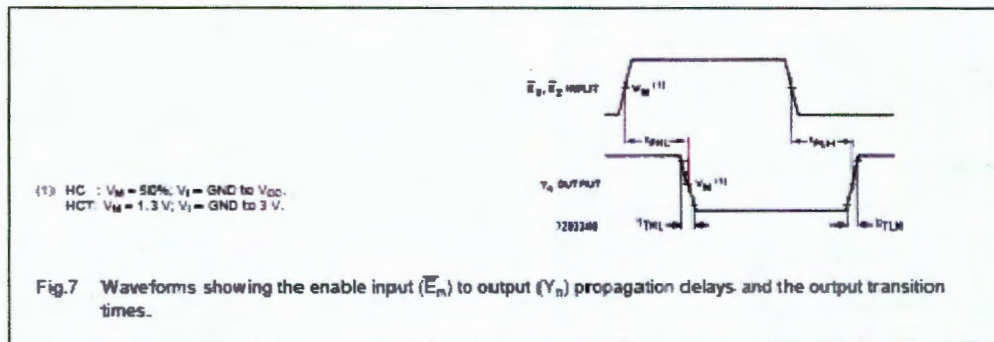
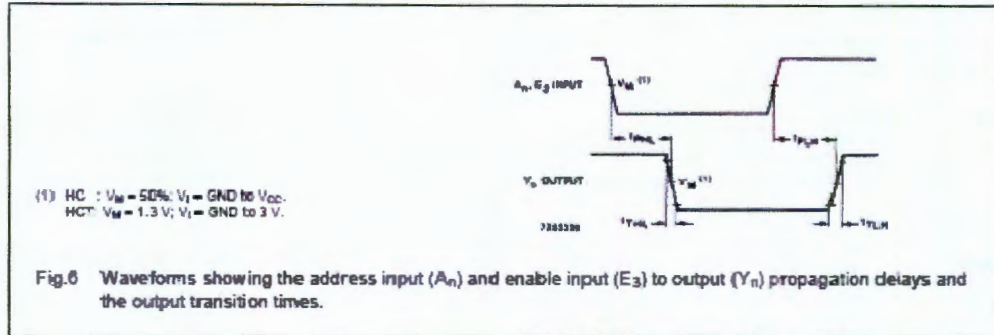
GND = 0 V;  $t_r = t_f = 6$  ns;  $C_L = 50$  pF

SYMBOL	PARAMETER	$T_{amb}$ (°C)						UNIT	TEST CONDITIONS		
		74HCT							$V_{CC}$ (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
$t_{pHL}$	propagation delay $A_n$ to $Y_n$		21	35		44		53	ns	4.5	Fig.6
$t_{pLH}$	propagation delay $A_n$ to $Y_n$		17	35		44		53	ns	4.5	Fig.6
$t_{pHL}$	propagation delay $E_3$ to $Y_n$		22	37		46		56	ns	4.5	Fig.6
$t_{pLH}$	propagation delay $E_3$ to $Y_n$		18	37		46		56	ns	4.5	Fig.6
$t_{pHL}$	propagation delay $\bar{E}_n$ to $Y_n$		21	35		44		53	ns	4.5	Fig.7
$t_{pLH}$	propagation delay $\bar{E}_n$ to $Y_n$		18	35		44		53	ns	4.5	Fig.7
$t_{rHL} / t_{rLH}$	output transition time		7	15		19		22	ns	4.5	Figs 6 and 7

## 3-to-8 line decoder/demultiplexer

74HC/HCT238

## AC WAVEFORMS



## PACKAGE OUTLINES

See "74HC/HCT/HCU/HCMOS Logic Package Outlines".

## 8.4 ANEXO 3 – HOJA DE DATOS 74HC245

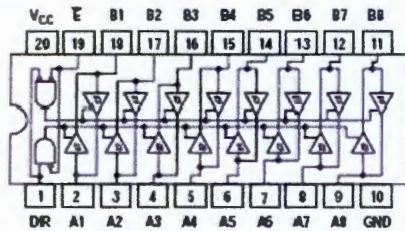
### SN74LS245

#### Octal Bus Transceiver

The SN74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input ( $\bar{E}$ ) can be used to isolate the buses.

- Hysteresis Inputs to Improve Noise Immunity
- 2-Way Asynchronous Data Bus Communication
- Input Diodes: Limit High-Speed Termination Effects
- ESD > 3500 Volts

#### LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)



#### TRUTH TABLE

INPUTS		OUTPUT
E	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Immaterial

#### GUARANTEED OPERATING RANGES

Symbol	Parameter	Min	Typ	Max	Unit
$V_{CC}$	Supply Voltage	4.75	5.0	5.25	V
$T_A$	Operating Ambient Temperature Range	0	25	70	°C
$I_{OH}$	Output Current – High			-3.0	mA
				-15	mA
$I_{OL}$	Output Current – Low			24	mA



**ON Semiconductor**  
Formerly a Division of Motorola  
<http://onsemi.com>

**LOW  
POWER  
SCHOTTKY**



PLASTIC  
N SUFFIX  
CASE 738



SOIC  
DW SUFFIX  
CASE 751D

#### ORDERING INFORMATION

Device	Package	Shipping
SN74LS245N	16 Pin DIP	1440 Units/Box
SN74LS245DW	16 Pin	2500/Tape & Reel

## SN74LS245

## DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
$V_{IL}$	Input LOW Voltage			0.8	V	Guaranteed Input LOW Voltage for All Inputs
$V_{T+}-V_{T-}$	Hysteresis	0.2	0.4		V	$V_{CC} = \text{MIN}$
$V_{IK}$	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{IN} = -10 \text{ mA}$
$V_{OH}$	Output HIGH Voltage	2.4	3.4		V	$V_{CC} = \text{MIN}$ , $I_{OH} = -3.0 \text{ mA}$
		2.0			V	$V_{CC} = \text{MIN}$ , $I_{OH} = \text{MAX}$
$V_{OL}$	Output LOW Voltage		0.25	0.4	V	$I_{OL} = 12 \text{ mA}$ , $V_{CC} = V_{CC} \text{ MIN}$ , $V_{IN} = V_{IL} \text{ or } V_{IH}$ per Truth Table
			0.35	0.5	V	$I_{OL} = 24 \text{ mA}$
$I_{OZH}$	Output Off Current HIGH			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{OUT} = 2.7 \text{ V}$
$I_{OZL}$	Output Off Current LOW			-200	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{OUT} = 0.4 \text{ V}$
$I_{IH}$	Input HIGH Current	A or B, DR or E		20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{IN} = 2.7 \text{ V}$
		DR or E		0.1	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 7.0 \text{ V}$
		A or B		0.1	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 5.5 \text{ V}$
$I_{IL}$	Input LOW Current			-0.2	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 0.4 \text{ V}$
$I_{OS}$	Output Short Circuit Current (Note 1)	-40		-225	mA	$V_{CC} = \text{MAX}$
$I_{CC}$	Power Supply Current Total, Output HIGH			70	mA	$V_{CC} = \text{MAX}$
				90		
				95		

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

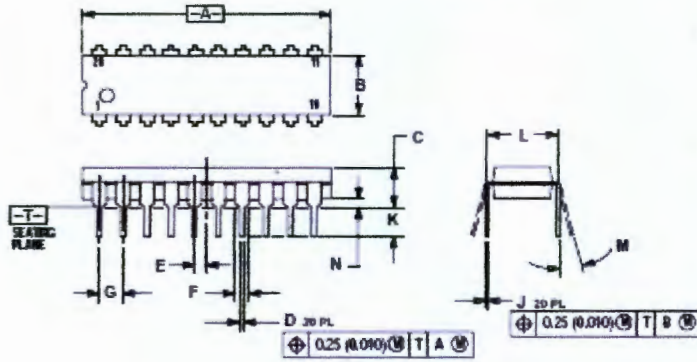
AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 5.0 \text{ V}$ ,  $T_{RISE}/T_{FALL} \leq 5.0 \text{ ns}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$t_{PLH}$	Propagation Delay, Data to Output		8.0	12	ns	$C_L = 45 \text{ pF}$ , $R_L = 667 \Omega$
$t_{PHL}$			8.0	12		
$t_{RZH}$	Output Enable Time to HIGH Level		25	40	ns	
$t_{RZL}$	Output Enable Time to LOW Level		27	40	ns	
$t_{PLZ}$	Output Disable Time from LOW Level		15	25	ns	
$t_{PHZ}$	Output Disable Time from HIGH Level		15	25	ns	$C_L = 5.0 \text{ pF}$ , $R_L = 667 \Omega$

# SN74LS245

## PACKAGE DIMENSIONS

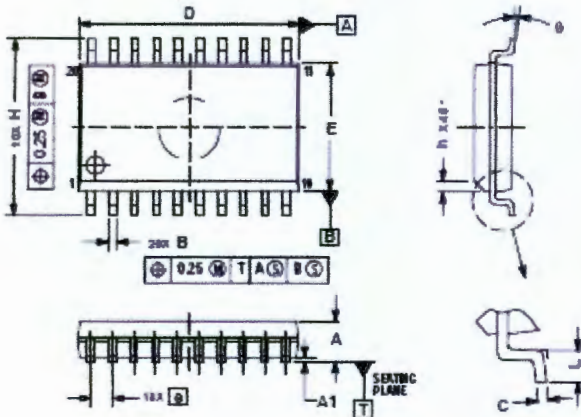
### N SUFFIX PLASTIC PACKAGE CASE 738-03 ISSUE E



- NOTES:
1. DIMENSIONS AND TOLERANCING PER ANSI Y14.9M, 1982.
  2. CONTROLLING DIMENSION: INCH.
  3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
  4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.070	1.270	25.85	32.11
B	0.240	0.260	6.10	6.60
C	0.150	0.190	3.81	4.82
D	0.015	0.027	0.39	0.68
E	0.000 BSC 1.27 BSC			
F	0.060	0.070	1.27	1.77
G	0.100 BSC 2.54 BSC			
J	0.005	0.015	0.21	0.38
K	0.150	0.240	3.80	6.09
L	0.200 BSC 5.08 BSC			
M	0	15	0	37
N	0.025	0.240	0.64	6.10

### D SUFFIX PLASTIC SOIC PACKAGE CASE 751D-05 ISSUE F



- NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
  2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.9M, 1994.
  3. DIMENSIONS D AND E DO NOT INCLUDE MOLD PROTRUSION.
  4. MAXIMUM MOLD PROTRUSION 0.15 PER SIDE.
  5. DIMENSION B DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE PROTRUSION SHALL BE 0.13 TOTAL IN EXCESS OF B DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS	
	MIN	MAX
A	2.35	2.65
A1	0.30	0.25
B	0.15	0.40
C	0.23	0.32
D	12.55	0.95
E	7.40	7.60
H	1.27 BSC	
J1	10.00	10.55
K	0.25	0.35
L	0.50	0.90
S	0	7

## 8.5 ANEXO 4 – HOJA DE DATOS GP2A22

SHARP

GP2A20/GP2A22

## GP2A20/GP2A22

Light Modulation, Long Focal  
Distance Type OPIC  
Photointerrupter

## ■ Features

1. Light modulation type, free from external disturbing light
2. Long focal distance type  
Detecting range  
(GP2A20: 3 to 7mm)  
(GP2A22: 9 to 15mm)
3. Capable of TTL direct connection
4. With 3-pin connector provided for easier interface with peripheral control circuit

## ■ Applications

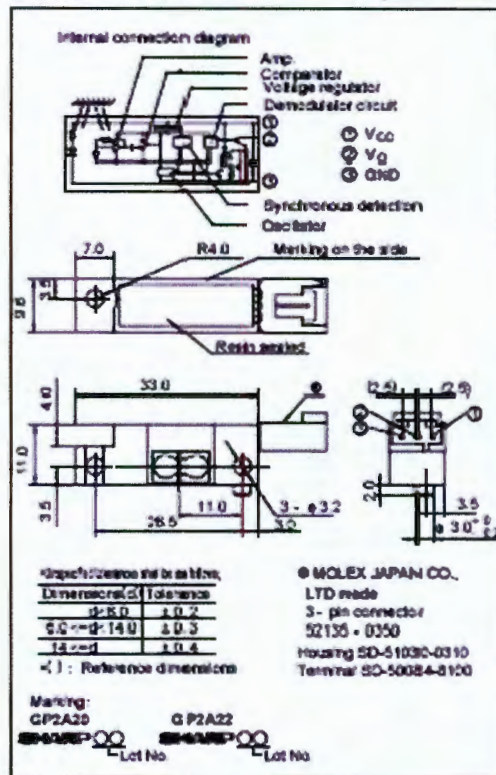
1. Copiers
2. Laser beam printers
3. Facsimiles

## ■ Line-ups

Model No.	Detecting range	
	3 to 7mm	9 to 15mm
	GP2A20	GP2A22

## ■ Outline Dimensions

(Unit: mm)



OPIC™ (Optical IC) is a trademark of the SHARP Corporation.  
An OPIC consists of a light-detecting element and signal-processing circuit integrated onto a single chip.

## ■ Absolute Maximum Ratings (Ta= 25°C)

Parameter	Symbol	Rating	Unit
Supply voltage	V <sub>cc</sub>	-0.5 to 7	V
*1 Output voltage	V <sub>o</sub>	30	V
*2 Low level output current	I <sub>OL</sub>	50	mA
*3 Operating temperature	T <sub>op</sub>	-10 to +60	°C
*4 Storage temperature	T <sub>stg</sub>	-20 to +80	°C

- \*1 Collector-emitter voltage of output transistor  
\*2 Collector current of output transistor  
\*3 The connector should be plugged in/out at normal temperature.

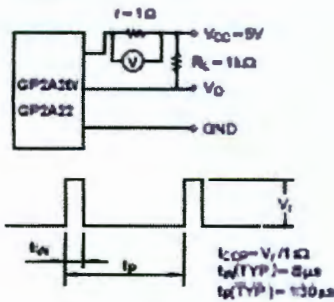
\* In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any SHARP's device, shown in catalogs, data sheets, etc. Contact SHARP in order to obtain the latest version of the device specification sheets before using any SHARP's device.\*

■ Electro - optical Characteristics

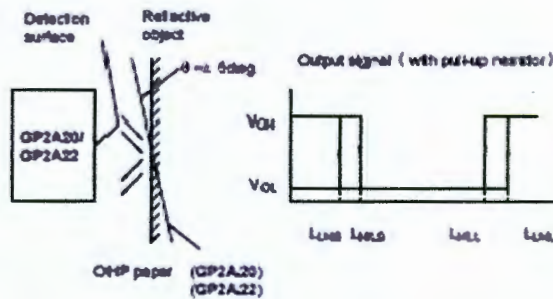
( $V_{CC} = 5V, T_a = 25^\circ C$ )

Parameter	Symbol	Conditions	MIN	TYP.	MAX.	Unit
Operating supply voltage	$V_{CC}$		4.75	-	5.25	V
Dissipation current	Peak pulse value	$I_{OP}$ $R_L = 1\Omega$	-	-	150	mA
	Smoothing value	$I_{OC}$ $R_L = \infty$	-	-	30	mA
Low level output voltage	$V_{OL}$	$I_{OL} = 16mA$ at detecting time	-	-	0.4	V
High level output voltage	$V_{OH}$	$R_L = 1k\Omega$ at non-detecting time	4.5	-	-	V
Non-detecting distance	GP2A20 GP2A22	L (mm) *Reflective object Kodat 90% reflective paper	-	-	20	mm
			-	-	50	
Minimum detecting distance	GP2A20 GP2A22 GP2A20 GP2A22 GP2A20 GP2A22	L (mm) *Reflective object Chloropren rubber *Reflective object Arstavek tape *Reflective object Kodat 90% reflective paper *Reflective object Black paper *Reflective object (GP paper: 8 - 5kg (X, Y direction))	-	-	2.5	mm
			-	-	3.0	
			-	-	1.0	
			-	-	7.0	
			-	-	1.0	
			-	-	3.0	
Maximum detecting distance	GP2A20 GP2A22 GP2A20 GP2A22 GP2A20 GP2A22	L (mm) *Reflective object Arstavek tape *Reflective object Kodat 90% reflective paper *Reflective object Black paper *Reflective object (GP paper: 8 - 5kg (X, Y direction))	7.0	-	-	mm
			2.0	-	-	
			1.0	-	-	
			15.0	-	-	
			2.0	-	-	
			1.0	-	-	
Response time	*High $\rightarrow$ Low cross transition delay time *Low $\rightarrow$ High propagation delay time	$t_{PHL}$	-	-	1	ms
		$t_{PLH}$	-	-	1	ms

\*4 Test Condition for Dissipation Current (Peak Pulse Value)



\*5 Test Condition for Detecting Distance Characteristics



\*6 Test Condition for Response Time

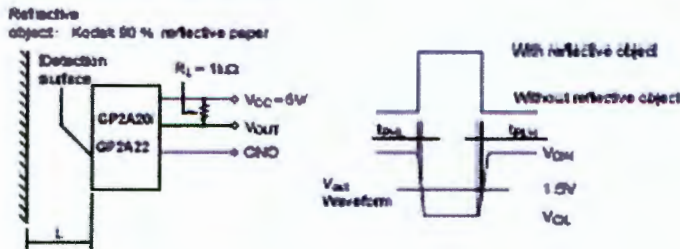




Fig. 1 Low Level Output Current vs. Ambient Temperature

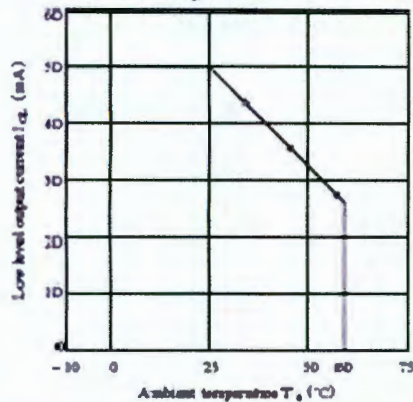


Fig. 2 Low Level Output Voltage vs. Ambient Temperature

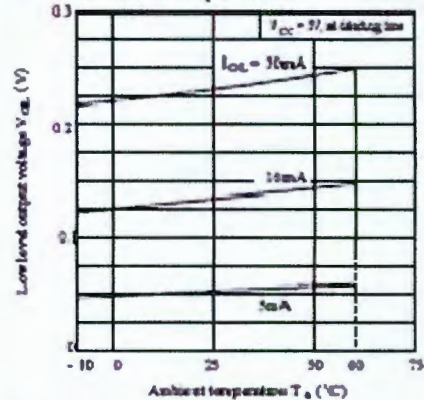


Fig. 3 Low Level Output Voltage vs. Low Level Output Current

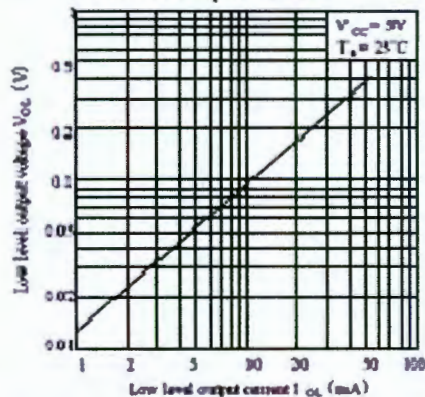
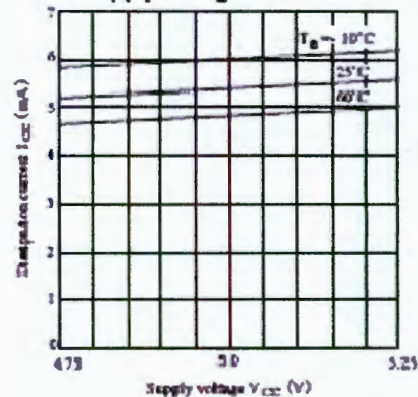


Fig. 4 Dissipation Current (Smoothing Value) vs. Supply Voltage



#### ■ Precautions for Use

- (1) In order to stabilize power supply line, connect a by-pass capacitor of more than 0.33μF between V<sub>CC</sub> and GND pin.
- (2) Please do not perform dip cleaning or ultrasonic cleaning because lens part of this product is an optical device of acrylic resin.
- (3) Remove dust or stains, using an air blower or a soft cloth moistened in cleaning solvent. However, do not perform the above cleaning using a soft cloth with cleaning solvent in the marking portion.

In this case, use only the following type of cleaning solvent used for wiping of:

Ethyl alcohol, Methyl alcohol, Isopropyl alcohol

When the cleaning solvents except for specified materials are used, please consult us.

- (4) As for other general cautions, refer to the chapter "Precautions for Use".

## 8.6 ANEXO 5 – HOJA DE DATOS BT138

Philips Semiconductors

Product specification

Triacs

BT138 series

## GENERAL DESCRIPTION

Glass passivated triacs in a plastic envelope, intended for use in applications requiring high bidirectional transient and blocking voltage capability and high thermal cycling performance. Typical applications include motor control, industrial and domestic lighting, heating and static switching.

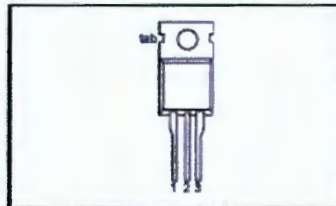
## QUICK REFERENCE DATA

SYMBOL	PARAMETER	MAX.	MAX.	MAX.	UNIT
$V_{DRM}$	Repetitive peak off-state voltages	500	600	800	V
$I_{T(RMS)}$	RMS on-state current	12	12	12	A
$I_{TSM}$	Non-repetitive peak on-state current	95	95	95	A

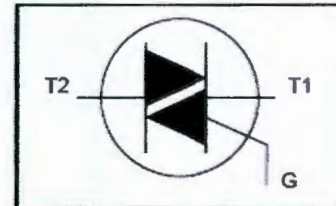
## PINNING - TO220AB

PIN	DESCRIPTION
1	main terminal 1
2	main terminal 2
3	gate
tab	main terminal 2

## PIN CONFIGURATION



## SYMBOL



## LIMITING VALUES

Limiting values in accordance with the Absolute Maximum System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.			UNIT
				-500 500 <sup>1</sup>	-600 600 <sup>1</sup>	-800 800	
$V_{DRM}$	Repetitive peak off-state voltages		-				V
$I_{T(RMS)}$	RMS on-state current	full sine wave; $T_{mb} \leq 99^\circ\text{C}$	-	12			A
$I_{TSM}$	Non-repetitive peak on-state current	full sine wave; $T_j = 25^\circ\text{C}$ prior to surge	-	95			A
		$t = 20\text{ ms}$	-	105			A
		$t = 16.7\text{ ms}$	-	45			A
$I^2t$	$I^2t$ for fusing	$t = 10\text{ ms}$	-				A <sup>2</sup> s
$di/dt$	Repetitive rate of rise of on-state current after triggering	$I_{TM} = 20\text{ A}$ ; $I_G = 0.2\text{ A}$ ; $di_G/dt = 0.2\text{ A}/\mu\text{s}$	-				A/ $\mu\text{s}$
		T2+ G+	-	50			A/ $\mu\text{s}$
		T2+ G-	-	50			A/ $\mu\text{s}$
		T2- G-	-	50			A/ $\mu\text{s}$
		T2- G+	-	10			A/ $\mu\text{s}$
$I_{GM}$	Peak gate current		-	2			A
$V_{GM}$	Peak gate voltage		-	5			V
$P_{GM}$	Peak gate power		-	5			W
$P_{GM(AV)}$	Average gate power	over any 20 ms period	-	0.5			W
$T_{MS}$	Storage temperature		-40	150			$^\circ\text{C}$
$T_j$	Operating junction temperature		-	125			$^\circ\text{C}$

<sup>1</sup> Although not recommended, off-state voltages up to 800V may be applied without damage, but the triac may switch to the on-state. The rate of rise of current should not exceed 15 A/ $\mu\text{s}$ .

## Triacs

## BT138 series

## THERMAL RESISTANCES

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$R_{th,mb}$	Thermal resistance junction to mounting base	full cycle	-	-	1.5	K/W
$R_{th,a}$	Thermal resistance junction to ambient	half cycle in free air	-	60	2.0	K/W K/W

## STATIC CHARACTERISTICS

 $T_J = 25\text{ }^\circ\text{C}$  unless otherwise stated

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.			UNIT
$I_{GT}$	Gate trigger current	BT138- $V_D = 12\text{ V}; I_T = 0.1\text{ A}$	-	-	...	...F	...G	
		T2+ G+	-	5	35	25	50	mA
		T2+ G-	-	8	35	25	50	mA
		T2- G-	-	10	35	25	50	mA
$I_L$	Latching current	BT138- $V_D = 12\text{ V}; I_{GT} = 0.1\text{ A}$	-	-	...	...F	...G	
		T2+ G+	-	22	70	70	100	mA
		T2+ G-	-	7	40	40	60	mA
		T2+ G-	-	20	60	60	90	mA
$I_H$	Holding current	BT138- $V_D = 12\text{ V}; I_{GT} = 0.1\text{ A}$	-	-	...	...F	...G	
		T2+ G+	-	8	40	40	60	mA
		T2+ G-	-	10	60	60	90	mA
		T2- G+	-	6	30	30	60	mA
$V_T$	On-state voltage	$I_T = 15\text{ A}$	-	1.4	1.65			V
$V_{GT}$	Gate trigger voltage	$V_D = 12\text{ V}; I_T = 0.1\text{ A}$	-	0.7	1.5			V
		$V_D = 400\text{ V}; I_T = 0.1\text{ A};$ $T_J = 125\text{ }^\circ\text{C}$	0.25	0.4	-			V
$I_D$	Off-state leakage current	$V_D = V_{DRM(max)}$ $T_J = 125\text{ }^\circ\text{C}$	-	0.1	0.5			mA

## DYNAMIC CHARACTERISTICS

 $T_J = 25\text{ }^\circ\text{C}$  unless otherwise stated

SYMBOL	PARAMETER	CONDITIONS	MIN.			TYP.	MAX.	UNIT
$dV_D/dt$	Critical rate of rise of off-state voltage	BT138- $V_{DM} = 67\% V_{DRM(max)}$ $T_J = 125\text{ }^\circ\text{C}$ ; exponential waveform; gate open circuit	100	...	...G	250	-	V/ $\mu\text{s}$
$dV_{com}/dt$	Critical rate of change of commutating voltage	$V_{DM} = 400\text{ V}; T_J = 95\text{ }^\circ\text{C};$ $I_{T(RMS)} = 12\text{ A};$ $di_{com}/dt = 5.4\text{ A/ms}$ ; gate open circuit	-	-	10	20	-	V/ $\mu\text{s}$
$t_{gt}$	Gate controlled turn-on time	$I_{TM} = 16\text{ A}; V_D = V_{DRM(max)}$ $I_G = 0.1\text{ A}; di_G/dt = 5\text{ A}/\mu\text{s}$	-	-	-	2	-	$\mu\text{s}$

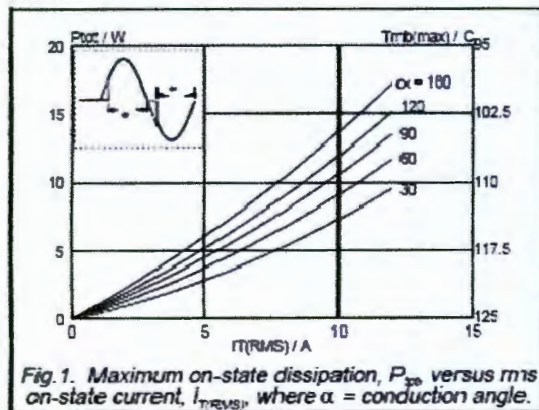


Fig. 1. Maximum on-state dissipation,  $P_{on}$  versus rms on-state current,  $I_{T(RMS)}$ , where  $\alpha$  = conduction angle.

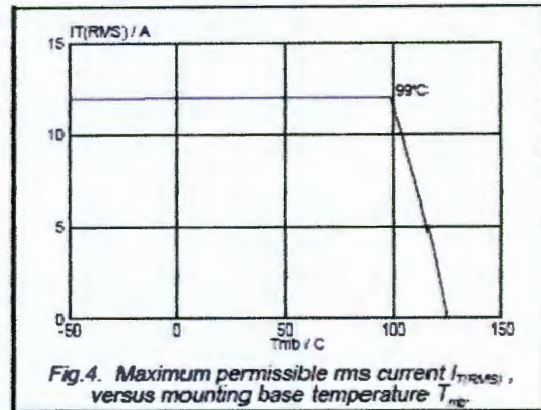


Fig. 4. Maximum permissible rms current  $I_{T(RMS)}$ , versus mounting base temperature  $T_{mb}$ .

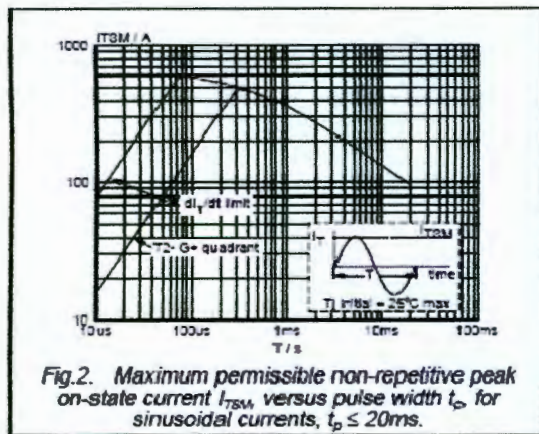


Fig. 2. Maximum permissible non-repetitive peak on-state current  $I_{TSM}$  versus pulse width  $t_p$  for sinusoidal currents,  $t_p \leq 20ms$ .

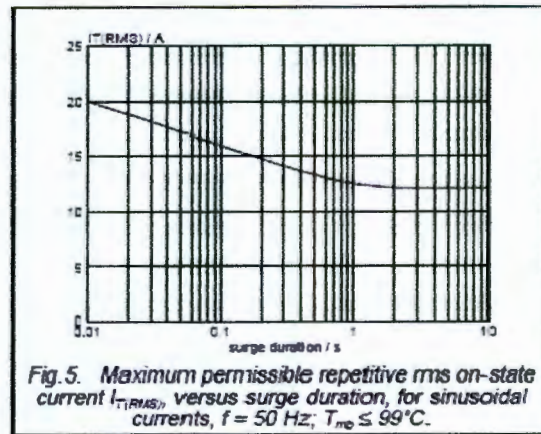


Fig. 5. Maximum permissible repetitive rms on-state current  $I_{T(RMS)}$  versus surge duration, for sinusoidal currents,  $f = 50 Hz$ ;  $T_{mb} \leq 99^\circ C$ .

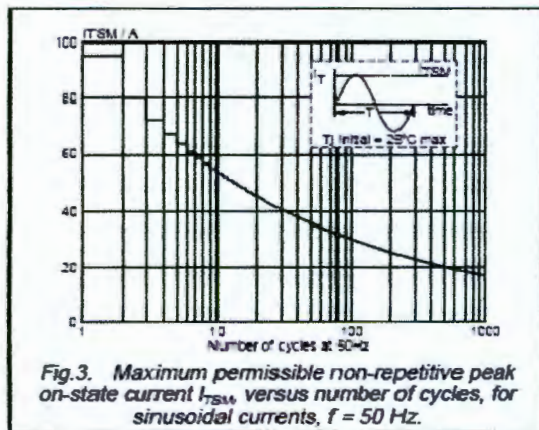


Fig. 3. Maximum permissible non-repetitive peak on-state current  $I_{TSM}$  versus number of cycles, for sinusoidal currents,  $f = 50 Hz$ .

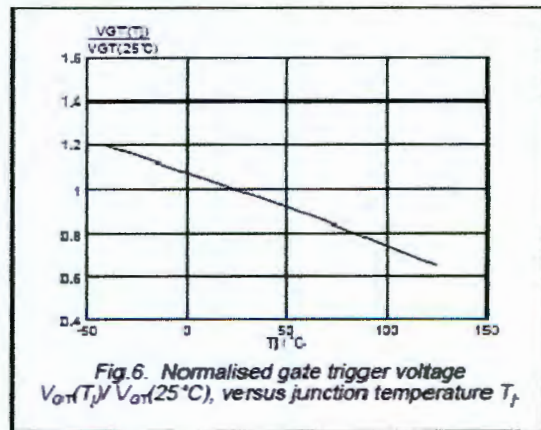


Fig. 6. Normalised gate trigger voltage  $V_{GT}(T_j)/V_{GT}(25^\circ C)$ , versus junction temperature  $T_j$ .

Triacs

BT138 series

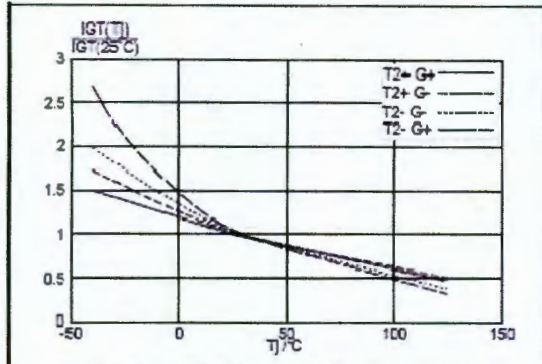


Fig. 7. Normalised gate trigger current  $I_{GT}(T_j)/I_{GT}(25^\circ\text{C})$ , versus junction temperature  $T_j$ .

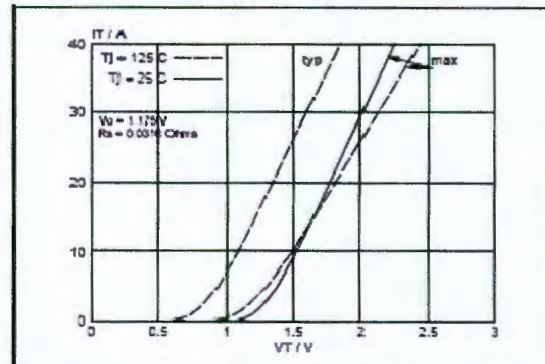


Fig. 10. Typical and maximum on-state characteristic.

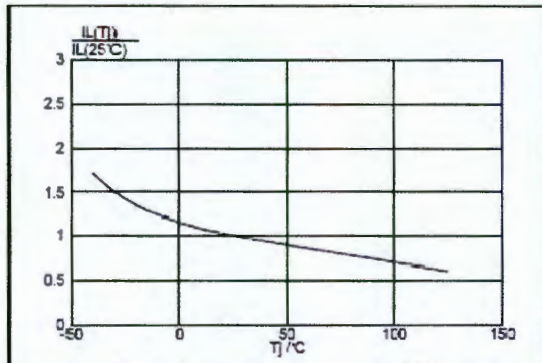


Fig. 8. Normalised latching current  $I_L(T_j)/I_L(25^\circ\text{C})$ , versus junction temperature  $T_j$ .

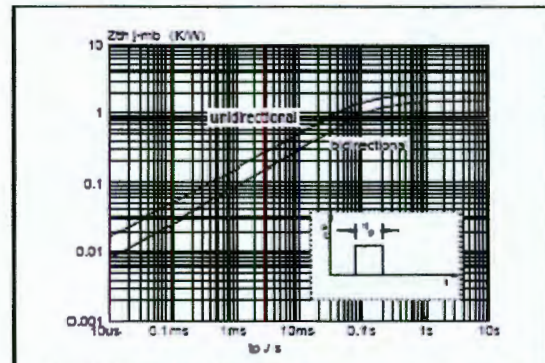


Fig. 11. Transient thermal impedance  $Z_{th(jmb)}$  versus pulse width  $t_p$ .

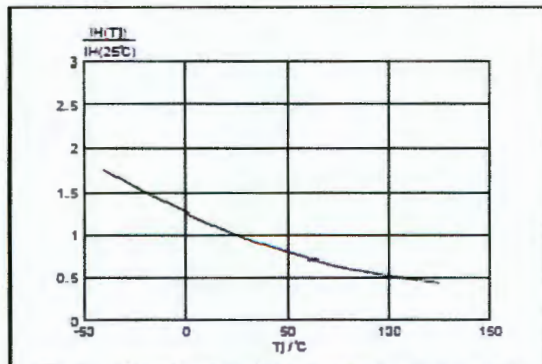


Fig. 9. Normalised holding current  $I_H(T_j)/I_H(25^\circ\text{C})$ , versus junction temperature  $T_j$ .

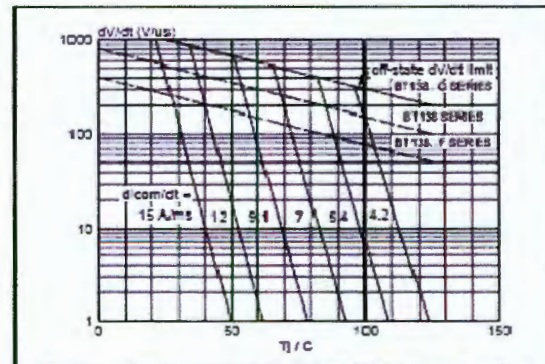


Fig. 12. Typical commutation  $dV/dt$  versus junction temperature, parameter commutation  $dI_T/dt$ . The triac should commute when the  $dV/dt$  is below the value on the appropriate curve for pre-commutation  $dI_T/dt$ .

## MECHANICAL DATA

Dimensions in mm

Net Mass: 2 g

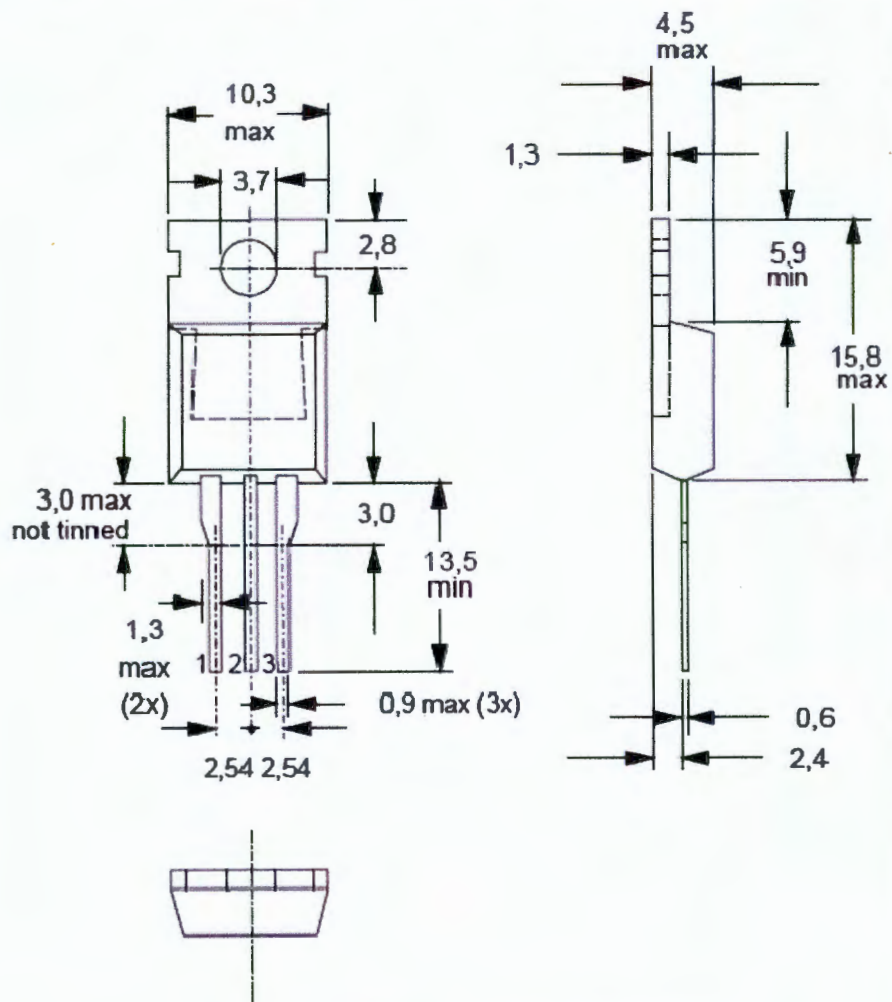


Fig.13. TO220AB; pin 2 connected to mounting base.

## Notes

1. Refer to mounting instructions for TO220 envelopes.
2. Epoxy meets UL94 V0 at 1/8".

## 8.7 ANEXO 6 – HOJA DE DATOS MOC3031

**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M    MOC3032M    MOC3033M    MOC3041M    MOC3042M    MOC3043M

**DESCRIPTION**

The MOC303XM and MOC304XM devices consist of a AlGaAs infrared emitting diode optically coupled to a monolithic silicon detector performing the function of a zero voltage crossing bilateral triac driver.

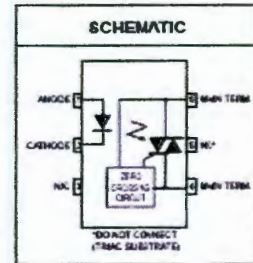
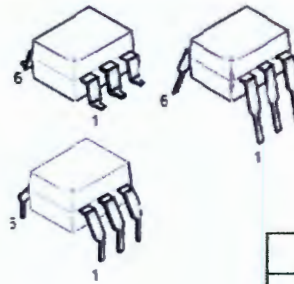
They are designed for use with a triac in the interface of logic systems to equipment powered from 115 VAC lines, such as teletypewriters, CRTs, solid-state relays, industrial controls, printers, motors, solenoids and consumer appliances, etc.

**FEATURES**

- Simplifies logic control of 115 VAC power
- Zero voltage crossing
- $dv/dt$  of 2000 V/μs typical, 1000 V/μs guaranteed
- VDE recognized (File # 94786)
- ordering option V (e.g., MOC3043VM)

**APPLICATIONS**

- Solenoid/valve controls
- Static power switches
- Temperature controls
- AC motor starters
- Lighting controls
- AC motor drives
- E.M. contactors
- Solid state relays

**ABSOLUTE MAXIMUM RATINGS** ( $T_A = 25^\circ\text{C}$  unless otherwise noted)

Parameters	Symbol	Device	Value	Units
<b>TOTAL DEVICE</b>				
Storage Temperature	$T_{STG}$	All	-40 to +150	$^\circ\text{C}$
Operating Temperature	$T_{OPR}$	All	-40 to +85	$^\circ\text{C}$
Lead Solder Temperature	$T_{SOL}$	All	260 for 10 sec	$^\circ\text{C}$
Junction Temperature Range	$T_J$	All	-40 to +100	$^\circ\text{C}$
Isolation Surge Voltage <sup>(1)</sup> (peak AC voltage, 60Hz, 1 sec duration)	$V_{ISO}$	All	7500	Vac(pk)
Total Device Power Dissipation @ 25 $^\circ\text{C}$ Derate above 25 $^\circ\text{C}$	$P_D$	All	250	mW
			2.94	mW/ $^\circ\text{C}$
<b>EMITTER</b>				
Continuous Forward Current	$I_F$	All	50	mA
Reverse Voltage	$V_R$	All	6	V
Total Power Dissipation 25 $^\circ\text{C}$ Ambient Derate above 25 $^\circ\text{C}$	$P_D$	All	120	mW
			1.41	mW/ $^\circ\text{C}$
<b>DETECTOR</b>				
Off-State Output Terminal Voltage	$V_{DRM}$	MOC3031M/2M/3M	250	V
		MOC3041M/2M/3M	<100	
Peak Repetitive Surge Current (Pulse = 100 μs, 100 pulses)	$I_{TCM}$	All	1	A
Total Power Dissipation @ 25 $^\circ\text{C}$ Ambient Derate above 25 $^\circ\text{C}$	$P_D$	All	150	mW
			1.76	mW/ $^\circ\text{C}$

**Note**

1. Isolation surge voltage,  $V_{ISO}$ , is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common and Pins 4, 5 and 6 are common.

**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M MOC3032M MOC3033M MOC3041M MOC3042M MOC3043M

**ELECTRICAL CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  Unless otherwise specified)

**INDIVIDUAL COMPONENT CHARACTERISTICS**

Parameters	Test Conditions	Symbol	Device	Min	Typ	Max	Units
<b>EMITTER</b>							
Input Forward Voltage	$I_F = 30\text{ mA}$	$V_F$	All		1.25	1.5	V
Reverse Leakage Current	$V_R = 8\text{ V}$	$I_R$	All		0.01	100	$\mu\text{A}$
<b>DETECTOR</b>							
Peak Blocking Current, Either Direction	Rated $V_{DRM}$ , $I_F = 0$ (note 1)	$I_{DRM1}$	All			100	nA
Peak On-State Voltage, Either Direction	$I_{TM} = 100\text{ mA peak}$ , $I_F = 0$	$V_{TM}$	All		1.8	3	V
Critical Rate of Rise of Off-State Voltage	$I_F = 0$ (figure 9, note 3)	$dv/dt$	All	1000			V/ $\mu\text{s}$

**TRANSFER CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  Unless otherwise specified.)

DC Characteristics	Test Conditions	Symbol	Device	Min	Typ	Max	Units
LED Trigger Current	Main terminal voltage = 3V (note 2)	$I_{FT}$	MOC3031M/MOC3041M			15	mA
			MOC3032M/MOC3042M			10	
			MOC3033M/MOC3043M			5	
Holding Current, Either Direction		$I_H$	All		400		$\mu\text{A}$

**ZERO CROSSING CHARACTERISTICS** ( $T_A = 25^\circ\text{C}$  Unless otherwise specified.)

Characteristics	Test Conditions	Symbol	Device	Min	Typ	Max	Units
Inhibit Voltage	$I_F = \text{rated } I_{FT}$ , MT1-MT2 voltage above which device will not trigger off-state	$V_{IH}$	All			20	V
Leakage in Inhibited State	$I_F = \text{rated } I_F$ , rated $V_{DRM}$ , off-state	$I_{DRM2}$	All			500	$\mu\text{A}$

Note

1. Test voltage must be applied within  $dv/dt$  rating.
2. All devices are guaranteed to trigger at an  $I_F$  value less than or equal to max  $I_{FT}$ . Therefore, recommended operating  $I_F$  lies between max  $I_{FT}$  (15 mA for MOC3031M & MOC3041M, 10 mA for MOC3032M & MOC3042M, 5 mA for MOC3033M & MOC3043M) and absolute max  $I_F$  (60 mA).
3. This is static  $dv/dt$ . See Figure 9 for test circuit. Commutating  $dv/dt$  is a function of the load-driving thyristor(s) only.



**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M

MOC3032M

MOC3033M

MOC3041M

MOC3042M

MOC3043M

Figure 1. LED Forward Voltage vs. Forward Current

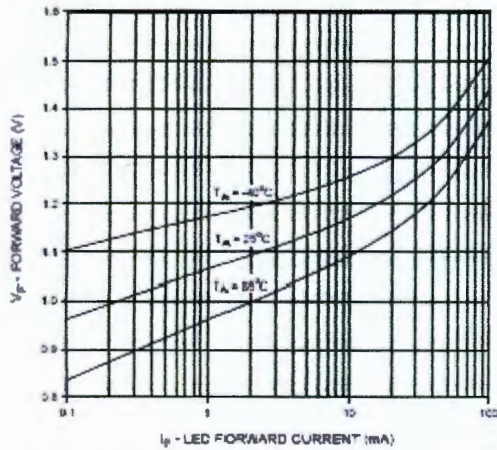


Figure 2. On-State Characteristics

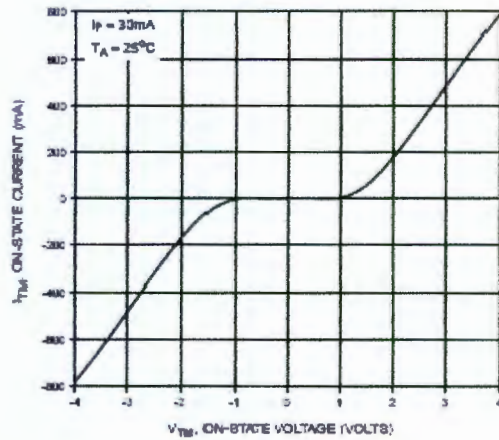
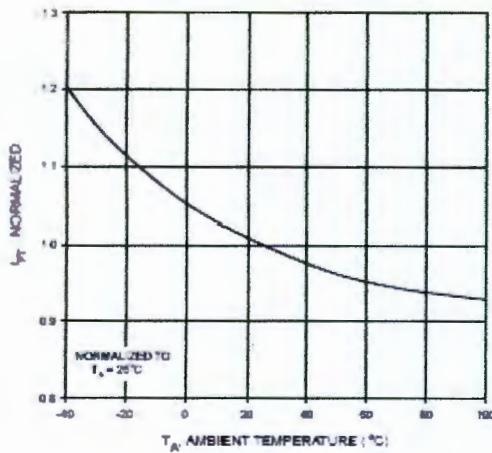
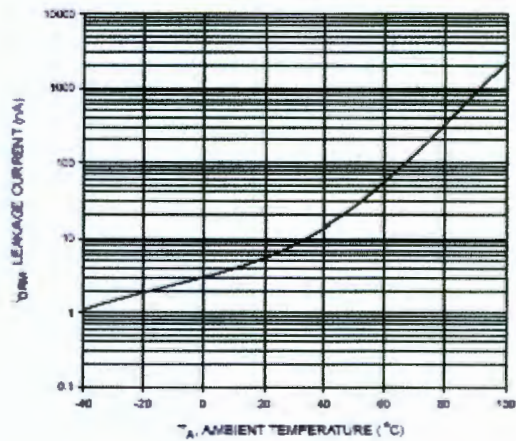


Figure 3. Trigger Current vs. Temperature

Figure 4. Leakage Current,  $I_{D(Leak)}$  vs. Temperature

**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M

MOC3032M

MOC3033M

MOC3041M

MOC3042M

MOC3043M

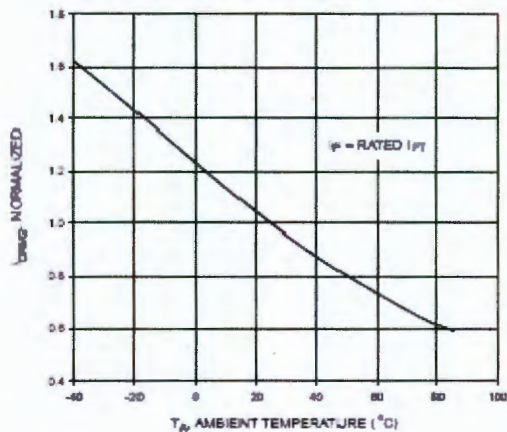
Figure 5.  $I_{DRM2}$  - Leakage in Inhibit State vs. Temperature

Figure 6. LED Current Required to Trigger vs. LED Pulse Width

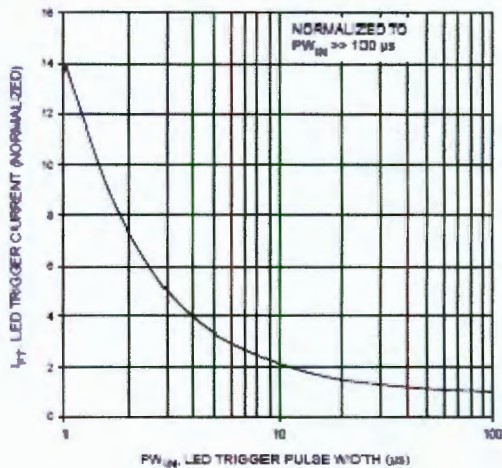
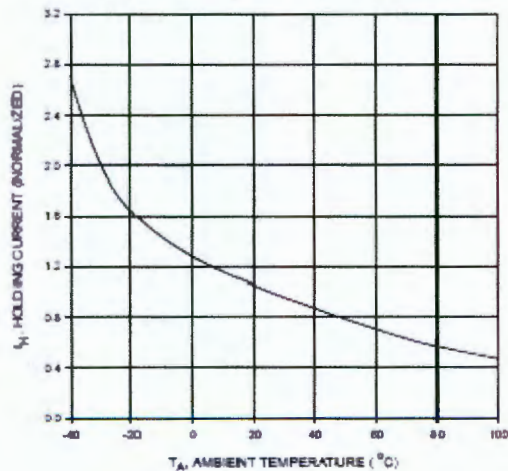
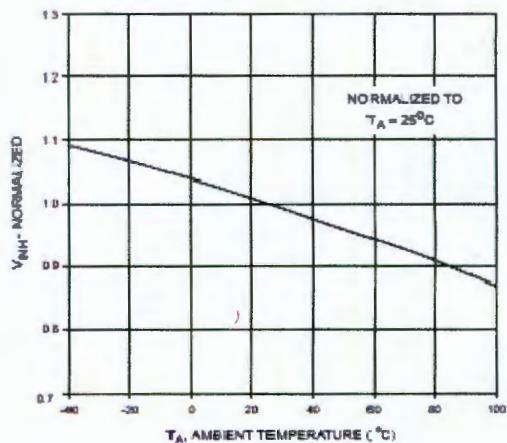
Figure 7. Holding Current,  $I_H$  vs. Temperature

Figure 8. Inhibit Voltage vs. Temperature



**FAIRCHILD**  
SEMICONDUCTOR®

## 6-PIN DIP ZERO-CROSS OPTOISOLATORS TRIAC DRIVER OUTPUT (250/400 VOLT PEAK)

MOC3031M

MOC3032M

MOC3033M

MOC3041M

MOC3042M

MOC3043M

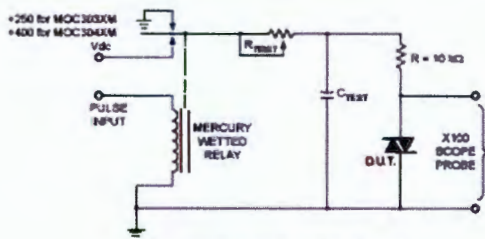
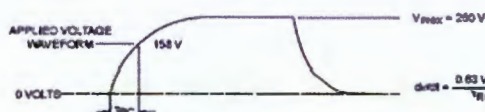
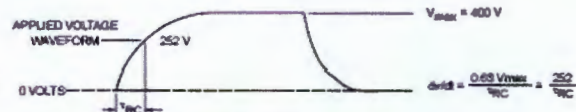


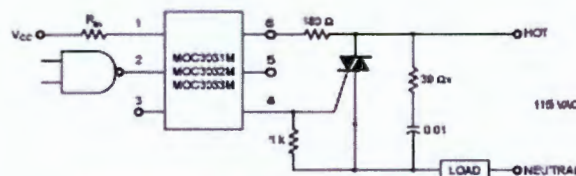
Figure 9. Static dv/dt Test Circuit

1. The mercury wetted relay provides a high speed repeated pulse to the D.U.T.
2. 100x scope probes are used, to allow high speeds and voltages.
3. The worst-case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable  $R_{TEST}$  allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering.  $\tau_{RC}$  is measured at this point and recorded.

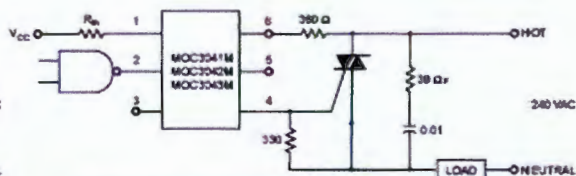
Figure 10. Static dv/dt Test Waveform  
(MOC3031M, MOC3032M, MOC3033M)Figure 11. Static dv/dt Test Waveform  
(MOC3041M, MOC3042M, MOC3043M)

Typical circuit (Fig 12, 13) for use when hot line switching is required. In this circuit the "hot" side of the line is switched and the load connected to the cold or neutral side. The load may be connected to either the neutral or hot line.

$R_{IN}$  is calculated so that  $I_F$  is equal to the rated  $I_{FT}$  of the part, 5 mA for the MOC3033M and MOC3043M, 10 mA for the MOC3032M and MOC3042M, or 15 mA for the MOC3031M and MOC3041M. The 39 ohm resistor and 0.01  $\mu$ F capacitor are for snubbing of the triac and may or may not be necessary depending upon the particular triac and load used.



\*For highly inductive loads (power factor < 0.5), change this value to 360 ohms.

Figure 12. Hot-Line Switching Application Circuit  
(MOC3031M, MOC3032M, MOC3033M)

\*For highly inductive loads (power factor < 0.5), change this value to 330 ohms.

Figure 13. Hot-Line Switching Application Circuit  
(MOC3041M, MOC3042M, MOC3043M)

**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M MOC3032M MOC3033M MOC3041M MOC3042M MOC3043M

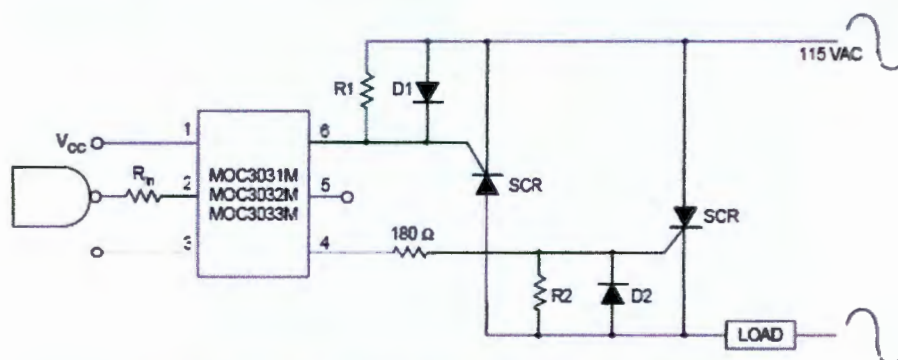


Figure 14. Inverse-Parallel SCR Driver Circuit  
(MOC3031M, MOC3032M, MOC3033M)

Suggested method of firing two, back-to-back SCR's with a Fairchild triac driver. Diodes can be 1N4001; resistors, R1 and R2, are optional 1 k ohm.

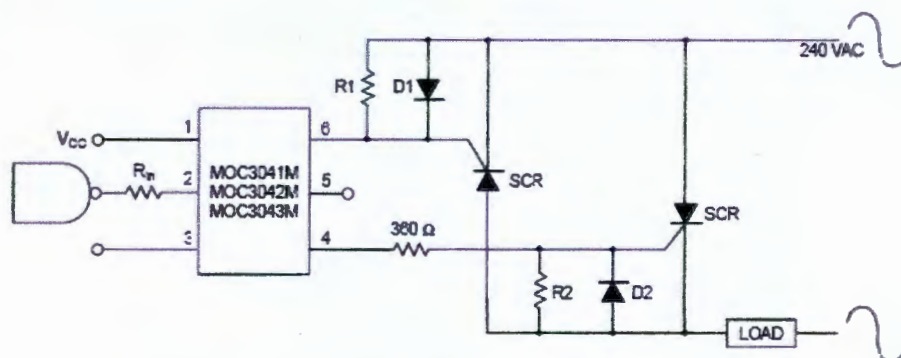


Figure 16. Inverse Parallel SCR Driver Circuit  
(MOC3041M, MOC3042M, MOC3043M)

Suggested method of firing two, back-to-back SCR's with a Fairchild triac driver. Diodes can be 1N4001; resistors, R1 and R2, are optional 330 ohm.

Note: This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only.

**FAIRCHILD**  
SEMICONDUCTOR®

**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M

MOC3032M

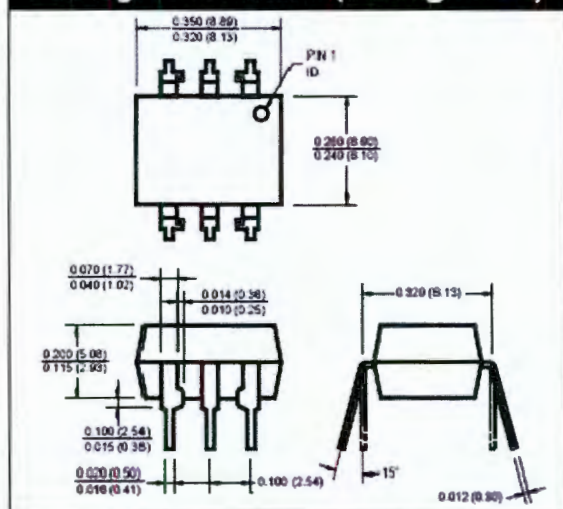
MOC3033M

MOC3041M

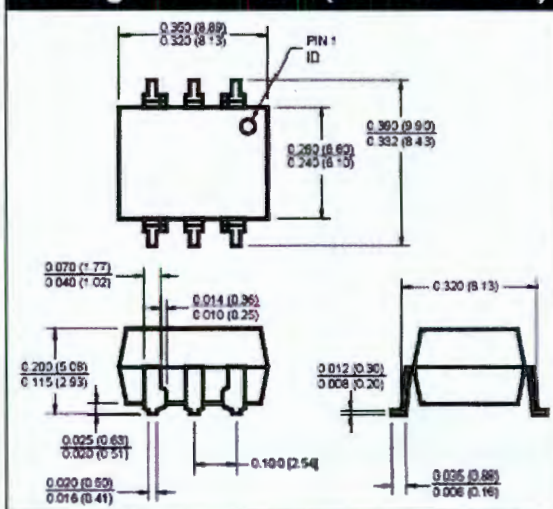
MOC3042M

MOC3043M

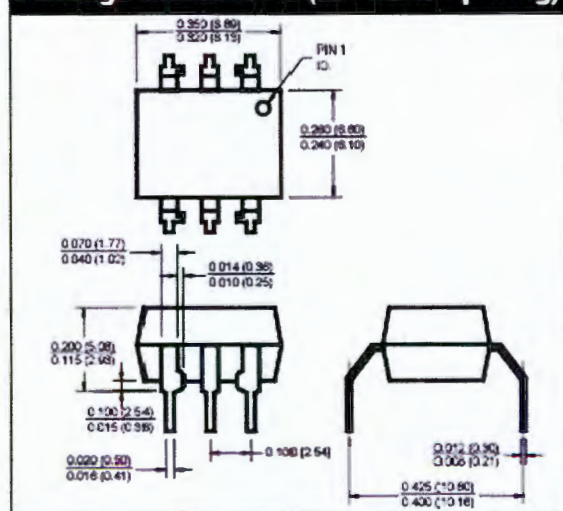
**Package Dimensions (Through Hole)**



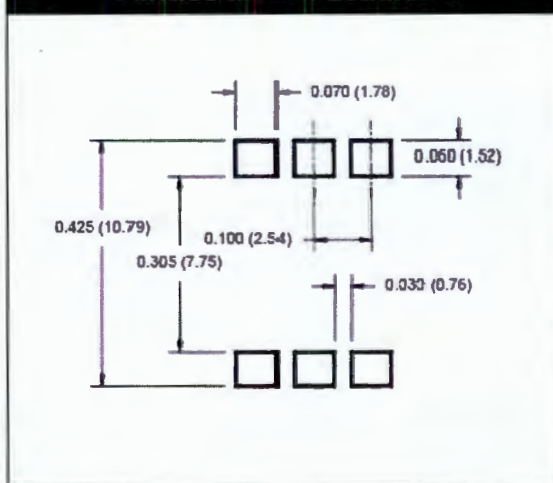
**Package Dimensions (Surface Mount)**



**Package Dimensions (0.4" Lead Spacing)**



**Recommended Pad Layout for  
Surface Mount Leadform**



**NOTE**

All dimensions are in inches (millimeters)

**FAIRCHILD**  
SEMICONDUCTOR\*

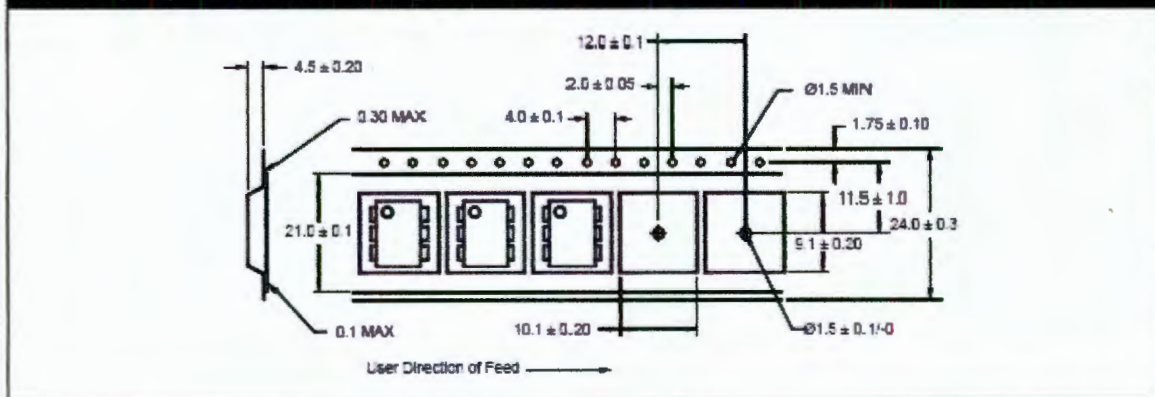
**6-PIN DIP ZERO-CROSS  
OPTOISOLATORS TRIAC DRIVER OUTPUT  
(250/400 VOLT PEAK)**

MOC3031M MOC3032M MOC3033M MOC3041M MOC3042M MOC3043M

**ORDERING INFORMATION**

Option	Order Entry Identifier	Description
S	S	Surface Mount Lead Bend
SR2	SR2	Surface Mount, Tape and reel
T	T	0.4" Lead Spacing
V	V	VDE 0884
TV	TV	VDE 0884, 0.4" Lead Spacing
SV	SV	VDE 0884, Surface Mount
SR2V	SR2V	VDE 0884, Surface Mount, Tape & Reel

**Carrier Tape Specifications ("D" Taping Orientation)**



**NOTE**

All dimensions are in inches (millimeters)