"EQUIPO AUXILIAR DE MEDICIÓN CONTROLADO POR COMPUTADORA PARA LABORATORIOS DE ELECTRÓNICA"

TRABAJO DE GRADUACIÓN PREPARADO PARA LA FACULTAD DE ESTUDIOS TECNOLÓGICOS

PARA OPTAR AL GRADO DE:

TÉCNICO EN INGENIERÍA ELECTRÓNICA
OPCIÓN: CONTROL Y AUTOMATIZACIÓN

UNIVERSIDAD DON BOSCO
Biblioteca Central

PRESENTADO POR:

JOSÉ LUIS MOLINA OROZCO
SAÚL ERNESTO ARCE ECHEVERRÍA

NOVIEMBRE DEL 2001
SOYAPANGO, EL SALVADOR, CENTRO AMÉRICA

UNIVERSIDAD DON BOSCO

RECTOR:

ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL:
LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE ESTUDIOS TECNOLÓGICOS:
ING. VÍCTOR ARNOLDO CORNEJO MONTANO

ASESOR DEL TRABAJO DE GRADUACIÓN:
ING. FEDERICO LAÍNEZ

JURADO EVALUADOR:
TEC. FRANCISCO SALVADOR TAURA
TEC. CESAR MELGAR ACOSTA

UNIVERSIDAD DON BOSCO

FACULTAD DE ESTUDIOS TECNOLÓGICOS

JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN:

"EQUIPO AUXILIAR DE MEDICIÓN CONTROLADO POR COMPUTADORA PARA LABORATORIOS DE ELECTRÓNICA"

Tec. Francisco Taura
JURADO

Tec. Cesar Melgar Acosta
JURADO

Ing. Federico laínez
ASESOR

INTRODUCCIÓN

Todo laboratorio de electrónica para propósitos educativos en las áreas de Ingeniería y Tecnológico, tiene que estar debidamente equipado con diferentes tipos de accesorios e instrumentos de medición, los cuales son necesarios para el correcto aprendizaje y desarrollo de las prácticas de aquellas personas que experimentan en él.

Debido a un creciente avance en los sistemas computacionales y electrónicos, se han logrado desarrollar nuevos y mejores equipos de medición para los laboratorios de electrónica. En nuestro país este tipo de instrumentos se difunden a un ritmo muy lento, debido a diversas circunstancias en cuanto a desarrollo tecnológico, precios inaccesibles, dificultad para adquirirlos, entre otras razones. Pero es ese mismo ritmo cambiante y acelerado en cuanto a tecnología, que nos incentiva a desarrollar nuevas y modernas ideas, así como también mejorar aquellos equipos que en un momento dado no logran suplir las necesidades de los alumnos y que están expuestos a fallar por un mal uso.

Es precisamente por ello que proponemos un equipo auxiliar de medición para laboratorios de electrónica, con un adecuado sistema de protección para evitar fallos por un mal manejo, con todos los componentes esenciales para realizar las prácticas, además con una interfaz usuario – equipo más interactiva y sencilla de utilizar, cuya manipulación se hará directamente de una computadora para facilitar al usuario la interpretación y apreciación de características que provengan de algún tipo de circuito en análisis.

OBJETIVO GENERAL

- Construir un equipo de medición de laboratorio con múltiples funciones, que sea preciso, eficiente, interactivo y sencillo de utilizar.

OBJETIVO ESPECIFICO

- Disminuir el espacio de operación que comúnmente suelen ocupar otros equipos de medición por encontrarse de forma separada cada uno de los instrumentos.

- Reducir los gastos a corto y largo plazo que son ocasionados por un mal manejo, mediante la implementación de este equipo de medición.

- Construir una interfaz que permita que la PC y el EAM (Equipo Auxiliar de medición) puedan interactuar.

- Elaborar un programa que sea capaz de controlar la interfaz, analizar el muestreo, permitir que el usuario manipule los instrumentos y que pueda realizar la presentación gráfica de las variables y señales eléctricas.

- Proteger al equipo de medición de manipulaciones inadecuadas que se cometen con mayor frecuencia.

ALCANCES

- Se tiene acceso a todos los componentes del equipo de medición desde la computadora, mediante la cual se tomarán las decisiones de control y programación del funcionamiento del equipo.
- En caso de que el usuario cometa un mal manejo del equipo al momento de realizar una medición, este, se protegerá para evitar posibles daños.
- Elaboración de un EAM que sea capaz de realizar funciones de protección, adquisición de datos, almacenamiento temporal de información, conversión de una variable eléctrica a otra, conversión de variables eléctricas a digitales y viceversa, etc.
- El sistema se podrá instalar en aquellas computadoras que cumplan con los requerimientos del programa.
- Elaboración de un manual con toda la información necesaria para el correcto manejo del equipo; logrando con esto que el usuario esté sabedor del uso adecuado que se le debe de dar a cada instrumento para que obtenga de él un máximo provecho.

LIMITACIONES

- Por su forma de protección, está más inclinada a proteger al EAM y no al circuito externo, con aquellas fallas que se cometen con mayor frecuencia, esto; debido a que se puede conectar de una forma incorrecta, provocando que la protección no detecte esta anomalía ocasionando el daño al circuito pero protegiendo al equipo.

- Los rangos de medición del EAM controlado por computadora, tendrán un límite de medición los cuales se detallan en la tabla de la pag. 32.

- El usuario no tendrá acceso directo a la base de datos, solamente a las variables de trabajo y opciones de la interfaz gráfica.

- Los requerimientos mínimos de instalación del equipo será para computadoras 80386SX (33MHZ) o superior, 4MB de memoria RAM y bajo un ambiente DOS.

INDICE

CAPITULO 1 — Pagina

FUNDAMENTACION TEÓRICA. **1**

Funcionamiento del bus ISA. 1
Convertidores A/D y D/A. 6
Generalidades de las fuentes de voltaje. 12
Contadores de frecuencia. 14
Generadores de funciones. 16
Introducción a los Osciloscopios. 17
Generalidades de los multimetros. 24
Sistema de protección en los equipos de laboratorio. 27

CAPITULO 2

DESCRIPCIÓN GENERAL DEL PROYECTO. **29**

Aspectos generales del proyecto. 29
Funcionamiento de las etapas del proyecto. 30
Sistema de protección del EAM. 33
Metodología de diseño e implementación. 35

CAPITULO 3

CONSIDERACIONES PARA EL DISEÑO DEL EAM. **37**

Interfaz interna. 37
Interfaz externa. 38
Fuente de voltaje. 41
Frecuencimetro. 42
Generador de funciones. 44
Osciloscopio. 48
Multímetro. 52
Diagramas esquemáticos del EAM. 56

CAPITULO 4

TÉCNICAS DE CONTROL POR SOFTWARE. **64**

Determinación de la frecuencia para el frecuencímetro. 64
Determinación del voltaje de salida de la fuente. 64
Determinación de la amplitud y frecuencia para el GEN – FUN. 65
Proceso para determinar las escalas del multímetro. 67
Determinación de las escalas VOLT/DIV y TIME/DIV. 71

Proceso de presentación en pantalla. 73
Determinación del periodo y magnitud de la señal. 75
Flujograma general del programa. 80
Programa de la interfaz gráfica. 88

CAPITULO 5

RESULTADOS OBTENIDOS Y SUS POSIBLES MEJORAS. 162

Pruebas y análisis de resultados. 162
Recomendaciones y mejoras. 169
Uso de la interfaz gráfica del EAM. 173

Bibliografía. 177

ANEXOS 178

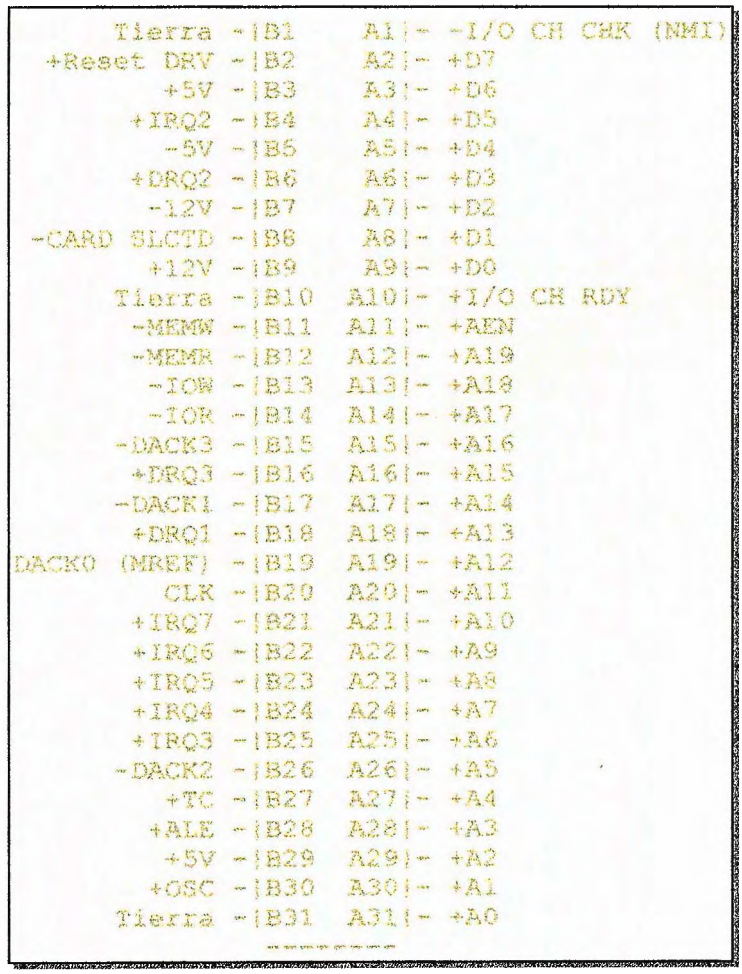
Presupuesto del proyecto. 178
Cronograma de actividades. 180
Diferentes vistas del EAM. 184
Circuitos impresos del EAM. 185
Glosario técnico. 193
Hojas de especificaciones técnicas. 198

CAPITULO 1

FUNDAMENTACIÓN TEÓRICA.

Estructura del BUS ISA 8/16 bits

El BUS clásico de un PC (ISA BUS) se compone de dos partes:
La clásica de 8 bits, perteneciente a los PC, XT y AT. La extensión de 16 bits de los AT. Entre ambos forman el bus ISA que todos los ordenadores PC actuales poseen (no confundir con VEZA o PCI, siendo el primero una tercera ampliación del bus ISA de 8 bits y el PCI un bus totalmente diferente) estructura del BUS de 8 bits PC, XT, y AT:



Nota: La numeración empieza desde la parte posterior de la máquina.

SEÑAL	DESCRIPCIÓN
A0 - A19	Bits de dirección 0-19, permiten direccionar 1Mb de memoria y 64k de puertos de E/S
A17 - A23	Bits de dirección 17-23, permite direccionar desde 256Kb de memoria a 16Mb
AEN	Address enable; cuando está activa el controlador DMA posee el control de las líneas de dirección y del BUS de datos, conforme se indique en MEMR/MEMW cuando está inactiva la CPU tiene control de estas líneas.
ALE	Address latch enable (salida); se emplea para que el CPU esté aislado de líneas de dirección (triestado) es forzado activado durante los ciclos de DMA.
CARD SLCTD	Card selected; indica que una tarjeta ha sido activada en el slot XT de 8bits.
CLK	Señal de reloj del sistema(actual velocidad del BUS)
D0-D7	Bits de datos 0 - 7 para E/S a memoria o puertos de E/S
DACK0-DACK3	Reconocimiento DMA para los canales 0 al 3; empleada por un controlador para reconocer una petición DMA (validación de acceso DMA) DACK0 es empleada para el refresco de memoria (MREF).
DRQ0-DRQ3	Petición de DMA 0-3 empleada por periféricos que desean los servicios de controlador DMA; se mantiene activa hasta que la correspondiente señal DACKx se hace activa.
I/O CH CHK	I/O Channel check; genera una interrupción no enmascarable
I/O CH RDY	I/O Channel ready; en puesta inactiva por memoria o dispositivos de E/S para retardar el acceso a memoria o los ciclos de E/S. Normalmente es empleada por dispositivos lentos para añadir estados de espera. No debe ser inactiva durante mas de 17 ciclos.
I/O CS 16	I/O Chip select 16 bits; indica ciclo de E/S de 16 bits.
MEM CS 16	Memory chip select 16 bit; Indica ciclo de memoria de 16 bits
IOR	I/O Read; indica a un dispositivo de E/S que coloque su dato en el bus del sistema
IOW	I/O WRITE; Indica que la CPU mandara información a un dispositivo de E/S
IRQ2-IRQ7	Petición de interrupción 2-7; indica a la CPU que un dispositivo de E/S necesita servicio.
MEMR	Memory read; esta señal es producida por la CPU o es el controlador DMA e indica a la memoria que debe introducir el dato direccionado en el BUS del sistema, presente tanto en le BUS PC como en la extensión AT.
MEMW	Memory write; esta señal es producida por la CPU o el controlador DMA e indica a la memoria que debe leer y almacenar el dato presente en el BUS. Presente tanto en el BUS PC como en la extensión AT.
OSC	Oscilador, señal de reloj de 14.31818 MHZ (período de 70ns), 50% del ciclo de servicio.
RESET DRV	Reset driver; empleada Para resetear la lógica del sistema.
SBHE	System BUS high enable; activa los bits de datos 8 -15 de la extensión AT del bus.
MASTER	Empleado por DRQ para ganar el control del sistema.
TC	Terminal count; produce un impulso cuando la cuenta final de un canal DMA es alcanzado

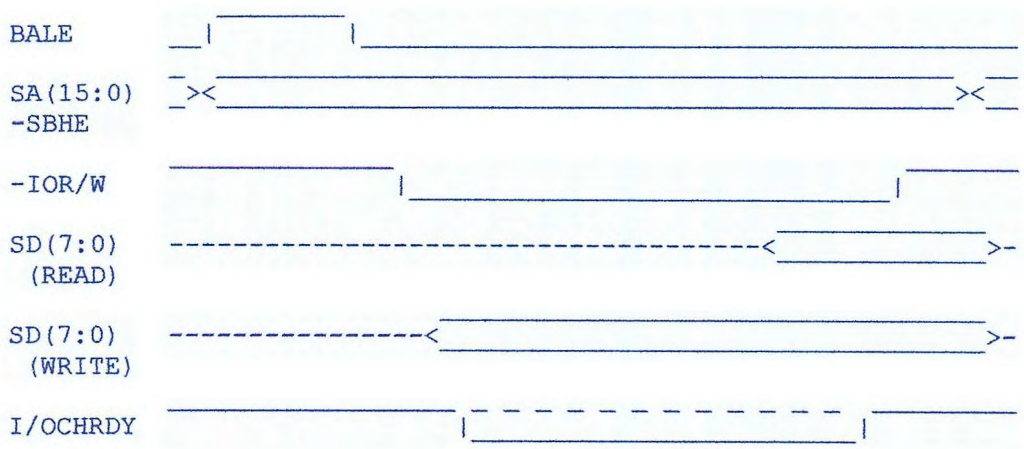
Nota: Todas las señales del Bus ISA emplean niveles TTL standard de Entrada /Salida.

Extension AT de 16 bits:

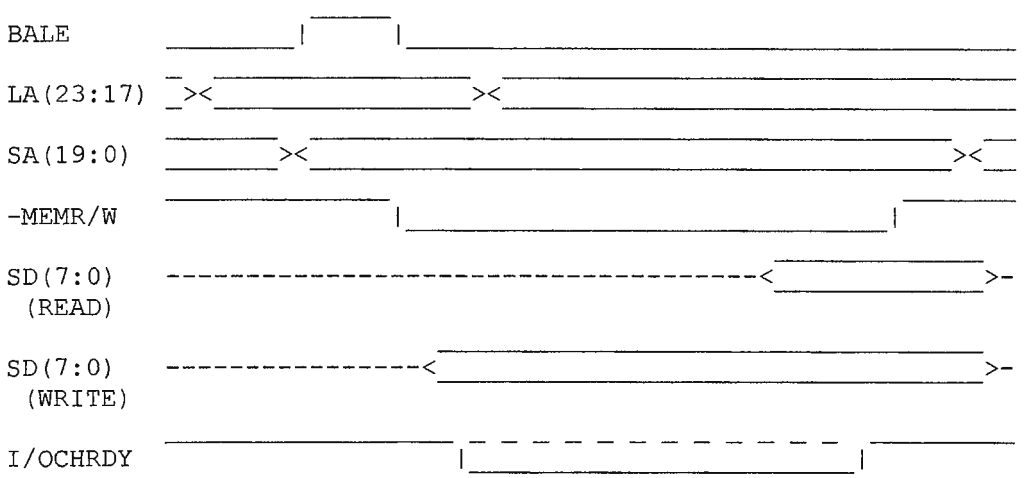
MEM CS16	-|D1	C1|-	SBHE
I/O CS16	-|D2	C2|-	A23
IRQ10	-|D3	C3|-	A22
IRQ11	-|D4	C4|-	A21
IRQ12	-|D5	C2|-	A20
IRQ15	-|D6	C6|-	A19
IRQ14	-|D7	C7|-	A18
-DACK0	-|D8	C8|-	A17
DRQ0	-|D9	C9|-	-MEMR
-DACK5	-|D10	C10|-	-MEMW
DRQ5	-|D11	C11|-	D8
-DACK6	-|D12	C12|-	D9
DRQ6	-|D13	C13|-	D10
-DACK7	-|D14	C24|-	D11
DRQ7	-|D15	C15|-	D12
+5V	-|D16	C16|-	D13
-Master	-|D17	C17|-	D14
Tierra	-|D18	C18|-	D15

- **Diagramas de Tiempo del BUS ISA:**

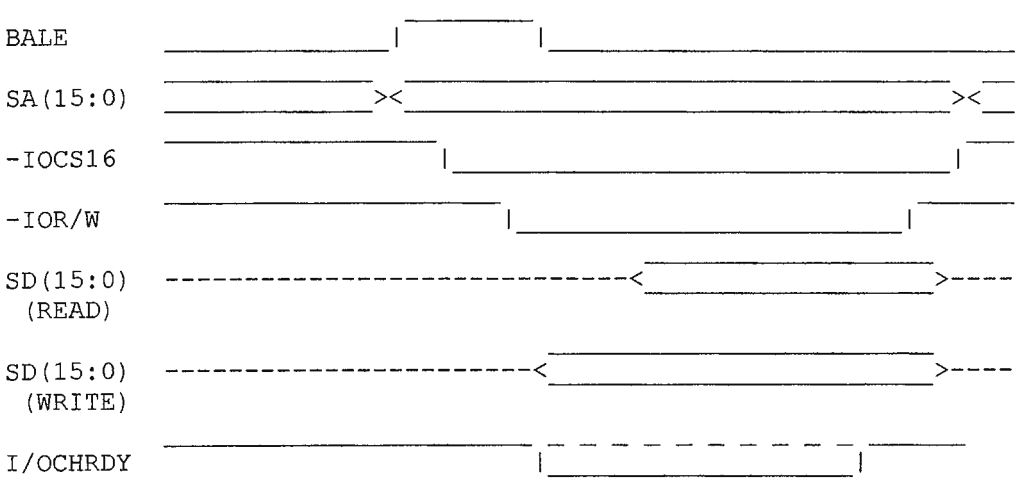
8-Bit I/O Bus Cycles



8-Bit Memory Bus Cycles



16-Bit I/O Bus Cycles



16-Bit Memory Bus Cycles (1 or more Wait States)

```
BALE          _____________________|      |_____________________________
              ___ ___________________________ ___________________________
LA(23:17)     ___><_________________________><_________________________
              _________________ ___________________________________ _
SA(19:0)      _________________><___________________________________><_
              _______ ___________________________
-MEMCS16              |_______________________|
              ___________________________
-MEMR/W                                 |_____________________|  _____
                                                 _______________
SD(15:0)      -------------------------------<_______________>----
 (READ)
                                                 _______________________
SD(15:0)      -------------------------<_______________________>----
 (WRITE)
              ___________________________________ _ _ _ _ _ _ _ _ _ _____
I/OCHRDY                                |_______________|
```

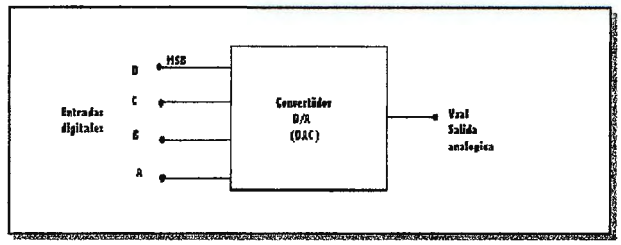
16-Bit Memory Bus Cycles (0 Wait State)

```
BALE          _____________________|      |_____________________________
              ___ ___________________________ ___________________________
LA(23:17)     ___><_________________________><_________________________
              _________________ ___________________________ _________
SA(19:0)      _________________><___________________________><_________
              _______ ___________________________
-MEMCS16              |_______________________|
              ___________________________ ___________________________
-OWS                                    |____|
              ___________________________ _________________
-MEMR/W                                 |____________|
                                                 ______
SD(15:0)      -------------------------------<______>-------------
 (READ)
                                                 _____________
SD(15:0)      -------------------------<_____________>-------------
 (WRITE)
```

Convertidores DA y AD:

CONVERSIÓN DE DIGITAL A ANALÓGICO (D/A).

Básicamente, la conversión D/A es el proceso de tomar un valor representado en código digital (como binario directo o BCD) y convertirlo en un voltaje o corriente que sea proporcional al valor digital. En la siguiente figura se muestra el diagrama a bloques de un convertidor D/A común de cuatro bits.



Las entradas digitales D,C,B y A se derivan generalmente del registro de salida de un sistema digital.

Por cada número de entrada, el voltaje de salida del convertidor D/A es un valor distinto. De hecho, el voltaje de salida analógica Vsal es igual en voltios al número binario. También podría tener dos veces el número binario o algún otro factor de proporcionalidad. La misma idea sería aplicable si la salida del D/A fuese la corriente Isal.

En general: Salida analógica = K x entrada digital

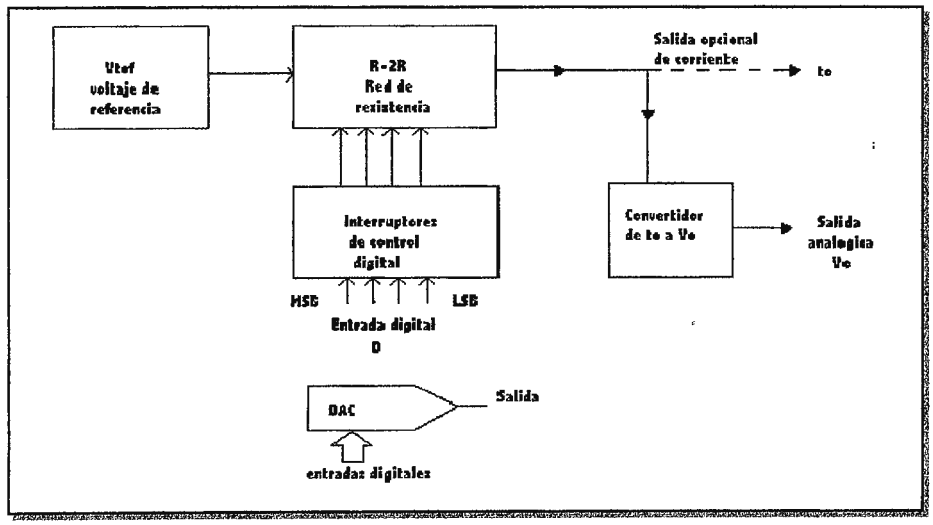
Donde K es el factor de proporcionalidad y tiene un valor constante para un DAC dado. Claro que la salida analógica puede ser un voltaje o una corriente. Cuanto es un voltaje, K tiene unidades de voltaje y, cuando es una corriente, K tiene unidades de corriente. Para el DAC de la figura anterior K=1V así que:

Vsal = (1V) X entrada digital

Se puede utilizar la expresión anterior para calcular Vsal para cualquier valor digital de entrada.

- **Diagrama a bloques:**

Diagrama de bloques y símbolo de circuito para un convertidor digital analógico (DAC) básico.



Salida analógica.

Desde el punto de vista técnico, la salida de un DAC no es una cantidad analógica, ya que sólo puede tomar valores específicos, como los 16 posibles niveles de voltaje para Vsal de la siguiente tabla.

De este modo, y en este sentido, la salida en realidad es digital.

Sin embargo, como se verá más adelante, se puede reducir la diferencia entre dos valores consecutivos al aumentar el número de diferentes salidas, mediante el incremento del número de bits de entrada.

Esto permite producir una salida cada vez más similar a una cantidad analógica, la que cambia de manera continua sobre un rango de valores.

Factores de ponderación de entrada.

Para el DAC de la figura anterior debe observarse que cada entrada digital contribuye con una cantidad diferente de salida analógica. Esto se puede apreciar fácilmente si se examinan los casos donde sólo una entrada es ALTA:

D	C	B	A	Vsal
0	0	0	1	1
0	0	1	0	2
0	1	0	0	4
1	0	0	0	8

A las contribuciones de cada entrada digital se les asignan factores de ponderación según su posición en el número binario. Por lo tanto, A, que es el LSB, tiene un factor de ponderación de 1V, B de 2V, C de 4V y el MSB tiene el mayor, 8V.

Los factores de ponderación se duplican sucesivamente por cada bit, comenzando con el LSB. Por consiguiente, podemos considerar a Vsal como la suma de los factores de ponderación de las entradas digitales.

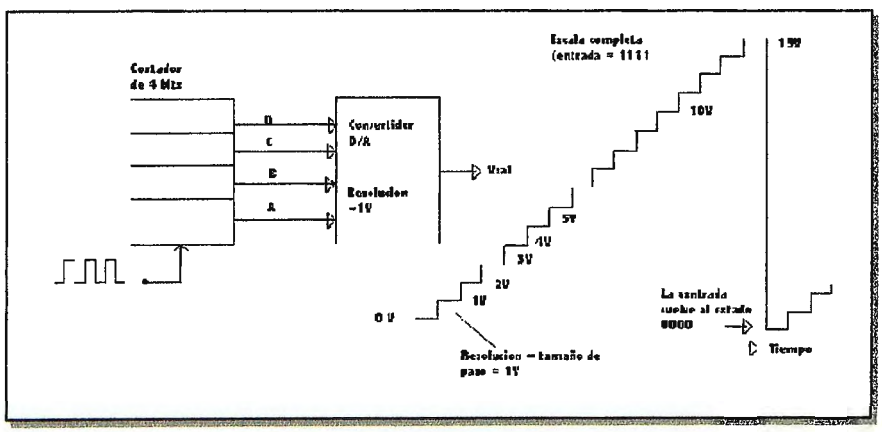
RESOLUCIÓN (tamaño de paso).

La resolución de un convertidor D/A se define como la menor variación que puede ocurrir en la salida analógica como resultado de un cambio en la entrada digital.

Haciendo referencia a la tabla de la siguiente figura, podemos apreciar que la resolución es 1V, puesto que Vsal puede variar en no menos que 1V cuando cambie el código de entrada.

La resolución siempre es igual al factor de ponderación del LSB y también se conoce como tamaño de paso, ya que es la cantidad de Vsal.

A continuación se muestra la figura donde se ilustran estas ideas, donde la salida de un contador binario de cuatro bits son la entradas al DAC.



PORCENTAJE DE RESOLUCIÓN:

resolución porcentual = tamaño de paso X 100 /escala completa

Especificaciones del DAC.

Se dispone de una amplia variedad de DAC como CI o bien como paquetes encapsulados auto contenidos. Uno debe estar familiarizado con las especificaciones más importantes de los fabricantes a fin de evaluar un DAC en una determinada aplicación.

- **Resolución :** Depende únicamente del número de bits. Por esta razón, los fabricantes por lo general especifican una resolución de DAC como el número de bits.

- **Precisión :** Un DAC tiene varias maneras de especificar la precisión. Las dos más comunes se les llama error de escala completa y error de linealidad, que normalmente se expresan como un porcentaje de la salida a escala completa del convertidor (% F.S.).

- **Tiempo de establecimiento:** La velocidad de operación de un DAC por lo general se especifica dando su tiempo de establecimiento, que es el tiempo requerido para que la salida del DAC cambie desde 0 hasta su valor a escala completa cuando todos los bits de la entrada binaria cambian de 0 a 1.
  Los valores más comunes para el tiempo de establecimiento se encuentran entre 50 ns y 10µs.

- **Voltaje de offset:** Desde el punto de vista ideal, la salida de un DAC siempre es cero cuando todos los bits de la entrada son cero. Sin embargo, en la práctica, para esta situación existe un voltaje muy pequeño en la salida, lo cual genera un error de offset. Este error, si no se corrige se suma en todos los casos a la salida esperada del DAC.

- **Monoticidad:** Un DAC es monotónico cuando su salida aumenta a medida que la entrada binaria se incrementa de un valor a otro. Otra forma de describir esta característica es que la salida en escalera no tendrá pasos hacia abajo a medida que la entrada aumenta desde cero hasta su valor de escala completa.

APLICACIONES DEL DAC:

Los DAC se utilizan siempre que la salida de un circuito digital tiene que proporcionar un voltaje o corriente analógico para manejar un dispositivo analógico. Algunas de las aplicaciones más comunes se describen a continuación.

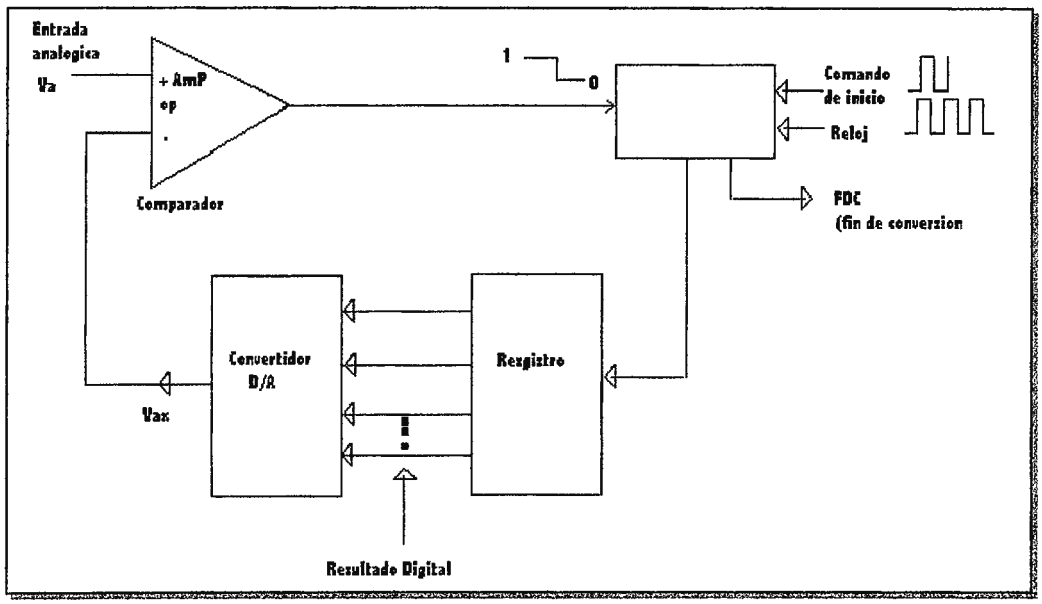
- **Control:** La salida digital de una computadora puede convertirse en una señal de control analógica para ajustar la velocidad de un motor, la temperatura de un horno o bien para controlar cualquier variable física.

- **Pruebas automáticas:** Las computadoras pueden ser programadas para generar las señales analógicas (a través de un DAC) que se necesitan para probar circuitos analógicos. La respuesta de la salida analógica del circuito de prueba normalmente se convertirá en un valor digital por un ADC y se alimentará a la computadora para ser almacenada, exhibida y algunas veces analizada.

- **Reconstrucción de la señal:** En muchas aplicaciones se digitaliza una señal analógica, lo cual significa que varios puntos sucesivos de ésta se convierten en su equivalente digital y se guardan en la memoria. Esta conversión se lleva a cabo con un convertidor analógico - digital (ADC). El DAC puede entonces utilizarse para convertir los datos digitales en memoria de regreso, un punto a la vez, a una señal analógica, reconstruyendo de esta manera la señal original. Esta combinación de digitalización y reconstrucción se emplea en los osciloscopios de almacenamiento digital, los sistemas de audio de disco compacto y la grabación digital de cintas para audio.

CONVERTIDORES DE ANALOGICO A DIGITAL (A/D).

Un convertidor A/D toma un voltaje de entrada analógico y después de cierto tiempo produce un código de salida digital que representa la entrada analógica. El proceso de conversión A/D es generalmente más complejo y largo que el proceso D/A, y se han creado y utilizado muchos métodos.



Varios tipos importantes de ADC utilizan un convertidor D/A como parte de sus circuitos. En la figura anterior se presento un diagrama de bloque general para esta clase de ADC. La temporización para realizar la operación la proporciona la señal de reloj de entrada.
La unidad de control contiene los circuitos lógicos para generar la secuencia de operaciones adecuada en respuesta al comando de INICIO, el cual comienza el proceso de conversión.
El comparador con amplificador operacional tiene dos entradas analógicas y una salida digital que intercambia estados, según la entrada analógica que sea mayor.
La operación básica de los convertidores A/D de este tipo consta de los pasos siguientes:

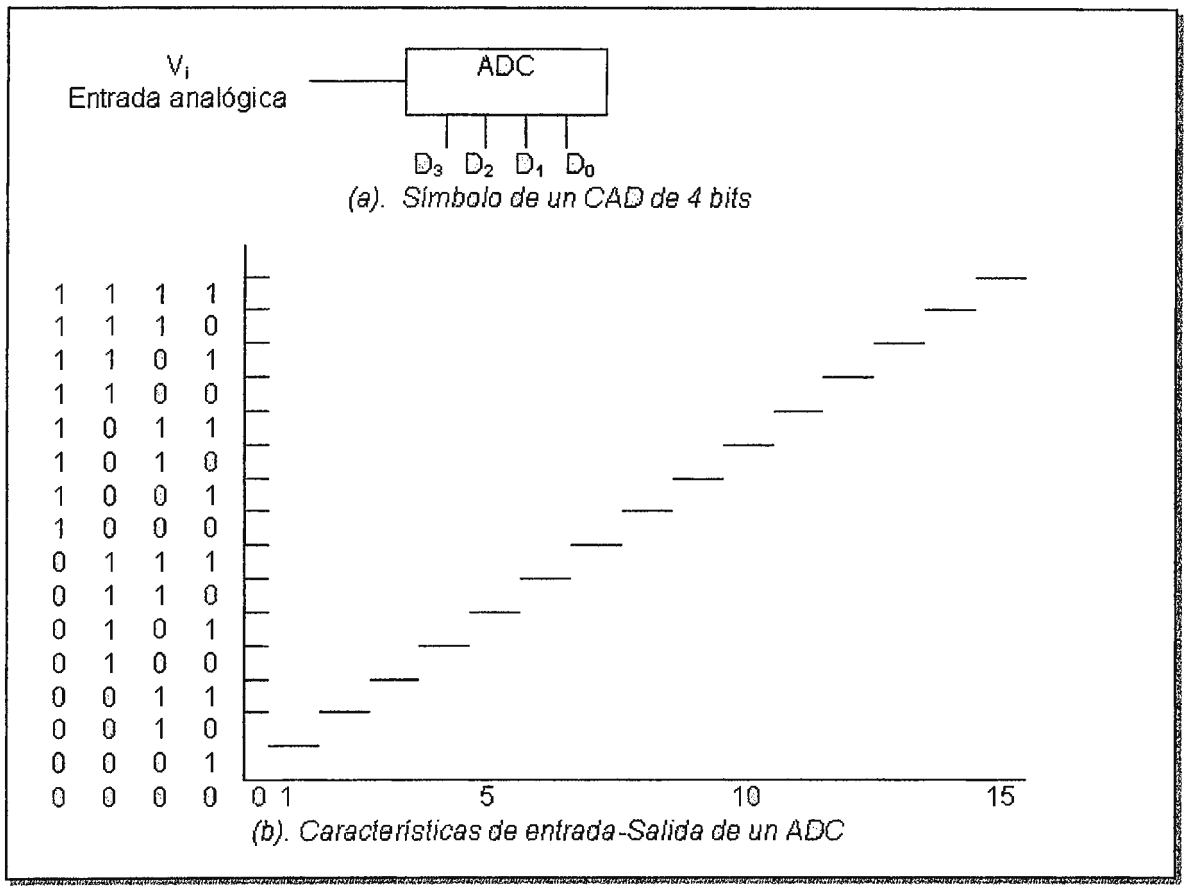
1. El comando de INICIO pasa ALTO, dando inicio a la operación.
2. A una frecuencia determinada por el reloj, la unidad de control modifica continuamente el número binario que está almacenado en el registro.
3. El número binario del registro es convertido en un voltaje analógico, VAX, por el convertidor D/A.
4. El comparador compara VAX con la entrada analógica VA. En tanto que VAX < VA, la salida del comparador permanece en ALTO. Cuando VAX excede a VA por lo menos

en una cantidad = VT (voltaje de umbral), la salida del comparador pasa ABAJO y suspende el proceso de modificación del número del registro. En este punto, VAX es un valor muy aproximado de VA, y el número digital del registro, que es el equivalente digital de VAX, es asimismo equivalente digital de VA, en los límites de la resolución y exactitud del sistema.

5. La lógica de control activa la señal de fin de conversión, FDC, cuando se completa el proceso de conversión.

Características del Convertidor Analógico a Digital.

La salida digital de un convertidor analógico a digital ideal de 4 bits, se gráfica en función del voltaje de entrada analógica en la siguiente. Figura.



La resolución de un convertidor analógico a digital se define de dos maneras. Primero, es el número máximo de códigos de salida digital.

Esta expresión de la resolución del convertidor es la misma que en el caso del convertidor analógico y se repite aquí:

$$\text{Resolución} = 2^n$$

La resolución también se define como la razón de cambio del valor en el voltaje de entrada, Vi, que se necesita para cambiar en 1 LSB la salida digital.

Si se conoce el valor del voltaje de entrada a escala completa, ViFS, que se requiere para producir una salida digital de todos los unos, es posible calcular la resolución mediante:

$$\text{resolución} = ViFS/\ 2^n-1$$

Error de Cuantificación:

En la figura anterior se muestra que la salida binaria es 0101 para todos los valores de Vi entre 4.5 y 5.5 V.

Existe una incertidumbre inevitable respecto al valor exacto de Vi cuando la salida es 0101.

Dicha incertidumbre se especifica como error de cuantificación. Su valor es ±½ LSB. Al incrementar la cantidad de bits se logra una mejor resolución y el error de cuantificación es menor.

Otros Métodos de Conversión A/D.

Existen varios métodos de conversión A/D cada uno con sus ventajas y sus desventajas, a continuación se describen algunos de ellos.

- **ADC de rampa digital ascendente/descendente (ADC de seguimiento).**
  El convertidor ADC de rampa digital es relativamente lento debido a que el contador tiene que volver a ponerse en cero al inicio de cada conversión.
  Utiliza un contador ascendente/descendente para reducir este tiempo desperdiciado, el contador reemplaza al contador ascendente que proporciona las entradas al DAC.

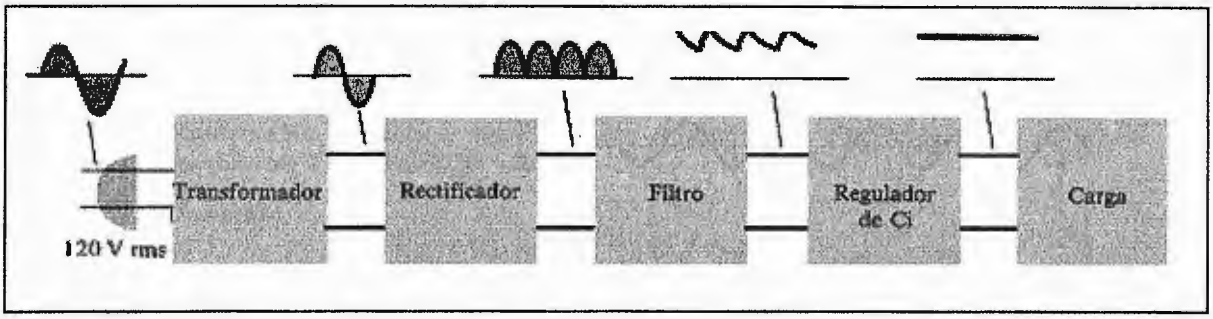
- **ADC de voltaje a frecuencia:** Es más sencillo que los otros ADC, debido a que no hace uno de un DAC.
  En lugar de éste se emplea un oscilador lineal controlado por voltaje que produce como salida una frecuencia que es proporcional al voltaje aplicado en su entrada.
  Una de las principales aplicaciones de este tipo de convertidor se encuentra en ambientes industriales con mucho ruido y donde se tienen que transmitir señales analógicas de poca amplitud desde los circuitos transductores hacia la computadora de control.

- **ADC de doble pendiente:** Es uno de los que tienen mayor tiempo de conversión, pero ofrece la ventaja de un costo relativamente bajo, ya que no requiere de componentes de precisión tales como un DAC o un VCO.
  La operación básica de este convertidor se apoya en la carga y descarga, ambas en forma lineal, de un capacitor mediante el uso de corrientes de valor constante.
  Otra ventaja es su baja sensibilidad al ruido y las variaciones en los valores de sus componentes causados por los cambios de temperatura.

Generalidades de las fuentes de voltaje:

La fuente de voltaje es de gran importancia para poder alimentar un dispositivo o algún equipo, en muchas fuentes de voltaje la salida puede ser regulada o simplemente proporcionan un voltaje fijo, también se encuentran fuentes bipolares.
Para la síntesis de una fuente de voltaje, podemos analizarla por medio de diagramas de bloques, como conjunto de bloques independientes, cuyos parámetros están relacionados, pero que sin embargo se pueden analizar por separado. Los bloques en los que se puede dividir una fuente son: Transformador; rectificador - filtro y regulador de voltaje.

Explicación de los bloques de una fuente de voltaje básica:



- **<u>El bloque transformador</u>:** Es el elemento que suministra la energía a la fuente y suelen estar conectados a 110VRMS o 220VRMS. Generalmente son reductores de tensión y están formados por un devanado primario y uno o más secundarios. El funcionamiento es el siguiente: Cuando se hace circular una corriente alterna por el devanado primario, se origina un flujo magnético en el núcleo, la cual a su vez crea una tensión inducida en el secundario, como es esta tensión inducida directamente proporcional al número de espiras del arrollamiento, modificando la cantidad de éstas se podrá cambiar la tensión inducida de salida en el secundario.

- **<u>El bloque rectificador- filtro</u>:** Este bloque puede ser considerado como dos sub bloques individuales, cada uno de ellos cumpliendo con la función de rectificado uno y filtrado el otro.
  Un rectificador es una disposición de diodos (puede ser uno solo o varios), los cuales tienen la función de convertir la tensión alterna en continúa pulsante. Esto es gracias a que el diodo es un elemento que sólo conduce en un sentido, dejando pasar solamente el semiciclo positivo, o negativo en caso de colocar los diodos en sentido contrario, pero no que cambie de signo.
  La función del filtro es filtrar la tensión pulsante que procede del rectificador con el fin de atenuar las variaciones, de manera que su valor sea lo más continuo posible, con una mínima componente de alterna.
  Todo lo que el filtro haga por eliminar la componente de alterna es beneficioso para la fuente, debido a que el objetivo de ésta es entregar una tensión continua lo más pura posible al regulador de voltaje.

- **Regulador de voltaje:** Un CI regulador de voltaje recibe una entrada de voltaje CD casi constante y lo suministra como salida a un valor un poco más bajo del voltaje CD.
  El valor de voltaje de salida del regulador esta dado por las características del regulador que puede ser un valor fijo o variables; el regulador mantiene fijo o regulado dentro de un amplio intervalo de variación de la corriente de carga o voltaje de entrada. Proporciona una tensión estabilizada de salida a partir de una no regulada que procede del rectificador - filtro.

Fuentes de voltaje digitales:

Las fuentes de voltaje digital están teniendo una gran demanda por la exactitud y rapidez para obtener el voltaje a la salida, estas constan de:

- **Una interfaz:** Que normalmente es un teclado donde el usuario introduce el voltaje que desea.

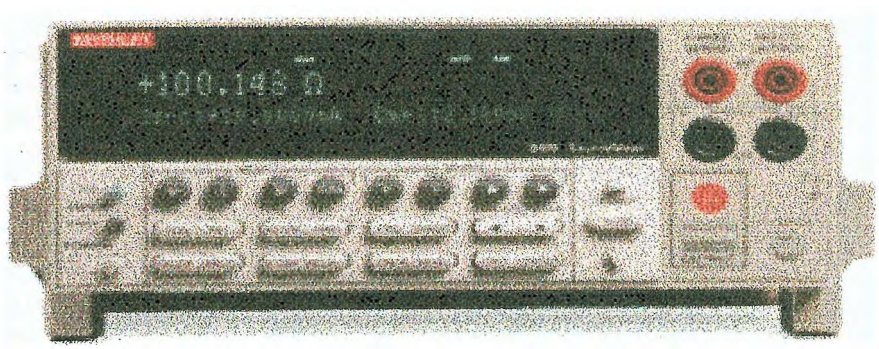
- **Una etapa decodificadora:** Que es la que se encarga de convertir el valor digitado a un valor binario para que pueda ser utilizado por el convertidor.

- **El convertidor análogo a digital:** Este se encarga de convertir el dato binario equivalente digitado por el usuario al valor análogo deseado.

- **Etapa de potencia**: Esta etapa tiene como función entregar la corriente necesaria para la carga.

Muchas fuentes se protegen al detectar una corriente elevada, esta corriente a la que se protege normalmente es fija, pero en algunas fuentes la corriente a la que se protege puede ser programada, con el objetivo de evitar graves daños a la fuente y al dispositivo que se está alimentando.

Fuente de voltaje Digital programable



CONTADORES DE FRECUENCIA (FRECUENCIMETROS).

Un contador de frecuencia es un circuito que puede medir y exhibir de manera visual la frecuencia de una señal. Uno de los métodos más sencillos para construir un contador de frecuencia se muestra en la figura de manera simplificada:

Método básico para el conteo de frecuencia.

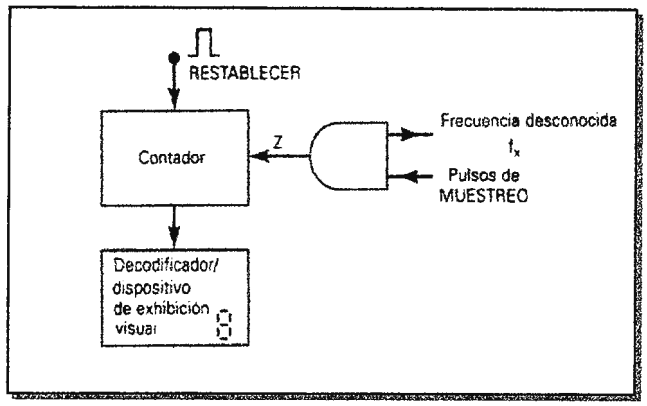
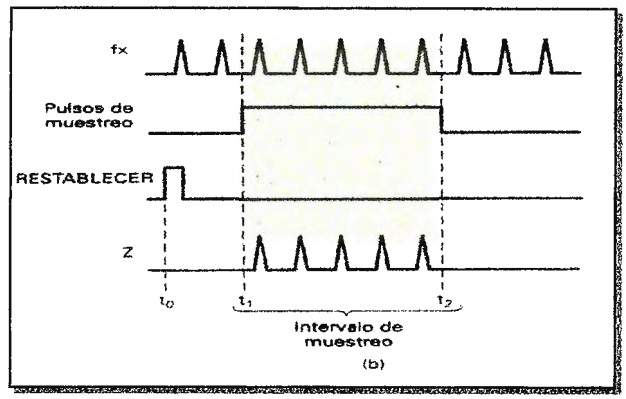


Diagrama de tiempo del contador de frecuencia.

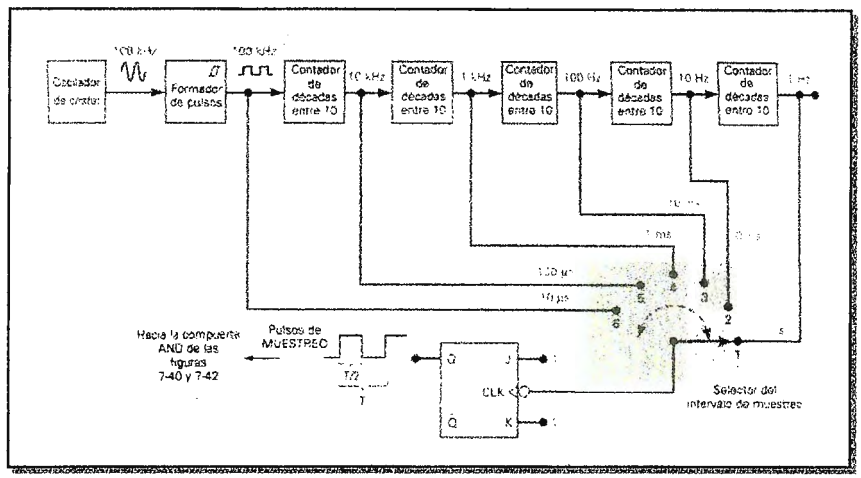


Contiene un contador con su circuitería electrónica decodificadora y de exhibición asociados y una compuerta AND. Las entradas de la compuerta AND incluyen los pulsos cuya frecuencia es desconocida, f_x, y un pulso de MUESTREO que controla el tiempo disponible para que éstos pasen a través de la compuerta AND hacia el contador. Este se halla constituido, por lo general por contadores BCD con conexión en cascada y por la unidad decodificadora y de exhibición que convierte las salidas BCD en una exhibición decimal para su fácil monitoreo.

La forma de onda de la figura anterior muestra que se aplica un pulso RESTABLECER al contador en el tiempo t_0 para iniciar el contador en cero. Antes de t_1 la forma de onda del pulso MUESTREO es BAJA, así que la salida de AND, Z, será BAJA y el contador no funcionara. El pulso de MUESTREO pasa al estado ALTO de t_1 a t_2; a este se le denomina **intervalo de muestreo.** Durante este intervalo los pulsos de frecuencia desconocida pasan a través de la compuerta AND y son contados por el contador. Después de t_2 la salida de la AND retorna abajo y el contador se detiene. De este modo, el contador habría contado el número de pulsos que ocurrieron durante el intervalo de

muestreo y el contenido resultante del contador es una medida directa de la frecuencia de la forma de onda del pulso.
La exactitud de este método depende casi totalmente de la duración del intervalo de muestreo, que debe controlarse con mucha precisión.
El método que se utiliza comúnmente para obtener pulsos de muestreo exactos se ilustra en la siguiente figura:

- **Método para obtener intervalos de muestreo para el contador de frecuencia.**

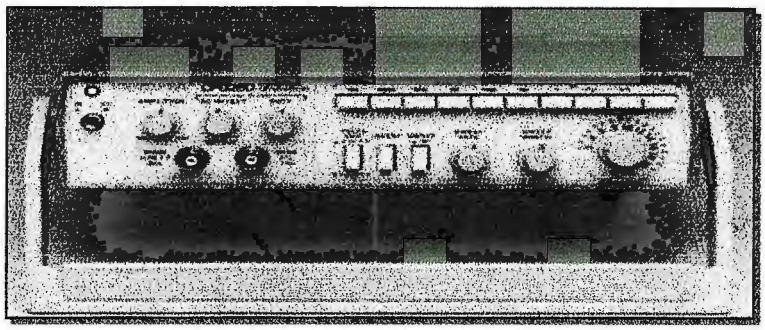


Se utiliza un oscilador controlado por cristal para generar una forma de onda exacta de 100KHz, que se forma en pulsos cuadrados y se alimenta a una serie de contadores de décadas que se utilizan para dividir sucesivamente esta frecuencia de 100KHz entre 10.
Las frecuencias presentes en las salidas de cada contador de decenas son tan precisas (en porcentaje) como la frecuencia del cristal. Estos contadores de décadas suelen ser binarios o Johnson.

El interruptor se utiliza para seleccionar una de las frecuencias de salida del contador de décadas que será alimentada a un FF individual que se dividirá entre 2. Por ejemplo, en la posición 1 del interruptor, los pulsos de 1Hz son alimentados al FF Q, que actúa como un FF de complemento, de manera que su salida será una onda cuadrada con un período de T=2 y una duración de pulso de t_p = T/2= 1s. Esta duración de pulsos es el intervalo de muestreo de 1s que se busca. En la posición 2 el intervalo de muestreo seria 0.1s y así sucesivamente para las otras posiciones.

Generadores de Funciones (Gen – Fun).

Un generador de funciones es un instrumento versátil que genera diferentes formas de onda cuyas frecuencias son ajustables en un amplio rango. Las salidas más frecuentes son ondas senoidales, triangulares, cuadradas, diente de sierra y TTL. Las frecuencias de estas ondas pueden ser ajustadas desde una fracción de hertz hasta varios cientos de kilohertz.



La capacidad máxima del Gen – Fun depende en gran medida del tipo de VCO que se utilice y de los dispositivos lineales usados (AMP-OP multiplicadores análogos, etc). Existen dos tipos de generadores de funciones entre ellos se encuentran los análogos y digitales, siendo la diferencia el tipo de control que se utiliza para cada uno.

Algunas de las funciones disponibles en los generadores de funciones son las siguientes:

1. Botones de Función (Function buttons). Los botones de onda senoidal, cuadrada o triangular determinan el tipo de señal provisto por el conector en la salida principal.

2. Botones de Rango (Range buttons) (Hz). Esta variable de control determina la frecuencia de la señal del conector en la salida principal.

3. Control de Frecuencia (Frecuency Control). Esta variable de control determina la frecuencia de la señal del conector en la salida principal tomando en cuenta también el rango establecido en los botones de rango.

4. Control de Amplitud (Amplitude Control). Esta variable de control, dependiendo de la posición del botón de voltaje de salida (VOLTS OUT), determina el nivel de la señal del conector en la salida principal.

5. Botón de inversión (Invert button). La señal del conector en la salida principal se invierte.

La frecuencia y amplitud de la señal de un GEN – FUN análogo se controla utilizando un vernier y conmutador para la escala. Es utilizado en algunos casos en combinación con un frecuencímetro para contabilizar la frecuencia. La desventaja es el tiempo que le lleva al usuario calibrar los parámetros requeridos.

Un GEN – FUN digital puede ser programado (frecuencia, amplitud y tipo de señal) desde un teclado y no necesita un frecuencímetro ya que los presentadores del teclado

representan el valor de frecuencia y amplitud que se necesita, lo cual representa una ventaja. Otra ventaja que se puede incluir en algunos Gen – Fun es el control automático de las escalas (auto rango).

Osciladores controlados por voltaje (VCO)

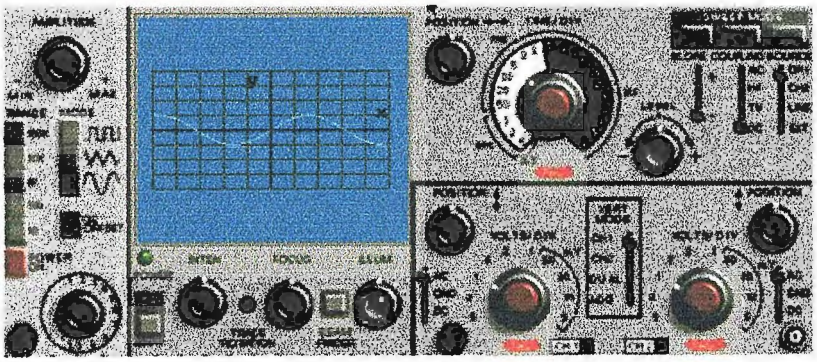
Un oscilador es un circuito que es capaz de generar a su salida una forma de onda estable, periódica y con una frecuencia determinada. Para poder ser utilizado como generador de funciones; un oscilador debe ser controlable. Existen osciladores controlados por corriente o tensión (en los que la frecuencia de oscilación es proporcional a un voltaje de entrada al circuito), y osciladores controlados digitalmente (en los que la frecuencia de oscilación es proporcional a un número binario calculado por un sistema digital).

Introducción a los osciloscopios.

La palabra Osciloscopio describe un instrumento que permite observar oscilaciones (por oscilación se entiende cualquier vibración o cambio en un fenómeno físico que se pueda convertir en señal eléctrica proporcional a ella). El osciloscopio gráfica los cambios de amplitud de una señal con respecto al tiempo, por lo tanto permite determinar amplitud, período, nivel DC entre otras características.

Sincronismo

Los circuitos de sincronismo son los encargados de que el trazo en la pantalla sea estable. La estabilidad de la imagen es necesaria para que la señal se pueda medir y detallar.



La fuente de sincronismo puede ser la señal proveniente del amplificador vertical, una señal externa o la línea entre otras. El circuito de disparo toma la fuente de sincronismo y la compara con un nivel de voltaje y una pendiente de referencia. Cuando son iguales genera un pulso llamado pulso de sincronismo, el cual sirve como entrada para el circuito de compuerta.

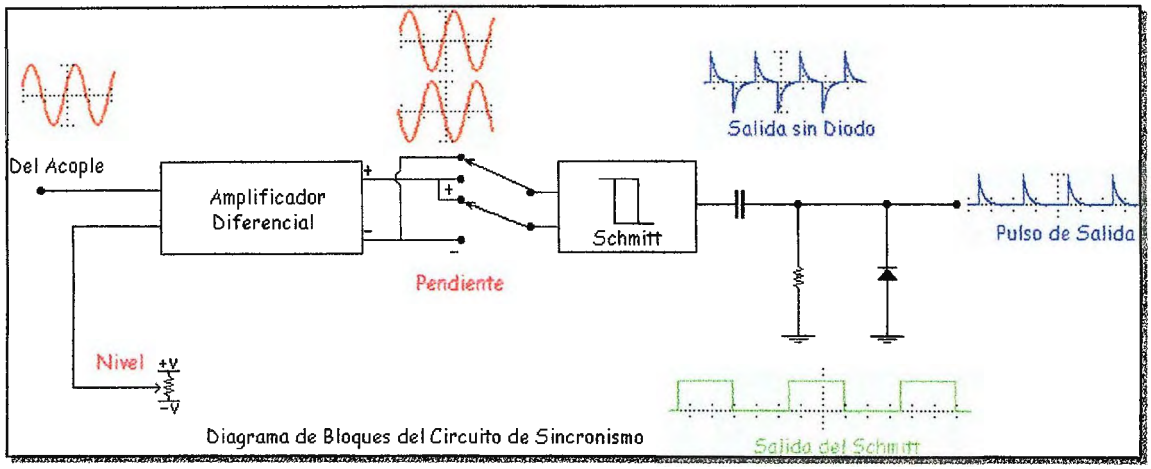


Diagrama de Bloques del Circuito de Sincronismo

Algunos osciloscopios cuentan con acople de la fuente de sincronismo antes de ser comparada, algunos de estos acoples son: DC, LF, HF, TV line y TV field. El acople DC deja pasar la señal sin hacerle ningún cambio. Los acoples LF y HF sirven para rechazar ruido de alta frecuencia y suprimir componentes indeseados de baja frecuencia respectivamente. Sin embargo, las funciones de los acoples LF y HF pueden intercambiarse dependiendo del fabricante. Los acoples TV line y TV field son más complicados, ya que toman una señal de televisión y generan el disparo basados en su distribución interna. Existe otro acople llamado AC, éste funciona en forma similar al HF (o al LF), pero la frecuencia de corte es mucho más baja, del orden de 10 Hz. El acople AC prácticamente filtra únicamente la componente DC de la señal.

Para evitar que en la comparación se generen varios pulsos, se puede utilizar un comparador Schmitt Trigger. El circuito de sincronismo hace uso de un comparador Schmitt Trigger, un amplificador diferencial, un interruptor de pendiente, un filtro pasa altos y un diodo.

El nivel puede estar determinado por el usuario o por un valor fijo dependiendo de la posición del selector de modo disparo. Cuando se utiliza el modo ***fijo*** (fix o fixed), el nivel se toma como el valor promedio de la señal de entrada. Cuando se utiliza el modo ***normal*** el nivel está determinado por el usuario. En el modo ***automático***, el nivel también lo determina el usuario pero en el caso de que no se presente disparo, debido a que el nivel seleccionado nunca es igualado por la señal, ésta aparece visible en la pantalla aunque de manera inestable. Si lo anterior ocurriera en modo normal, no se generaría ningún pulso de disparo y no aparecería nada en la pantalla. Cuando el circuito de disparo se encuentra en la posición ***single***, no se tiene en cuenta la fuente de sincronismo, el nivel ni la pendiente. Sólo se tiene un disparo válido cuando el botón single es oprimido. Este modo de disparo se utiliza cuando se desea observar una señal no periódica.

Osciloscopios de Almacenamiento Análogo

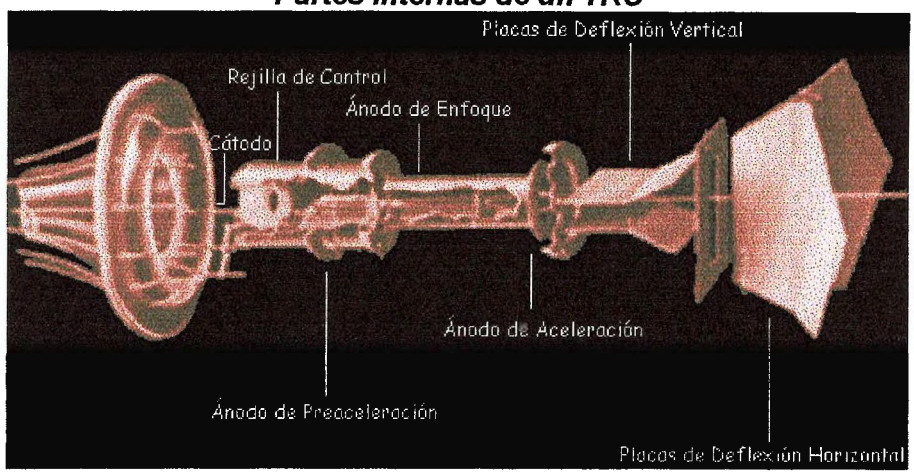
Para algunas aplicaciones de baja frecuencia o para aplicaciones de eventos esporádicos un osciloscopio normal no permite ver y/o medir con claridad la forma de onda de las señales, debido a que el haz de electrones no ha terminado el trazo cuando la parte inicial del mismo ya se ha borrado. Por ello es necesario almacenar la forma de onda para su posterior estudio y medición.

En estos casos no basta con el tipo de fósforo que se utilice, sino que se hace necesario adoptar mecanismos especiales como almacenamiento biestable, de persistencia variable o de transferencia de carga.

Tubo de Rayos Catódicos

El Tubo de Rayos Catódicos (TRC) es el dispositivo mediante el cual el osciloscopio muestra la señal. Todas sus entradas son eléctricas, mientras que su salida es una representación gráfica de la señal de entrada. Está constituido por el tubo al vacío, cañón electrónico, sistema de deflexión, sistema acelerador, postdeflexión, pantalla y gratícula.

Partes internas de un TRC



Deflexión Vertical

Los circuitos de deflexión vertical reciben la señal de entrada, la adaptan y entregan a las placas de deflexión vertical una señal amplificada y de nivel apropiado. Este procedimiento debe satisfacer las especificaciones de ancho de banda del instrumento y los tiempos requeridos para el buen funcionamiento del mismo.

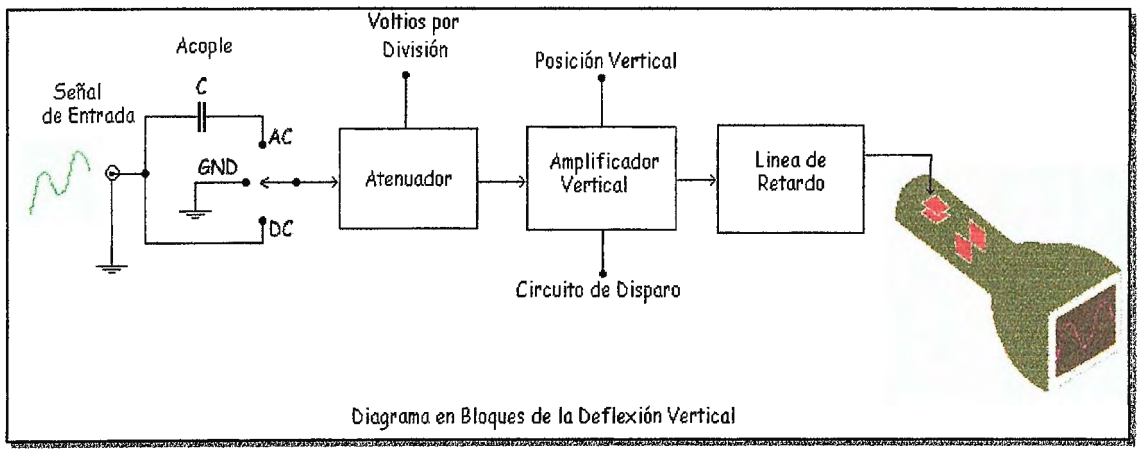


Diagrama en Bloques de la Deflexión Vertical

El bloque de acoplamiento tiene tres posiciones que son AC, DC y GND. Cuando el acople está en AC el condensador quita la componente DC de la señal de entrada. Cuando el acople está en DC, la señal de entrada pasa igual al siguiente bloque y cuando el acople se coloca en GND, se conecta internamente la entrada a un potencial de cero voltios; es decir, se conecta a tierra. En este ultimo caso, el osciloscopio muestra cero voltios, éste no coloca en corto la entrada, ya que la conexión es interna y no externa, evitando daños en el circuito en el cual se está realizando la medición.

Deflexión Horizontal.

La deflexión horizontal del haz de electrones debe llevar una velocidad constante; de esta manera, cada división en la pantalla corresponde a intervalos iguales de tiempo. Con la escala de tiempo (Tiempo por División) se determina la velocidad, con la cual el haz de electrones recorre cada división horizontal y por tanto la pantalla completa de izquierda a derecha.

La velocidad constante del haz se genera colocando una diferencia de potencial entre las placas de deflexión horizontal en forma de rampa, ya que la deflexión es directamente proporcional al voltaje colocado entre las placas de deflexión. Los circuitos de deflexión horizontal tienen como entrada el circuito de disparo y como salida las placas de deflexión horizontal del TRC.

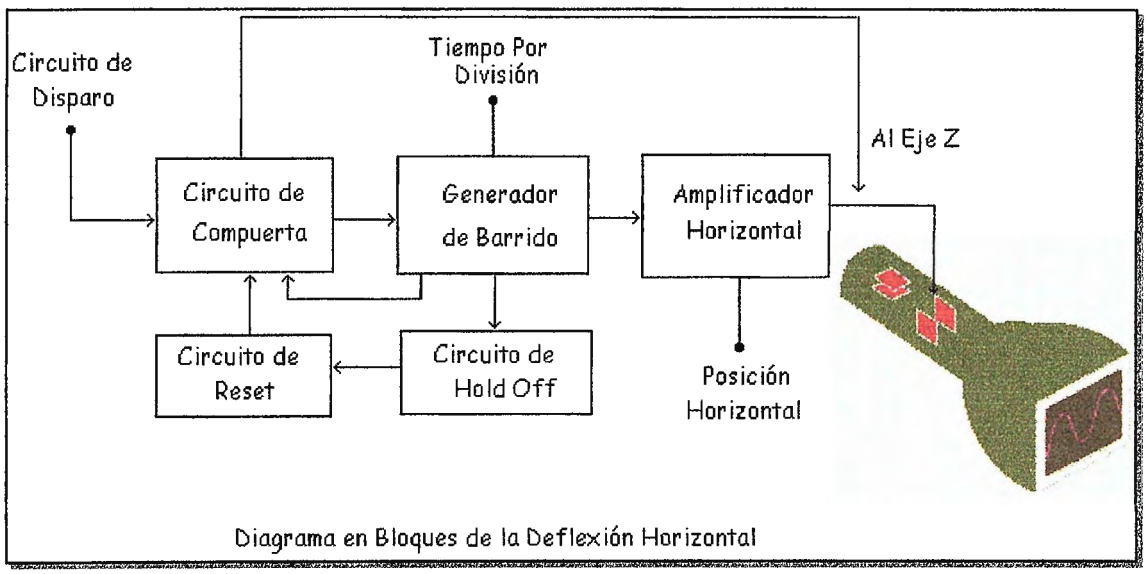
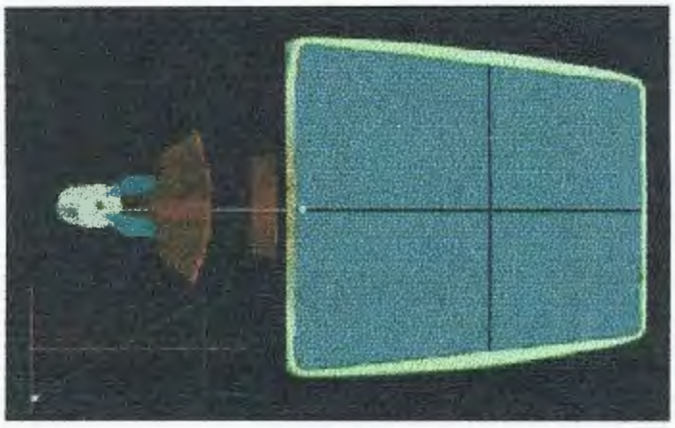


Diagrama en Bloques de la Deflexión Horizontal

Generador de Barrido

El generador de barrido es el encargado de generar la rampa de voltaje. Esto lo hace cargando un condensador con una fuente de corriente constante. La rampa de voltaje siempre tiene valores positivos, pero en realidad la diferencia de potencial en las placas de deflexión debe ir de voltajes negativos, trazo a la izquierda de la pantalla, hasta voltajes positivos, trazo a la derecha de la pantalla.

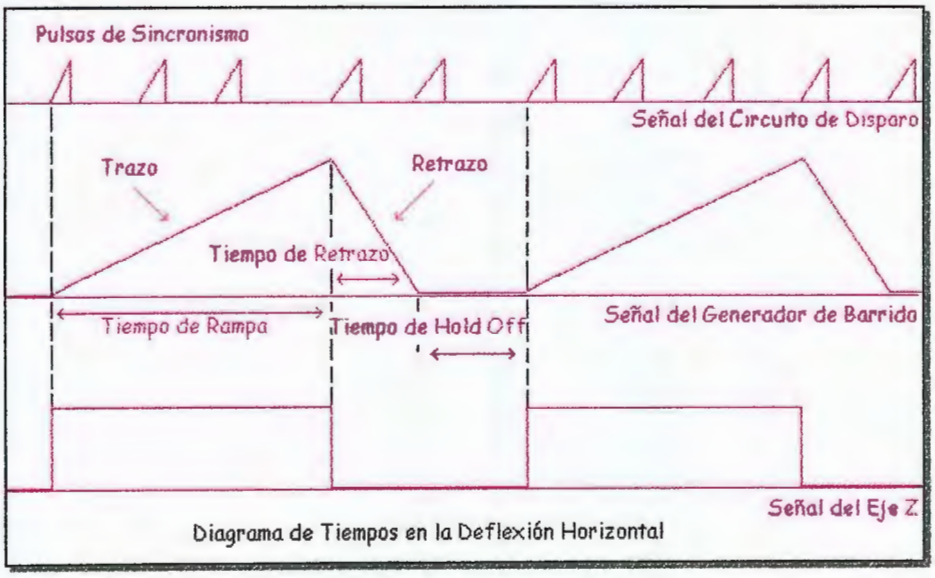
- **Variación del haz de electrones con respecto al voltaje de Barrido**



Para que la señal pueda ser visible, el trazo debe repetirse constantemente de tal forma que se vea una línea continua en la pantalla, además debe estar sincronizado con la señal de entrada para que se trace la misma trayectoria y se vea una imagen estable. Para lograr que se repita el trazo se debe comenzar de nuevo e indefinidamente la rampa de voltaje. El proceso de devolver el haz es llamado retraso o ***fly back***. Cuando se inicia el trazo el generador de barrido le indica al circuito de compuerta que deshabilite el haz de electrones de tal forma que el TRC se encuentre en corte.

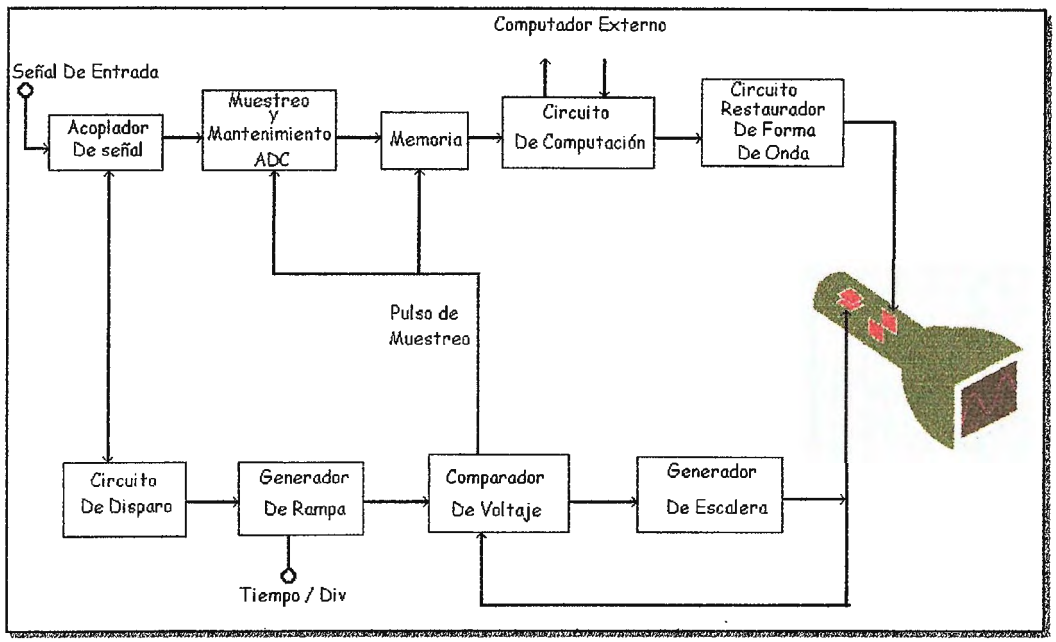
Hold-Off, Reset y Circuito de Compuerta

El circuito de **hold-off** sirve para que el usuario determine un tiempo muerto, que puede ser cero, después de que el retraso ha terminado.



Los osciloscopios digitales al igual que los osciloscopios análogos, son instrumentos que permiten visualizar y medir una señal eléctrica, con la diferencia que utilizan la digitalización de la señal de entrada. La digitalización consiste en asignar un código binario a muestras que se toman de la señal de entrada análoga. Estos códigos se

ordenan y almacenan secuencialmente de modo que pueda reproducir la señal de entrada en cualquier momento. Para realizar la digitalización se necesita un circuito que obtenga las muestras y un conversor análogo-digital (ADC); para su almacenamiento es necesario una memoria. Después que los datos están almacenados, se pueden hacer diferentes tipos de cálculos realizados por el mismo osciloscopio o asistido por computador. El control de disparo, la deflexión horizontal, la sincronización, etc., se realiza con bloques especializados.



El bloque acoplador entrega una señal análoga adecuada para el circuito de muestreo; además, para la sincronización se comunica con el circuito de disparo. El circuito de muestreo y mantenimiento toma una muestra del voltaje de entrada cuando el comparador le envía un pulso. La muestra la almacena en un condensador con bajas perdidas. Este voltaje debe ser constante durante el tiempo que necesite el ADC, de tal manera que la conversión sea correcta. Para realizar esto se cuenta con buffers que son dispositivos activos que presentan muy alta impedancia de entrada, ganancia unitaria y muy baja impedancia de salida. De esta manera no cargan al circuito acoplador de señal y reducen el tiempo de transferencia del voltaje al condensador.

Métodos de Muestreo

Una señal se puede muestrear de dos formas diferentes: *en tiempo real* o en tiempo equivalente. El muestreo en tiempo real consiste en tomar muestras consecutivas de una señal durante un solo barrido, mientras que el muestreo en tiempo equivalente toma muestras, pero en diferentes barridos. El muestreo en tiempo equivalente sólo se aplica a señales periódicas y las muestras pueden tomarse secuencial y ordenadamente una por cada período.

Existe también el muestreo equivalente aleatorio. En éste el instante en que se toma la muestra no lo determina el comparador, sino un reloj interno independiente. De esta manera se toman muestras aleatorias de la señal durante diferentes períodos y se almacenan en posiciones de memoria que correspondan a las muestras que se están

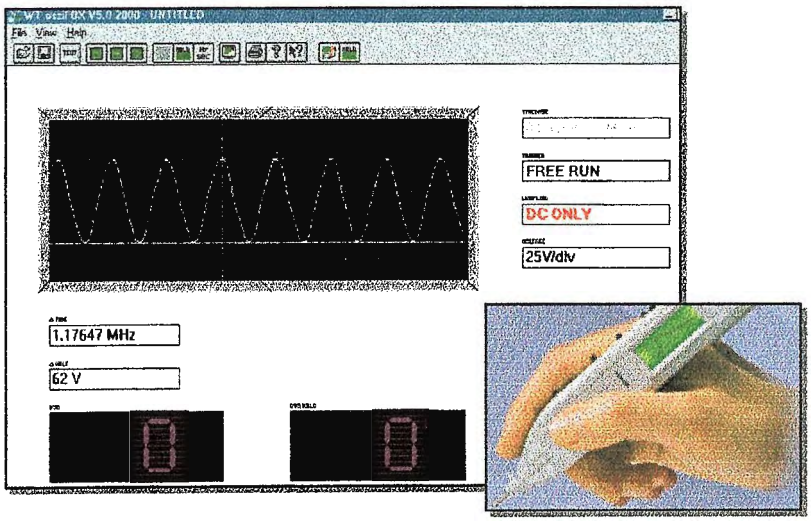
tomando. En conclusión, los osciloscopio con muestreo en tiempo equivalente pueden visualizar formas de ondas de mucha mayor frecuencia que la frecuencia de muestreo. Cuando la frecuencia de muestreo es comparable con la frecuencia de la señal de entrada, la visualización en la pantalla aparece como unos puntos dispersos. Este error es llamado percepción cambiada y genera mediciones erróneas de la frecuencia de una señal. La solución para ello es utilizar un esquema de ***interpolación.*** Interpolar es adicionar puntos entre la muestras almacenadas en memoria.

En resumen, el ancho de banda real de un osciloscopio digital depende de muchos factores como la frecuencia de muestreo, el sistema interpolador, la forma de onda y el tipo de muestreo entre otras. Por esta razón es muy útil conocer la frecuencia de muestreo de un osciloscopio digital para determinar si está en capacidad o no de mostrar fidedignamente una señal en particular.

Osciloscopios de nueva generación:

En la actualidad sé esta abriendo paso a una nueva generación de osciloscopios digitales (similar a la transición de los multímetros análogos a digitales); desde aquellos que son portátiles y compactos hasta aquellos que pueden ser conectados y manejados mediante una computadora personal, con todas las ventajas que está ofrece al usuario. Estos tipos de osciloscopios brindan una mayor exactitud, rapidez, confiabilidad de la información y presentación de la misma. El usuario puede analizar características importantes de las señales, sin necesidad de hacer muchos cálculos siendo el osciloscopio el que realice de forma automática todas las operaciones.

A continuación se presenta dos tipos de osciloscopios de nueva generación:



ADQUISICIÓN DE DATOS

Existen muchas aplicaciones en que los datos analógicos tienen que ser digitalizados (convertidos en digital) y transferidos a la memoria de una computadora. El proceso por el cual la computadora adquiere estos datos analógicos digitalizados se conoce como adquisición de datos. La computadora puede ejecutar varias tareas con los datos, según la aplicación. En una aplicación de almacenamiento, como una grabación de audio digital

o un osciloscopio digital, la macrocomputadora interna almacenará los datos y un tiempo después los transferirá a un DAC para reproducir los datos analógicos.

En una aplicación de control de proceso, la computadora puede examinar los datos o realizar cálculos con ellos para determinar que salidas de control generan.

Generalidades de los multimetros.

Multimetros análogos

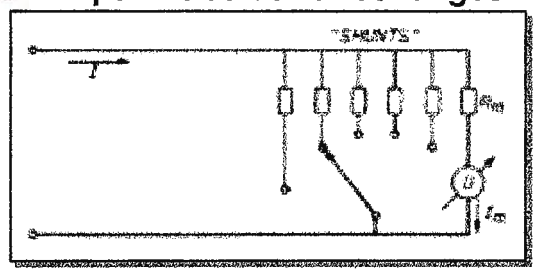
La totalidad de los amperímetros, voltímetros y óhmetros de tipo análogo son fabricados a partir de un galvanómetro de D´Arsonval y un conjunto de resistencias acompañadas algunas veces de una fuente de poder. Si se añaden resistencias en serie obtenemos un voltímetro, si se añaden en paralelo obtenemos un amperímetro y si a uno de los arreglos conectamos una fuente fácilmente se obtiene un óhmetro.

Amperímetro Analógico

Un amperímetro siempre se conecta en serie con una rama del circuito y mide la corriente que pasa a través de él. Un amperímetro ideal sería capaz de efectuar la medición sin cambiar o perturbar la corriente en la rama. Sin embargo, los amperímetros reales poseen siempre algo de resistencia interna y hacen que la corriente en la rama cambie debido a la inserción del medidor.

Los amperímetros electromecánicos industriales y de laboratorio se emplean para medir corrientes desde 1m A hasta varios cientos de amperes. El paso de una corriente por la bobina de un galvanómetro genera un movimiento de la aguja proporcional a tal corriente. Para construir un amperímetro se emplea en la mayoría de los casos una llave selectora para conectar diferentes resistencia en paralelo con el galvanómetro llamadas comúnmente shunts, permitiendo de esta manera hacer lecturas de corrientes más grandes que las que puede hacer el galvanómetro solamente, que normalmente son pequeñas. El cambio de shunt debe hacerse sin corriente, ya que sin estas resistencias toda la corriente pasaría por el galvanómetro causándole daños graves.

Amperímetro de varios rangos



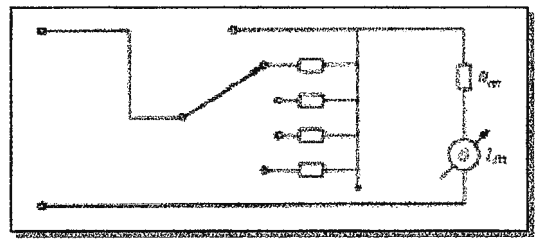
Los medidores típicos para bancos de laboratorio tienen exactitudes de aproximadamente 1 % del valor de la escala completa debido a las inexactitudes del movimiento del medidor. Además de este error, la resistencia de la bobina del medidor introduce una desviación con respecto al comportamiento de un amperímetro ideal, este instrumento es ideal cuando su resistencia interna es cero, es decir si actúa como un corto entre los puntos del circuito donde se desea realizar la medición.

Voltímetros Analógicos.

Un voltímetro se conecta en paralelo con los elementos que se miden. Mide la diferencia de potencial (voltaje) entre los puntos en los cuales se conectan. Al igual que el amperímetro ideal, el voltímetro ideal no debería hacer cambiar la corriente y el voltaje en el circuito que se está midiendo. Esta medición ideal del voltaje sólo se puede alcanzar si el voltímetro no toma corriente alguna del circuito de prueba.

La mayor parte de los voltímetros emplean el movimiento de D' Arsonval. Este movimiento se puede considerar en sí mismo un voltímetro, si se considera que la corriente que pasa por él, multiplicada por su resistencia interna origina una determinada caída de voltaje. En el caso del voltímetro el instrumento es ideal si ofrece resistencia infinita entre los puntos sobre los cuales sé esta realizando la medición del voltaje, es decir constituye un circuito abierto entre sus puntas de prueba, pero esto es difícil de lograr por lo que para aumentar el voltaje que se puede medir mediante ese instrumento, se agrega una resistencia más en serie a la resistencia propia del medidor. La resistencia adicional (que se llama un multiplicador) limita la corriente que pasa por el circuito del medidor.
Para construir un voltímetro de múltiple rango, se puede emplear un interruptor que conecte resistencias de varias magnitudes en serie con el movimiento del medidor. Para obtener una deflexión hacia los valores altos de la escala, los bornes se deben conectar con el voltímetro con la misma polaridad que las marcas de las terminales. Los voltímetros típicos de corriente directa (CD) de laboratorio tienen exactitudes de ± 1 % de la escala completa.
La sensibilidad de un voltímetro se puede especificar por el voltaje necesario para una deflexión de escala completa. Pero otro criterio de sensibilidad, que se usa ampliamente, es la capacidad de ohmios por voltios.

Voltímetro de varios rangos



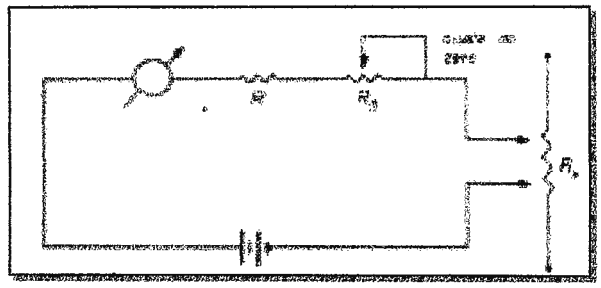
Óhmetro:

Es un instrumento que mide la resistencia o simplemente continuidad de un circuito o parte de él, directamente en ohmios sin necesidad de cálculos, su principio de funcionamiento se basa en el método del voltímetro para medir resistencias y se configura habitualmente en circuitos tipo serie y/o derivación.

Óhmetro Tipo Serie:

El óhmetro tipo serie consta de un galvanómetro o movimiento D`Arsonal conectado en serie con una resistencia y una batería, con un par de terminales a las cuales se conecta la resistencia desconocida. La corriente que circula a través del galvanómetro depende de la magnitud de la resistencia desconocida y la indicación del medidor es proporcional a su valor, siempre y cuando se hayan tomado en cuenta una debida calibración

Óhmetro tipo serie.



Cuando la resistencia desconocida Rx = 0 (terminales A y B en cortocircuito), circula corriente máxima en el circuito. En estas condiciones, la resistencia de derivación R2 se ajusta hasta que el galvanómetro indique la corriente a escala completa (Ifsd). La posición de la aguja para la corriente de escala completa se marca "0Ω ". En forma similar, cuando Rx = ∝ (terminales A y B abiertas) la corriente en el circuito es cero y el galvanómetro indica cero corriente, esta posición se marca ∝ en la escala. Se colocan las marcas intermedias en la escala conectando valores conocidos de resistencia Rx en las terminales del instrumento. La exactitud de estas marcas depende de la exactitud respectiva del galvanómetro y de las tolerancias de las resistencias de calibración.

Aún cuando el óhmetro tipo serie es un diseño popular y se utiliza extensamente en los instrumentos portátiles para servicio general, tiene ciertas desventajas. Las más importantes se relacionan con la disminución del voltaje de la batería interna con el tiempo y el uso, de forma que la corriente a escala completa disminuye y el medidor no lee "0" cuando A y B están en cortocircuito. La resistencia de derivación R2 provee un ajuste para contrarrestar el efecto de la descarga de la batería. Es posible ajustar la aguja a escala completa con R1 eliminando a R2; pero esto cambiaría la calibración en toda la escala. El ajuste de R2 es una mejor solución, ya que la resistencia equivalente del paralelo de R2 y la bobina Rm siempre es baja comparada con R1, y por consiguiente el cambio requerido en R2 para el ajuste no cambia mucho de calibración.

Una cantidad conveniente al uso en el diseño de un óhmetro tipo serie es el valor de Rx que origina media deflexión en el medidor. A esta posición, la resistencia a través de las terminales A y B se define como la resistencia de media escala Rh. El circuito es analizable a partir de la corriente a escala completa Ifsd y la resistencia interna del galvanómetro Rm, se reduce la corriente a 1/2 Ifsd y la resistencia desconocida debe ser igual a la resistencia interna total del óhmetro.

Multimetros digitales.

Los multimetros digitales utilizan diferentes técnicas para transformar la variable eléctrica que se mide, en un valor con características digitales que puede ser utilizado para presentar dichas variables en forma de dígitos en una pantalla. Para lograrlo se hace necesario utilizar convertidores análogo a digital, pero el principio sigue siendo el mismo que se utiliza en los multimetros análogos.

En la actualidad se encuentran multímetros digitales muy precisos, ya que los multímetros análogos, aparte del porcentaje de error que ellos proporcionan, está el error que le agrega el usuario a la hora de tomar la lectura, además los multímetros digitales traen

otras funciones que no traen los multímetros análogos, como es inductancia, capacitancia, probadores de diodos y medidores de frecuencia, etc.

Existen multímetros de nueva generación, que pueden ser conectados a la computadora para lograr funciones más complejas como por ejemplo realizar cálculos matemáticos mas precisos y desplegar gráficas, con mayor calidad, a demás los datos pueden ser almacenados para tener estadísticas de las mediciones realizadas.

- **Ejemplos de multímetros de nueva generación.**

Sistema de protección en equipos de laboratorio.

Uno de los mayores problemas que se ha tenido y que aún no se ha podido resolver es la vulnerabilidad de los equipos de laboratorio al sufrir daños por usos inadecuados y que son ocasionados por personas que empiezan a utilizar los instrumentos de medición (osciloscopio, multímetro, frecuencímetro, generador de señales, fuente de alimentación etc.), también aquellos errores que surgen por la ignorancia, falta de experiencia y la falta de concentración en un momento dado.
Actualmente este tipo de problemas se da con mucha frecuencia, lo cual implica gastos innecesarios a corto y largo plazo, donde los equipos con que cuentan universidades, colegios, institutos técnicos, centros de capacitación, entre otros, se encuentran expuestos a sufrir daños parciales (en cuyo caso el equipo cuente con protección para alguno de sus componentes) y en el peor de los casos totales. Por consiguiente, no se logra satisfacer en su totalidad las necesidades de los estudiantes, por la carencia de dichos instrumentos. En otros lugares ni siquiera se cuenta con un equipo auxiliar de medición para realizar las prácticas de laboratorio, por ser estos muy costosos, difíciles de adquirir y manipular. En cuyo caso se cuente con equipos, en la mayoría de los casos son muy viejos, ocupan mucho espacio en el lugar de trabajo por no estar en un mismo módulo de operaciones, si se tiene la suerte de que sea equipo donado; este equipo no es precisamente el que se necesita en el centro educativo y es por estas razones que existe una necesidad de actualizarlo.
A continuación se presentan los rangos y características del PU - 2200. Este es el equipo que se utiliza en la Universidad Don Bosco, en la mayoría de las prácticas en las áreas básicas de electrónica.

Especificaciones generales del PU - 2200

INSTRUMENTO	ESPECIFICACIONES
FRECUENCIMETRO	Fc = $1H_Z$ ~ $100KH_Z$ Período = 10µS ~ 1s
GENERADOR DE FUNCIONES	F = $0H_Z$ ~ $100KH_Z$ V = 16vpp Con 5 rangos de operación y una salida auxiliar para TTL
FUENTE DE VOLTAJE	+5V: 5A +5V: 200mA +12V: 116mA -12V: 55mA
FUENTE DE VOLTAJE VARIABLE	V = ± 12V: 1A

Otras características e instrumentos externos al PU - 2200.

Multímetro:

El PU - 2200 cuenta con un multímetro que tiene las funciones básicas de operación como: voltímetro, amperímetro, óhmetro, etc. Estos multímetros sufrén averías muy comúnmente por no estar protegidos debidamente y sus fallas van desde un fusible quemado hasta provocar daños permanentes al circuito intergrado principal del multímetro (solamente este IC anda por el precio de unas cuantas centenas de colones). Incluso hace un par de ciclos (en la UDB) se tuvieron que cambiar en su mayoría, por estar fuera de uso o por no funcionar correctamente alguno de sus componentes, lo cual implica gastos y hasta la fecha hay algunos que no se han cambiado o que carecen de multímetro; esto se debe a la carencia de protección mencionada anteriormente.

Osciloscopio:

El PU - 2200 no tiene incluido su propio osciloscopio (de hecho la mayoría de equipos de medición no cuentan con este componente de esencial importancia), por lo que es necesario comprar además un osciloscopio externo al equipo. Esto lo hace totalmente independiente de las funciones que tiene cada uno por separado y utilizando todos los equipos a la vez ocupan mucho espacio.

Protección:

El PU - 2200 cuenta con protección básica pero no para todos los componentes. La fuente esta protegida con fusibles (los cuales protegen, pero dejan al instrumento fuera de uso hasta colocar los nuevos fusibles), además, cuenta con un medio visual (LEDS) para indicar un cortocircuito. El frecuencímetro y el generador de funciones, cuentan con un arreglo de diodos y resistencias para proteger las salidas y las entradas; pero en estos casos no existe un medio automático que sea capaz de no sólo detectar la falla, sino que, realice una operación de apertura en el circuito para eliminarla.

CAPITULO 2

DESCRIPCIÓN GENERAL DEL PROYECTO.

Aspectos generales del proyecto.

La mayoría de equipos de medición con que se cuenta en nuestro país, tienen como limitante la protección de ellos mismos y muchas veces; se encuentran en módulos separados, ocupan mucho espacio, provocan desorden (por la cantidad de cables que provienen de distintos equipos y no de uno solo), en algunos casos en lugar de facilitar el aprendizaje del alumno lo complica, ya que son complejos o desactualizados y en el peor de los casos hay instituciones que no cuentan con un equipo para que los alumnos desarrollen sus prácticas, obligando a los alumnos a que ellos mismos se las arreglen para llevar los suyos tales como: fuente de voltaje, multímetro, etc.

Ya sea que los alumnos lleven sus propios instrumentos de medición o la institución se los proporcione; por no estar adecuadamente protegidos, están expuestos a fallar en un momento dado.

Hay equipos de medición que son muy exactos, precisos, fáciles de instalar, que poseen una buena interfaz para el usuario y con un verdadero sistema de protección. Pero su precio es muy excesivo y no está dentro del presupuesto para algunas instituciones y en cuyo caso se logre adquirir algunos equipos de estos, no logran cubrir a su totalidad las expectativas de aprendizaje de los usuarios.

La idea surge a partir de las carencias mencionadas anteriormente y de la necesidad que existe por suplirlas, es por ello que hemos diseñado y construido un proyecto que combinándolo con los sistemas computacionales y electrónicos, dé como resultado un equipo de medición auxiliar para los laboratorios de electrónica.

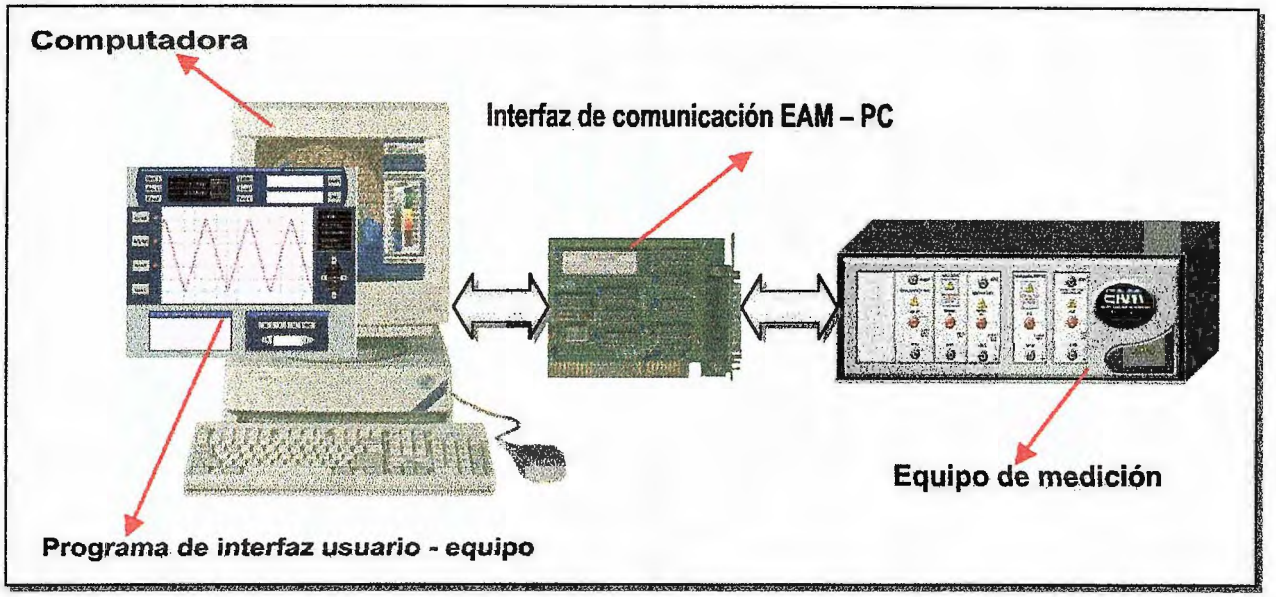
El presente proyecto se basa en un ***Equipo auxiliar de medición controlado por computadora para laboratorios de electrónica***. El cual tiene a disposición del usuario, todos los componentes necesarios para poder desarrollar las prácticas.

El equipo puede ser utilizado en su totalidad mediante una computadora, en donde se encontrará disponible un sistema operativo para que el usuario pueda manipular todos los componentes del EAM (equipo auxiliar de medición). La comunicación se realiza mediante una interfaz entre PC y EAM, en el EAM se encuentran disponibles todos los instrumentos como: multímetro, generador de funciones, frecuencímetro, fuente de voltaje, osciloscopio y sus respectivas líneas para trabajar con un determinado circuito. Como se mencionó anteriormente, el sistema está conformado por tres etapas: computadora, interfaz de comunicación y el equipo de medición; éstas se describen a continuación:

Funcionamiento de las etapas del proyecto.

A continuación, se presenta un esquema gráfico que representa las etapas generales más importantes del proyecto. Como puede observarse el esquema esta conformado por tres etapas: computadora (con su respectivo sistema de operación), interfaz de comunicación y el aparato de medición (EAM).

Esquema gráfico del EAM controlado por computadora

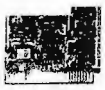


Computadora y sistema operativo:

Todos los datos son procesados por la PC, en la que se dispone de un programa operativo que le sirve al usuario, para controlar todo lo que sucede en el equipo como por ejemplo: prefijar todos los rangos, visualizar en pantalla los cambios que se den en cada instrumento, analizar la información que viene del EAM, etc. Los datos prefijados se podrán modificar sólo desde la PC, la interfaz gráfica (IG) cuenta con todas las opciones esenciales para que sea el operador quien controle al equipo. Los requerimientos mínimos que estimamos convenientes para la PC a utilizar son los siguientes: un microprocesador 80386 (con una F ≥ 33MHZ) o superior y bajo un ambiente DOS, no requiere de un puerto extra, ya que cuenta con su propio puerto mediante la incorporación de una tarjeta de interfaz para la comunicación (colocada en el bus ISA de la tableta madre de la PC).

Se asume que la computadora es externa al EAM, por lo que no estará expuesta a sufrir daños por algún desperfecto en el equipo (lo que seria una probabilidad muy remota por que el EAM está protegido).

El sistema operativo está construido bajo un lenguaje de programación de alto nivel (de tercera generación): **Turbo C++**.



Interfaz: Se justifica la existencia de una interfaz de comunicación por la necesidad de un entendimiento entre la PC y el EAM, para poder lograr un funcionamiento óptimo. En este tipo de comunicación; ambos dispositivos son capaces de

transmitir información en ambos sentidos pero no al mismo tiempo, aunque esto no afecta el rendimiento del proyecto, ya que se puede apreciar en forma simultánea diversas características.

La interfaz está construida en forma de tarjeta para que pueda colocarse en una ranura del bus ISA; de esta manera, la instalación será mucho más práctica y quedaran libres los otros puertos de la PC.

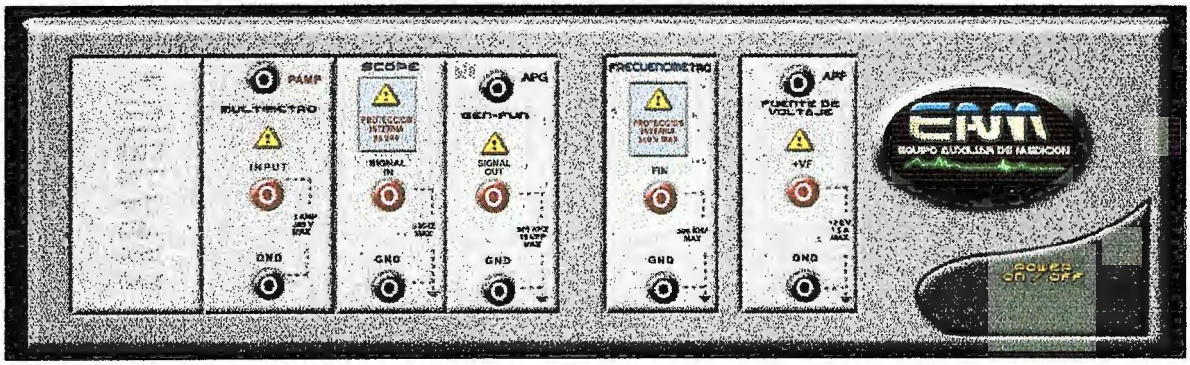
Esquema visual del EAM y descripción de cada modulo.

EAM: Esta es una de las etapas más importantes del proyecto, porque es aquí donde está alojada toda la circuitería encargada de convertir una señal analógica que proviene de un circuito externo a una señal digital, para ser enviada a la PC y finalmente procesada por el programa operativo, además, toda aquella información que el usuario introduce es procesada, al mismo tiempo es enviada hacia el equipo de medición como una señal digital, donde éste se encarga de convertir esta señal en una señal con características análogas.

La protección estará ubicada en esta etapa justamente en la salida y entrada de cada instrumento, ya que es precisamente aquí donde existe peligro de fallos por mal uso.

Se cuenta con un equipo externo a la computadora (EAM) donde se tenga acceso físico a las señales del generador de funciones, fuente de voltaje y las entradas para el multímetro, frecuencímetro y osciloscopio. Se pretende que éste sea lo más compacto posible para no ocupar mucho espacio e incomodar al usuario.

Panel frontal de trabajo:



A continuación se detalla la función que realiza cada instrumento del EAM, tomando en cuenta que el usuario podrá operarlos únicamente desde la interfaz gráfica alojada en la PC. Para poder apreciar los rangos y límites de medición para cada instrumento ver tabla de la Pag. 32.

1- FRECUENCIMETRO:

Este instrumento mide la frecuencia de cualquier tipo de señal; se pretende que dicha frecuencia se pueda medir desde una punta de prueba en el EAM, enviando esta información y presentándola en la PC.

2- FUENTE DE VOLTAJE:

Se puede obtener un valor determinado de tensión que el usuario necesite; por ser una fuente de tensión variable, este voltaje se podrá modificar desde el sistema operativo alojado en la PC.

3- GENERADOR DE FUNCIONES:

Se puede obtener una señal determinada que conste de una amplitud, período y tipo de señal (triangular, cuadrada y senoidal). La obtención de una función requerida se podrá lograr programando las características que se requieran desde la PC, para que posteriormente sean enviadas y manipuladas por la circuitería de dicho instrumento. La frecuencia y amplitud de la señal son auto rango.

4- OSCILOSCOPIO:

Se pueden encontrar en el mercado osciloscopios controlados por computadora en forma individual (a un precio no muy elevado, sí lo comparamos con un osciloscopio convencional), pero no incorporado a un equipo completo con múltiples funciones adquisitivas por el experimentador.

Con la implementación del osciloscopio se pueden apreciar en pantalla las señales que provengan de un circuito en análisis y manipular todos los controles desde la interface equipo - usuario. Las opciones de visualización y control son de mayor apreciación que las mostradas por un osciloscopio convencional. El osciloscopio funciona en forma definida y solo bastará manipular las opciones de escala (VOLT/DIV y TIME/DIV), así como también, las cotas (C1 - C4) para analizar una señal.

5- MULTIMETRO:

Consta de un óhmetro, amperímetro AC y DC, voltímetro AC y DC. Cada uno de estos instrumentos puede ser seleccionados desde la IG (interfaz gráfica) y sólo se tendrá acceso a uno a la vez (de igual forma que se hace en un multímetro convencional). En cuanto a la protección se pretende que sea de carácter individual para cada instrumento y en algunos casos, similar, adaptado a las necesidades del mismo.

Limite de los rangos para cada instrumento.

INSTRUMENTO	RANGO
FRECUENCIMETRO	F= 1H_Z ~ 100KH_Z V = 500vpp
GENERADOR DE FUNCIONES	F= 8H_Z ~ 500KH_Z V = 15vpp
OSCILOSCOPIO	Fmax= 5MH_Z V = 80vpp 0.5μ seg/ div – 100m seg / div 50mV / div – 10v / div
VOLTÍMETRO DC	V ≅ 0VDC ~ ± 500VDC
VOLTÍMETRO AC	V ≅ 0VAC ~ 350VAC
OHMETRO	R = 0Ω ~ 5MΩ
FUENTE DE VOLTAJE VARIABLE	V ≅ 0VDC - 12VDC: 1.5A
AMPERÍMETRO AC Y DC	I ≅ 500μA ~ 2A

Sistema de protección del EAM

Esta es una mejora que incorporamos a nuestro proyecto; otra razón que lo hace factible, ya que con un sistema de protección adecuado se evitarán daños parciales o permanentes al equipo de medición. La protección será de carácter individual para cada instrumento.
La protección es capaz de prevenir aquellos errores que se cometen más frecuentemente como lo son: un corto circuito en la fuente, un voltaje o corriente que sobrepase los límites de medición del multímetro y del osciloscopio, dispositivo seleccionado no adecuado para la medición que se requiere; cuando el multímetro está en ohmios y se quiere medir voltaje o corriente (este tipo de error es común), entre otros. La protección está vinculada con el EAM, de tal forma que al momento de efectuada la falla en algún módulo, este accione la apertura automática del circuito para protegerlo.
En algunos instrumentos la PC indica que el usuario a cometido un error al manipular el equipo.

- **Protección automática:** Este tipo de protección es muy utilizada en los equipos de medición y está conformada por diodos y resistencias.

- **Protección inteligente:** Es aquella que no solo protege al equipo en tiempo real sino que a la vez proporciona una señal visual y en algunos casos audible al momento de efectuado el cortocircuito. Dentro de la protección inteligente existe otro tipo de protección que es mucho más compleja en la cual el sistema realiza la acción de protección solo si se dan ciertas condiciones o patrones de consideración en el circuito.

Uno de los instrumentos de medición que más está expuesto a sufrir averías es el multímetro y los errores más frecuentes cuando se manipula un multímetro y bajo el cual hemos tratado de proteger este dispositivo son:

A. Se quiere medir un voltaje o corriente AC (alterno) cuando descuidadamente se ha dejado el multímetro para medir dicha variable en DC (directo) y por si fuera poco en una escala inadecuada.

B. El voltaje o corriente a medir, sobrepasa la escala seleccionada, lo cual puede superar las características eléctricas del fusible y que éste se queme; esto provoca daños parciales, por lo que no se puede utilizar estos componentes del multímetro.

C. Se quiere medir una variable eléctrica que no corresponde a la función seleccionada por el multímetro, esto es mucho más peligroso porque podría provocar fallas en el circuito integrado principal del multímetro y lo dejaría fuera de uso.

A continuación se presenta una tabla en donde se describe la protección para cada instrumento.

- **Protección para cada modulo de trabajo.**

EQUIPO	PROTECCION	DESCRIPCION	INDICADOR
Fuente de voltaje	**Protección inteligente**	Se protege al ser detectado un corto circuito o una carga que sobra pase los 1.5Amp.	Indicación visual mediante un Led en el EAM y advertencia proporcionada por la IG.
Frecuencímetro	**Protección automática**	Tiene un arreglo de resistencias y diodos zener que evitan corrientes muy altas al aplicarle un voltaje elevado.	
Generador de funciones	**Protección automática**	Pose una resistencia interna a la salida para lograr una corriente baja, cuando es realizado un corto circuito, evitando, así un daño al operacional de salida.	Voltaje de salida tiende a cero voltios.
Osciloscopio	**Protección automática**	Al aplicar un voltaje alto, un arreglo de resistencias y diodos logran recortar la señal para evitar dañar otros dispositivos en el osciloscopio.	La señal se puede observar recortada a plena escala.
Multímetro	**Este instrumento posee protección automática y protección inteligente**	El amperímetro se protege al detectar corrientes altas(protección inteligente), el óhmetro y el voltímetro poseen un arreglo para evitar voltajes elevados; pudiendo ocasionar daños al instrumento.	Al protegerse el amperímetro se activa el Led de protección del multímetro del EAM y hay una advertencia en la IG, por otro lado el óhmetro y el voltímetro presentan su máximo valor de escala.

Metodología de diseño e implementación.

La estrategia que se implementó para poder llevar a cabo la elaboración de este proyecto, fue la siguiente:

- Los conocimientos adquiridos a lo largo de la carrera sirvieron para obtener las bases para poder elaborar el proyecto en un nivel básico.

- Teniendo los conocimientos básicos, se consultaron libros referente al tema, manuales de integrados, revistas y otros documento para elaborar dicho proyecto.

- Teniendo los recursos o información de donde poder alimentarnos en conocimientos; se analizo esta información para poder obtener un mejor resultado en la implementación de este.

- Pero dada la naturaleza y complejidad de este, fue bastante difícil encontrar alguna información bastante concreta para poder adaptarla al equipo, ya que en los libros, Internet, manuales y revistas presentan proyectos, pero no se adaptan al tipo de control y lo más importante, cómo proteger al equipo de los errores que se comenten al realizar mediciones.

- Primero se diseño la interfaz por ser la encargada de que se puedan entender los instrumentos de medición con la computadora; en el momento del diseño se considero que tenia que ser capaz de cumplir todas las exigencias de los instrumentos y algo muy importante que tuviera la capacidad de que en un futuro se pudieran agregar otros instrumentos sin sufrir ningún cambio todas las tarjetas.

- Dadas las condiciones por las que fue aprobado el proyecto, se tubo que realizar investigaciones de algunos integrados para poder satisfacerlas, pero las características que exigen los integrados no se encuentran en nuestro país, por lo que hubo la necesidad de mandar a traer dichos integrados al extranjero.

- Se empezó a diseñar por etapas el proyecto, primero por la fuente de voltaje, el frecuencímetro, el generador de funciones, el osciloscopio y por ultimo el multímetro; antes de empezar a diseñar el generador de funciones y el osciloscopio sé tubo que hacer pruebas con los integrados que se mandaron a traer al extranjero para familiarizarnos con ellos.

- En algunos casos el diseño planteado lograba satisfacer las necesidades requeridas, pero la complejidad del problema aumentaba cuando sé tenia que tomar en cuenta la protección, razón por la cual sé tubo que rediseñar en algunos casos dicho circuito para lograr protegerlo.

- Dada la magnitud del proyecto, se tuvo que armar la mayoría de instrumentos individuales, ya que ocupaban demasiadas breadboards, por lo que una vez funcionando una etapa, se desarmaba el instrumento y se empezaba a diseñar y armar el siguiente.

- Cada uno de los diagramas se hicieron en el programa de EAGLE al final de cada instrumento, cuando éste ya estaba funcionando en breadboard, para tener el diagrama y las pistas de forma ordenada.

- El software para la fuente de voltaje, generador de funciones y el frecuencímetro se hizo de forma individual y no se realizo en forma gráfica.

- A partir del osciloscopio se empezó hacer el software de forma gráfica y con el objetivo que este programa iba a empezar a absorber los demás programas dada la complejidad de este.

- Se mando hacer el chasís del equipo y se le hicieron todos los ajustes necesarios para colocar las tarjetas que iban funcionando.

- Una vez diseñado y probado todos los circuitos en breadboard, se procedió a hacer las pistas del la interfaz interna, interfaz externa y la fuente de voltaje; se soldaron los elementos y se probaron. Funcionando estas tarjetas se procedió a realizar el software de forma gráfica.

- Luego se hicieron las tarjetas del frecuencímetro, osciloscopio y generador de funciones, se soldaron los elementos y se realizaron ajustes hasta que funcionaron adecuadamente, teniendo las tarjetas funcionando se procedió a la realización del software en forma gráfica y su respectiva calibración.

- La ultima pista que se realizo fue la del multímetro y se hizo el mismo procedimiento.

- Se realizaron ajustes finales.

CAPITULO 3

CONSIDERACIONES PARA EL DISEÑO DEL EAM.

A continuación se explica y detalla todo el proceso de diseño y funcionamiento de cada parte que conforma el EAM.

INTERFAZ INTERNA:

OPERACIÓN ESCRITURA:
Cuando la PC envía información a través del canal de datos (BUS ISA), (vea página 4, capitulo 1), coloca la dirección (IC3, IC1C, IC1B, IC2B) en el decodificador de direcciones, luego manda el pulso de escritura, poniendo la salida de ENABLE a cero "lógico", enviando el dato un tiempo después.

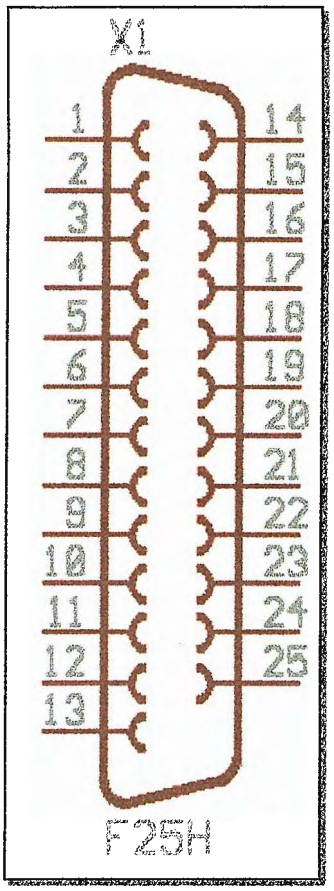
OPERACIÓN LECTURA:
Cuando la PC recibe información ésta coloca la dirección en el decodificador de direcciones, luego manda el pulso de lectura (poniendo la salida ENABLE en cero lógico) y la PC accesa los datos un tiempo despues (vea página 4, capitulo 1).

ENABLE:
Este terminal pasa a "cero" lógico únicamente si se necesita enviar o recibir información entre direcciones F3F0 y la F3FF, las salidas A0 a la A3 del bus de direcciones se utilizan para poder seleccionar específicamente el dispositivo que se encuentra en dicha dirección.

Como ENABLE pasa a cero lógico, habilita el buffer bidireccional de tres estados (74245); las salidas IOW y IOR están normalmente en uno lógico, como necesita enviar datos en este caso IOW pasa a cero lógico, IOR queda en uno lógico, por lo tanto coloca un uno lógico en el pin DIR del 74245 ("este es un buffer bidireccional que al estar habilitado (G=0) y Dir =1 deja pasar los datos de A hacia B y si Dir =0, deja pasar los datos de B hacia A, si no esta habilitado (G=1) éste pasa a alta impedancia"). Por lo que deja pasar los datos en la dirección especificada para que puedan ser escritos en algún registro del EAM.
Si ahora se desea leer un dato de alguna dirección que se encuentre entre F3F0 y F3FF, IOR pasa a cero lógico, por lo que provoca que ENABLE pase a cero lógico, IOW se queda en uno lógico, obligando a que Dir sea "0" logrando así leer el dato que se encuentra en dicha dirección, que puede venir usualmente de un Buffer 74244 del EAM.

La interfaz interna se comunica con la externa por medio de un puerto (DB25), donde se encuentran todas las líneas de datos (D0 - D7), ENABLE y líneas de dirección desde A0 hasta A3, tal como se muestra a continuación:

- **Bus de datos y control para la interfaz de comunicación.**



Conector DB-25

# DE PIN	FUNCION
1	D0
2	D1
3	D2
4	D3
5	D4
6	D5
7	D6
8	D7
9	EN
10	IOR
11	IOW
12	
13	
14	A0
15	A1
16	A2
17	A3
18	A3/
19	
20	
21	
22	
23	
24	
25	GND

Nota: La fuente de 5V de la PC alimenta únicamente a la interfaz interna.

INTERFAZ EXTERNA.

La interfaz externa es la continuación de la interfaz interna se separo, con el fin de no sacar excesivas líneas de la computadora y así poder utilizar un cable de 25 líneas (DB-25).

Esta parte es la encargada de decodificar las direcciones para seleccionar los distintos elementos del EAM.

Se utilizan decodificadores (74LS138) que es un circuito lógico que acepta un conjunto de entradas que representan números binarios y que activan solamente la salida que corresponde a dicho dato de entrada (las salidas están normalmente en uno lógico).

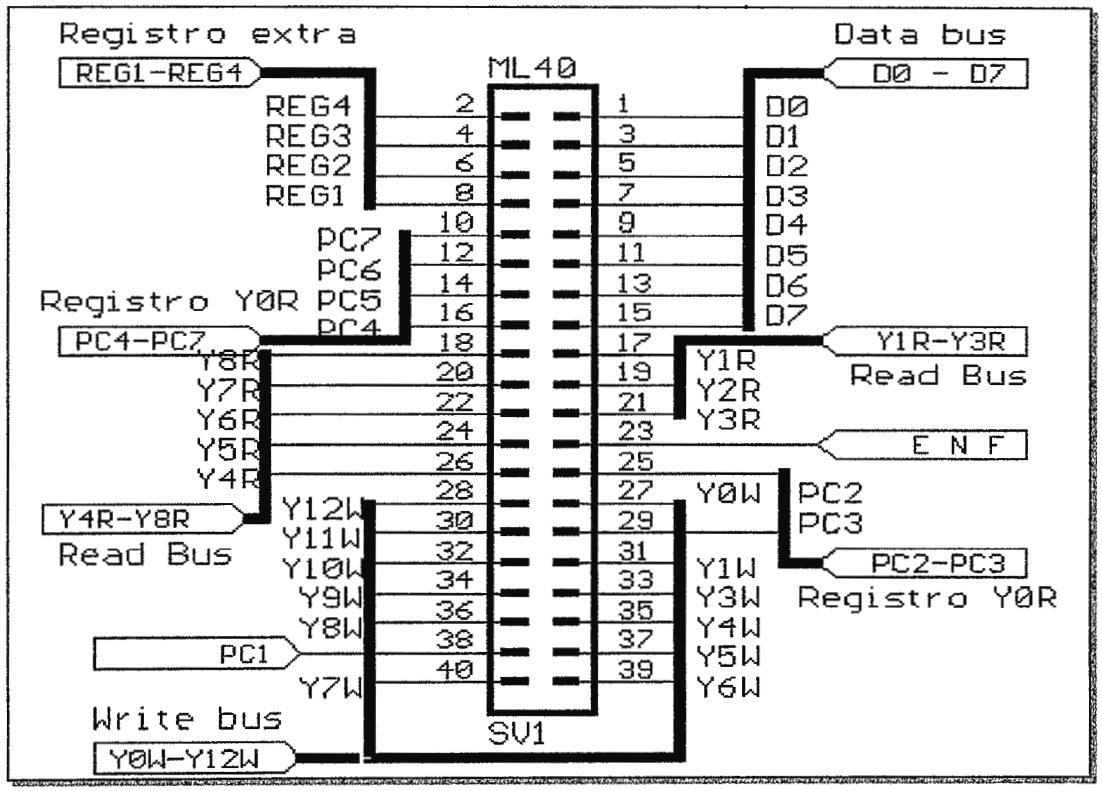
Este integrado (74LS138) tiene tres entradas de habilitación que sirven para activar las salidas del decodificador; una activa en alto (G1) y los otras dos (G2A y G2B) activas en bajo.

Cuándo se necesita enviar datos, el integrado IC1 decodificará las direcciones comprendidas entre la F3F0 y la F3F7. El integrado IC3 decodificara las direcciones comprendidas entré la F3F8 y la F3FF.

Cuando se necesita leer datos el integrado IC2 decodificara las direcciones comprendidas entre la F3F0 y la F3F7. El integrado IC4 decodificará las direcciones comprendidas entre la F3F8 y la F3FF.

El integrado IC5 es seleccionado únicamente cuando se lee el dato que se encuentra en la dirección F3F8.
El integrado IC6 es seleccionado únicamente cuando se envía un dato en la dirección F3FA.
Todas las líneas de selección y datos van a un conector común con las otras tarjetas para mayor detalle ver el conector y la tabla que se presenta a continuación:

- **Esquema de conexiones del bus de datos y control:**



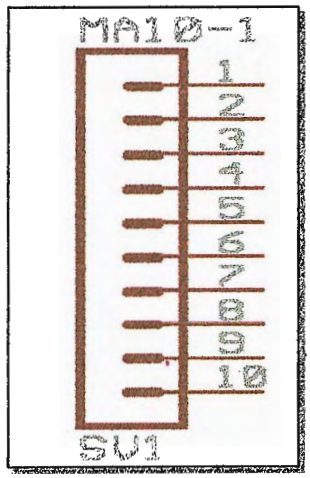
Bus de datos y control del EAM:

Nº DE PIN	NOMBRE	DSCRIPCION	DIR
1	**D0**	DATO 0	
3	**D1**	DATO 1	
5	**D2**	DATO 2	
7	**D3**	DATO 3	
9	**D4**	DATO 4	
11	**D5**	DATO 5	
13	**D6**	DATO 6	
15	**D7**	DATO 7	
17	**Y1R**	Dirección para la parte baja del frecuencímetro (b1)	F3F9
19	**Y2R**	Dirección para la parte media del frecuencímetro (b2)	F3FA
21	**Y3R**	Dirección para la parte alta del frecuencímetro (b3)	F3FB
23	**ENF**	Habilitador del frecuencímetro (activo en alto)	F3FA
25	**PC2**	Indica que el frecuencímetro ya termino de hacer la conversión (cuando es 1 lógico)	F3F8
27	**Y0W**	Sirve para cargar el valor binario equivalente del voltaje que se desea en la fuente	F3F8
29	**PC3**	Indica que el GEN - FUN sé a protegido (cuando es 1 lógico)	F3F8
31	**Y1W**	Sirve para quitar la protección de la fuente	F3F9
33	**Y3W**	Sirve para cargar la parte menos significativa del valor binario de la frecuencia para el generador de funciones (Byte 0)	F3FB
35	**Y4W**	Sirve para cargar la parte más significativa del valor binario de la frecuencia para el Gen - Fun (Byte 1)	F3FC
37	**Y5W**	Sirve para cambiar escalas y tipo de señal para el Gen - Fun	F3FD
39	**Y6W**	Escalas de VOLT/DIV para el SCOPE	F3FE
2	**REG4**		
4	**REG3**		
6	**REG2**		
8	**REG1**		
10	**PC7**		
12	**PC6**		
14	**PC5**		
16	**PC4**	Fin de adquisición de datos	F3F8
18	**Y8R**		F3F0
20	**Y7R**		F3FF
22	**Y6R**		F3FE
24	**Y5R**		F3FD
26	**Y4R**	Lectura de la información almacenada por el adquisidor de datos	F3FC
28	**Y12W**		F3F4
30	**Y11W**	Escalas del voltímetro	F3F3
32	**Y10W**	Escalas del óhmetro	F3F2
34	**Y9W**	Escalas del amperímetro	F3F1
36	**Y8W**	Escalas de TIME/DIV para el SCOPE	F3F0
38	**PC1**	Indica que sé ha protegido la fuente (cuando es 1 lógico)	F3F8
40	**Y7W**	Se utiliza para cargar el valor binario equivalente del vpp en el generador de funciones.	F3FF

- **REGISTRO Y0R (DIRECCION F3F8):**

BIT	NOMBRE	DESCRIPCION
0	PC1	Cuando es uno lógico indica que sé ha protegido la fuente
1	PC2	Si es uno lógico el frecuencímetro ya termino de hacer la conversión
2	PC3	Cuando es uno lógico indica que sé a protegido el generador de funciones
3	PC4	Fin de la adquisición para el SCOPE
4	PC5	Protección del amperímetro(activa con uno lógico)
5	PC6	Indica la polaridad de la variable eléctrica bajo medición
6	PC7	
7	PC8	

- **Bus común para la alimentación de todas las tarjetas:**



PIN	VOLTAJE	CORRIENTE(max)
1	**GND**	
2	**+5V**	2.0A
3	**+8.37V**	0.1A
4	**+12V**	0.5A
5	**+15V**	0.3A
6	**+22V**	1.6A
7	**-15V**	0.3A
8	**-12V**	0.3A
9	**-8.4V**	0.1A
10	**GND**	

Fuente de voltaje.

Cuando se escoge este dispositivo, la PC pedirá al usuario el voltaje que éste necesite (este valor se almacena en la variable voltaje), mandando el valor binario de la ecuación hacia el registro (IC1) por medio de la dirección F3F8, el DAC (U1) convierte este dato digital a un valor análogo (ver diagramas esquemáticos en pag.56).

Dato =(Voltaje / 22.1875e-03) / 2.1856

Donde:

Dato: Es el valor que sé envía al DAC.
Voltaje: Es el Voltaje deseado por el usuario.

El DAC puede lograr un valor de voltaje comprendido entre 0V y 5.6V dependiendo del valor de entrada digital (el Vref de DAC es de 5.6V), este voltaje análogo es amplificado por un AMP – OP en configuración no inversor (U2A) con un arreglo de transistores que amplifican dicho voltaje con una ganancia de 2.1428, para lograr el voltaje de salida deseado (el voltaje digitado por el usuario) este arreglo puede proporcionar una elevada corriente que es proporcionada por el transistor de potencia (2N3055).

La salida de la fuente lleva en serie un contacto normalmente cerrado y unas resistencia de monitoreo (R12, R17, R25, R26 conectadas en paralelo). Un diferenciador de voltaje (U3A) mide la diferencia de potencial que hay en las resistencias de monitoreo (este voltaje depende en gran medida de la corriente que fluye por la carga), que luego es amplificado (U3B) para ser comparado (U3C) con el voltaje del diodo rectificador (D3). Si el voltaje del diodo es mayor que el voltaje medido en las resistencias, el operacional es saturado negativamente, saturando también al transistor de conmutación, por lo que se coloca un "uno" lógico en el pin #1 del IC2A (esta condición permite que el F-F RS este en memoria). En caso contrario que el valor medido en las resistencias después de aplicarle la ganancia sea mayor que el voltaje del diodo rectificador, provocaría que el transistor vaya a corte obligando al pin #1 del IC2A que pase a "cero" lógico (condición de SET del F- F RS).
El objetivo de medir el voltaje en dichas resistencias (Rout de la fuente) es porque al aumentar la corriente de carga aumenta también el voltaje en las resistencias y el sistema de protección se ha diseñado de tal forma que al tener una corriente de carga mayor de 1.5Amp, el voltaje de las resistencias (ya amplificado) sea mayor que el voltaje del diodo rectificador (0.7V).
Cuándo la entrada de seteo del FLIP - FLOP RS (pin#1 del IC2A) pasa a cero lógico, la salida del flip- flop (Q) pasa a "uno" lógico provocando que el relé abra su contacto para que deje la fuente de entregar corriente de carga, además el Led se enciende indicando que la fuente se protegió (por que la salida Q` del FLIP - FLOP RS paso a cero lógico). Por otro lado, el software está observando el estado de la salida (Q) del FLIP - FLOP, cuando ésta pasa a uno lógico el programa entiende que la fuente sé ha protegido y coloca una advertencia en la pantalla, para que el usuario entienda que hay un error.

Se elimina la protección hasta que el usuario presione el botón de la fuente mandando un pulso a través de la dirección F3F8 bit "0" al pin de reset del flip- flop RS (pin #5 del IC2B) para que el relé se cierre.

Cuando el usuario quita la protección tiene que estar seguro que la falla que provocó la protección a sido eliminada por que sino la fuente se volverá a proteger.

El tipo de protección de este instrumento es bastante efectivo, ya que se hace en tiempo real por medio del hardware y la función del software es presentar el mensaje y eliminarlo cuando el usuario ya esté seguro que se elimino la falla.

FRECUENCIMETRO.

El frecuencímetro puede medir la frecuencia de voltajes elevados, ya que los diodos zener D2 y D3 se encargan de recortar el voltaje hasta un punto determinado, con el objetivo de proteger al operacional que compara los niveles de voltaje (U1A) para poder determinar si es "uno" lógico o "cero" lógico, una vez comparado el transistor se encarga de enviar los niveles de voltaje al "contador de pulsos de retardo" (los niveles de voltaje que proporciona dicho comparador son los voltaje de alimentación del operacional +12v y –12V por ese motivo se ocupa el transistor).

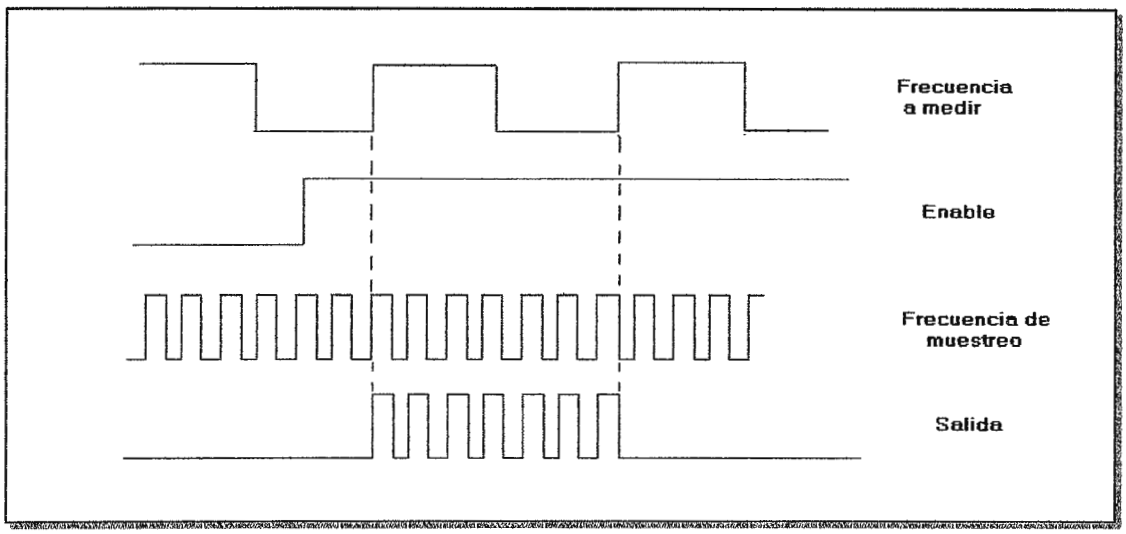
El frecuencímetro necesita un inicio de conversión (ENABLE =1) para poder empezar a medir la frecuencia, que es proporcionado por la dirección F3FA BIT 0. Una vez enable está en "uno" lógico deja al sistema en condiciones para poder medir el ciclo de una señal.

El contador de pulsos de retardo al detectar un flanco positivo (de la señal que se quiere medir), deja pasar la señal de muestreo hacia los contadores binarios (Ic6,Ic7,Ic8) durante un ciclo completo de la señal que se esta midiendo, al detectar otra vez el flanco positivo los FLIP-FLOPS JK detectan esta condición y bloquean con un cero lógico los pulso de muestreo a los contadores; mientras tanto el software está observando el estado de la salida Q del IC5B cuando éste pasa a 1 lógico indica que ya midió el ciclo de una señal (señal que indica fin de conversión a la PC) y que se pueden leer los datos de los contadores para poder determinar la frecuencia medida.

Los buffer 74LS244 (Ic9, Ic10 y Ic11) de tres estados se utilizan para evitar conflictos a la hora de empezar a leer los datos que tienen cada uno de los contadores, los datos se accesan a la PC en tres grupos de 8 bit, una vez leídos los datos el programa envía otro pulso de inicio de conversión (transición negativa para resetear los contadores y los flip- flop J-K) y este proceso se repite periódicamente.

La señal de muestreo (20MHZ) es con el objetivo de no depender de la frecuencia de la computadora porque hay unas mas lentas que otras y puede provocar lecturas erróneas.
A continuación se presenta el diagrama de tiempo del contador de pulsos de retardo:

- **Diagrama de tiempo del contador de pulsos de retardo.**



Registros de control para el frecuencímetro.

- **REGISTRO Y2W (DIRECCION F3FA)**

BIT	NOMBRE	DESCRIPCION
0	ENF	Se utiliza para habilitar el frecuencímetro (CON 1 LOGICO)
1	REG1	Expansión del EAM.
2	REG2	
3	REG3	
4	REG4	

Generador de funciones.

El integrado principal del generador de funciones es el XR2206, ya que es él, el que genera los tipos de señal y la frecuencia de dicha señal.
El XR2206 determina la frecuencia por medio del valor del capacitor que se encuentra entre los pines TC1 y TC2 y del valor de corriente que sale del pin TR1 (este pin tiene 3 voltios de diferencia con respecto al pin GND que en este caso está alimentado a -8.3V).

REGISTRO Y5W (DIRECCION F3FD)

BIT	NOMBRE	DESCRIPCION
0	Trg=0/sen=1	Selección de señal triangular[1] y senoidal
1	Square	Se selecciona la señal cuadrada (esta señal tiene mayor prioridad que las anteriores)
2	C1	Escala 1 (UNO LOGICO)
3	C2	Escala 2 (UNO LOGICO)
4	C3	Escala 3 (UNO LOGICO)
5	LD/DA	Sirve para que realice la conversión el DAC (normalmente en 1 lógico)
6	C5=0/C4=1	Con 1 lógico selecciono la escala 4 y con CERO lógico la escala 5[2]
7	Error gen	Se utiliza para deshabitar la protección del generador de funciones

1- En condiciones iniciales el programa se encarga de seleccionar la señal triangular.

2- El capacitor 5 siempre está cuando sé activa otro capacitor (c1, c2, c3); pero como es demasiado pequeño no afecta.

Cuando el usuario selecciona la frecuencia de una señal se envía un uno lógico a uno de los bit (B6, B4, B3, B2) de la dirección F3FD (estos datos son almacenados en el registro Y5W (IC3) para cerrar el relé y activar el capacitor que corresponde a la frecuencia seleccionada, en realidad el generador de funciones tiene 5 escalas, pero la escala que genera la frecuencia más alta tiene su capacitor conectado al contacto normalmente cerrado del relé K6, entonces, cuando se necesita utilizar la ultima escala, no está seleccionado ninguno de los relés que seleccionan los capacitores de las escalas, cuando se selecciona la primera escala, la segunda o la tercera (K2, K3, K4 respectivamente) este capacitor se le suma, pero es tan pequeño (la capacitancia) que no afecta.
Ninguno de los relés que seleccionan los capacitores son activados al mismo tiempo.
A continuación se presenta una tabla con los valores de los capacitores y las frecuencias obtenidas por estos:

Rangos de frecuencia para cada escala:

Escala	Capacitor	Fs(min) hz	Fs(max) hz
1	C1 = 10.96µF	9	88
2	C2 = 1.187µF	89	818
3	C3 = 125.0nF	819	7.15K
4	C4 = 12.23nF	7.151K	65K
5	C5 = 1.50nF	65.1K	524K

Ya teniendo seleccionado el capacitor, se necesita mandar los datos para lograr el voltaje que determina la frecuencia deseada, este dato se manda a las direcciones:

- Dirección F3FB que contiene la parte menos significativa de la palabra.
- Dirección F3FC que contiene la parte más significativa de la palabra.

Para que sean almacenados en los registros Y3W y Y4W (IC1 e IC2) y puedan ser utilizados por el convertidor (LTC1597).

El integrado LTC1597 es un convertidor de 16 bit (una palabra), con señales de escritura (W/L) (almacena los datos que se encuentran en las entradas digitales al registro de entrada), carga (su función es cargar los datos del registro de entrada al registro del DAC para que se pueda generar el voltaje deseado) y Clear (limpia el registro de entrada y el registro del DAC), el DAC está configurado para generar voltajes de 0V a 3V (3V es el voltaje de referencia que tiene el convertidor).

Una vez teniendo los datos en los registros se manda un pulso (está normalmente en uno lógico) con transición negativa a la dirección F3FD (Y5W), bit 5 para que sea convertido el dato a un valor análogo equivalente; el voltaje del convertidor es restado a un voltaje igual al voltaje de alimentación negativo para cuando el convertidor tenga su máximo valor, fluya una corriente a través del pin TR1 de 1µA, aproximadamente, cuando el DAC tenga su mínimo valor fluya una corriente de 3mA a través del pin TR1 del XR2206.

El dato equivalente al voltaje que se utiliza para lograr la frecuencia, junto con el capacitor Seleccionado, se determina por medio de la ecuación que tiene el software, nótese que en el software el generador de funciones es auto rango.

Ecuación para determinar el voltaje que se necesita para lograr la frecuencia:

$$N° = - [((Fs^* R^*C) - 1) Rc / R + 1] 3 / RESL$$

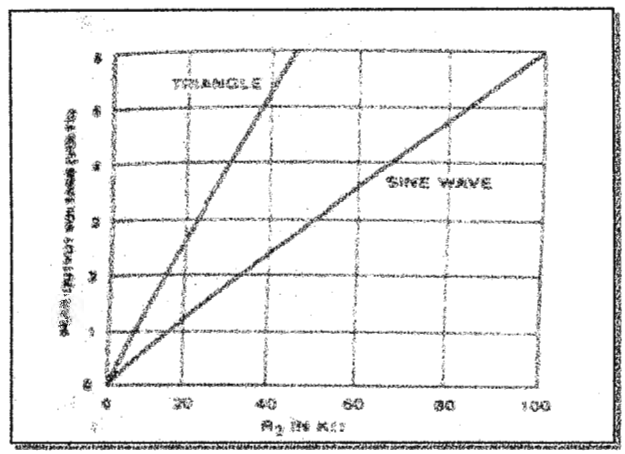
La señal triangular y senoidal está determinada por la cantidad de corriente que pasa entre los pines WAVEA (13) y WAVEA2(14) del XR2206,
cuando se necesita la señal triangular no fluye corriente por estos pines, cuando se necesita una señal senoidal fluye una corriente a través de estos pines.

Cuando el usuario selecciona el tipo de señal, se envía un "uno" al bit "0" de la dirección F3FD, para seleccionar la señal senoidal ó un "cero" lógico, para seleccionar la señal triangular, con el objetivo de cerrar o abrir los contactos del relé (K1) que se encuentra entre los pines WAVEA1 y WAVEA2 (COM2) y para uniformar el voltaje de salida de la señal triangular y senoidal a 1 VPP (COM1).

El potenciómetro R29 se utiliza para calibrar la señal senoidal, haciendo llegar este potenciómetro a una resistencia muy alta, la señal seniodal tiende a verse como una señal triangular y si es muy baja la resistencia, las crestas tienden a verse muy redondas.

La señal triangular o senoidal se pude observar en el pin STO (pin #2), pero el voltaje de salida no es igual en ambas señales, la señal triangular proporciona un voltaje más alto con una resistencia más baja en el pin MO (#3), que la señal senoidal, como se puede apreciar en la siguiente gráfica:

Gráfica de voltaje de salida Vrs resistencia (R29)



Entonces; como se mencionó anteriormente el relé K1 tiene 2 contactos, uno normalmente abierto y el otro normalmente cerrado, el contacto COM2 y el contacto COM1 respectivamente que se activan al mismo tiempo.
Como se puede observar, cuando se selecciona la señal triangular el contacto COM2 está abierto y el contacto COM1 esta normal mente cerrado, por lo que la corriente circula por el potenciómetro R38 y la R25, entonces se ajusta el potenciómetro hasta que la salida (STO) tenga 1Vpp, cuando se selecciona la señal senoidal cambian de posición los contactos, entonces la corriente fluye también por el potenciómetro R27, por lo que se ajusta hasta lograr 1 Vpp en el pin de salida.
Variando el potenciómetro R28 se agrega o se elimina un offset que desplaza la señal, en nuestro caso el offset es eliminado por necesitarse que el voltaje en la señal positiva sea igual al de señal negativa.
La señal cuadrada se obtiene del pin de sincronismo del integrado que proporciona +8 Voltios para un "1" logico y -8 Voltios para "Cero" logico, por lo que se necesita hacer este voltaje más pequeño, y se logra por medio de unos diodos rectificadores que hacen llegar el voltaje a aproximadamente 1.2VPP.
Cuando se selecciona la señal cuadrada se manda un "uno" al BIT 1 de la dirección F3FD para que el contacto del relé (K2) cambie de estado y pueda pasar la señal cuadrada hacia el multiplicador análogo, al seleccionarse cualquiera de las otras señales (senoidal y cuadrada) se manda un "cero" lógico para que el contacto del relé pase a su estado normal y pueda pasar cualquiera de estas señales (senoidal o triangular) al multiplicador análogo.
A través del potenciómetro R32 se puede variar el ciclo de trabajo de la señal desde un 1% hasta un 99%, pero en nuestro caso se ajusta a un 50% el ciclo de trabajo.

Proceso para lograr la amplitud de la señal (Vpp):

Cuando el usuario introduce el voltaje de la amplitud de la señal, este valor es introducido a la ecuación para determinar el valor binario que será sacado por el DAC a través de la dirección F3FF (REGISTRO Y7W), este valor binario será convertido a un valor análogo entre los 0 voltios y los 10 voltios.

Ecuación para determinar el valor binario de la amplitud de la señal:

$$N° = (Vpp / 39.21568627e\text{-}03) / \Delta v$$

Donde:

Δv = 1.5 para la señal triangular y senoidal
Δv = 1.875 para la señal cuadrada

El voltaje de salida (XR2206) de la señal esta ajustado para 1 Vpp para la señal senoidal y triangular y 1.2Vpp para la señal cuadrada. Este voltaje es multiplicado por el voltaje de salida del DAC0808 a través de un multiplicador análogo (AD633) donde la salida del voltaje esta determinada por la siguiente ecuación:

$$W = (X1 - X2)(Y1 - Y2) / 10 + Z$$

Donde:

X1= voltaje de la onda

Y1= voltaje del convertidor análogo (DAC0808) (Voltaje equivalente al voltaje deseado)

X2 = Y2 = Z = 0V.

Al analizar la ecuación, el voltaje que entrega el multiplicador análogo es 1Vpp para la señal senoidal y cuadrada. Este valor de voltaje es amplificado (U4D) 15 veces para poder lograr la amplitud de la señal a 15Vpp.
Para la señal cuadrada el voltaje que entrega el XR2206 es de 1.2Vpp, pero se necesita 15Vpp a la salida del amplificador U4D, entonces se le aplica otra ganancia a la ecuación que sirve para determinar el valor binario que se le entregara al DAC y así se logra la amplitud de la señal a 15Vpp.
La salida del amplificador U4D tiene un seguidor de voltaje (U3A) para que este entregue la corriente al circuito que se está alimentando con el generador de funciones.

Protección del generador de funciones.

La salida del seguidor de voltaje (U3A) tiene conectada una resistencia de 510 ohmios que se le suma a la resistencia de salida del operacional, para evitar que el operacional entregue una corriente capaz de producir graves daños al realizar cortos circuitos o poner una carga demasiada alta a la salida del generador de funciones (GEN -FUN).

Osciloscopio.

Condiciones iniciales:

En el momento que se energiza el módulo del osciloscopio los contadores y dispositivos de almacenamiento temporal son reseteados automáticamente por hardware (utilizando un arreglo RC). Mientras que al inicio del programa se colocan condiciones predefinidas para el osciloscopio como lo son: acople DC de la señal, escala de 1V/DIV y 0.5mS/DIV (como puede apreciarse en el programa). El adquisidor de datos comienza el proceso de adquisición con las escalas antes mencionadas y reproduce el primer muestreo basándose en estas condiciones. Es importante que desde el primer momento de encendido, el equipo y la IG comienzan el proceso de presentación gráfica, éste se repetirá de forma periódica y solo sé podrá detener bajo ciertas condiciones predefinidas.

Etapa de entrada:

La entrada del osciloscopio está conformada por el acople (A) de señal, el cual se controla por medio del registro Y6W, cuando está en A DC la PC a enviado un 0 logico al bit 4 de dicho registro, esto permite que la señal entre tal y como viene al circuito. Cuando está en A AC la PC a enviado un 1 lógico al bit 4 para energizar al relé y que éste conmute para A AC, en este punto se utiliza un capacitor para bloquear la componente DC de la señal y permitir que solo deje pasar la señal cámbiate de la misma. La etapa de entrada está conectada a una red de protección a través de un divisor de voltaje en combinación con diodos zener de 10V, la idea es atenuar la señal hasta un voltaje de 14vpp (con una ganancia de: 7Vp / 40Vp = △1 = 0.175) y que al introducir voltajes mayores a los 80Vpp los diodos zener se vean forzados a romper su barrera de agotamiento y recorten la señal (esta condición se puede apreciar al introducir una señal que supere los 80Vpp y es totalmente normal en un osciloscopio). Luego se aisla al divisor por medio de un separador de ganancia unitaria y se le aplica otra ganancia a través de un amplificador inversor:

△2 = 5Vp / 7vp = 0.71428

Mientras que la ganancia global del circuito a la salida de U2B se obtiene:

△3 = △1 * △2 = 0.175 * 0.71428 = 0.125 (10VPP/80VPP = 0.125)

Esta ganancia △3 es crucial para determinar la resistencia de escala VOLT/DIV (Ver la tabla: "Escalas de VOLT / DIV" en pag.72).

Escalas de VOLT/DIV:

Para controlar las escalas de VOLT/DIV se utiliza el registro Y6W (S0 - S2) con el cual se puede seleccionar un canal del multiplexor análogo DG408DJ. La PC envía códigos binarios hacia dicho registro, para que sean enganchados y colocados a las entradas de selección del M/D, esto permite cambiar el valor de la resistencia de retroalimentacion del OP – AMP (U2C) que a su vez proporcionara una atenuación adecuada de la señal.

Registro Y6W dirección F3FE (escalas VOLT/DIV del osciloscopio):

Q2	Q1	Q0	V/div	Vpp (max)	Hex
0	0	0	**10**	80	00
0	0	1	**5**	40	01
0	1	0	**2**	16	02
0	1	1	**1**	8	03
1	0	0	**0.5**	4	04
1	0	1	**0.2**	1.6	05
1	1	0	**0.1**	0.8	06
1	1	1	**50m**	0.4	07

Acople de señal (Q3)

Q3	Acople
0	DC
1	AC

En cada canal se encuentran las resistencias ajustables. La función especifica de este arreglo es para lograr que en cada escala seleccionada se logre un VD (salida del U2c) máximo de 2VPP (a plena escala), lo que provoca una ganancia global que dependerá de la escala VOLT/DIV:

1VP / VPE (max) = △T

Dada una señal de entrada, el bloque de "*reducción y acople de señal*" se encarga de reducir los niveles de tensión de dichas señales y que al mismo tiempo permite la

selección del acoplamiento de señal AC o DC, ofreciendo una alta impedancia de entrada a las señales bajo medición, con el propósito de no modificar sus características.
La salida de U2C se conecta a un sumador no inversor (U2D) que realiza la siguiente suma:

Ain+ = Vref + VD ⇨ Donde: Vref = 2.5

El voltaje de referencia se conecta al sumador, al terminal AIN- y sirve también para lograr la referencia del LTC1406. Si el voltaje de entrada es de 40Vpp y nos encontramos en la escala de 5V / DIV esto ocasiona que en el semiciclo (+) se tenga un voltaje de 1VP(max) en VD, por lo que:

Ain+ = 2.5 + 1V = 3.5V

En el semiciclo (-) se tendrá un VD máximo de = - 1v, esto da el siguiente resultado:

Ain+ = 2.5V + (-1V) = 1.5V

El resultado anterior refleja que la señal en Ain+ oscila en un intervalo de 1.5 - 3.5V (00 - FF para el LTC1406), la misma condición se cumple para cada escala de VOLT/DIV. El ADC tiene como tarea convertir las señales previamente atenuadas por las etapas anteriores, en códigos digitales que puedan ser almacenados en la memoria RAM para un determinado muestreo de señal. La PC sólo puede seleccionar una escala a la ves y no existe la posibilidad de seleccionar más de una escala (tal y como ocurre en un osciloscopio convencional).

Escalas de TIME/DIV:

Las escalas de TIME/DIV son logradas a partir de una frecuencia base que proviene de un oscilador cuya frecuencia es de 20MZ, esta frecuencia pasa a unos FLIP - FLOPs JK (74S76) y contadores programables (74LS193) para lograr divisores MOD 2 y MOD 5 respectivamente, los cuales se utilizan como la FM (frecuencia de muestreo para cada escala de TIME/DIV). La razón de utilizar estos divisores es la de lograr escalas de TIME/DIV comerciales, fácil de usar por el usuario y adaptables al cristal y ADC que se utilizan. El control de las escalas se realiza desde la IG donde el usuario puede seleccionar la escala deseada, luego la PC envía a través del registro Y8W (F3F0) un BYTE de control cuya función se detalla en la siguiente tabla:

Registro Y8W dirección F3F0 (escala TIME/DIV del osciloscopio):

Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	HEX	ADC	ADD	T/DIV
0	1	0	1	X	0	0	0	58	20M	20M	***0.5µs***
0	1	0	1	X	0	0	1	59	10M	10M	***1µs***
0	1	0	1	X	0	1	0	5A	2M	2M	***5µs***
0	1	0	1	X	0	1	1	5B	1M	1M	***10µs***
0	1	0	1	X	1	0	0	5C	200K	200K	***50µs***
0	1	0	1	X	1	0	1	5D	100K	100K	***0.1ms***
0	1	0	1	X	1	1	0	5E	20K	20K	***0.5ms***
0	0	0	0	X	0	0	1	09	10M	10K	***1 ms***
0	0	0	1	X	0	1	0	1A	2M	2K	***5 ms***
0	0	1	0	X	0	1	1	2B	1M	1K	***10 ms***
0	0	1	1	X	1	0	0	3C	200K	200	***50 ms***
0	1	0	0	X	1	0	1	4D	100K	100	***100 ms***
1	1	1	0	_---_	X	X	X	E7	Lectura RAM		

Adquisidor de datos del osciloscopio:

Para lograr la adquisición de datos se cuenta con un ADC de 8 bit, (LTC1406 vea especificaciones técnicas en los anexos) capaz de realizar conversiones a muy altas velocidades (20MHZ) en combinación con una memoria RAM, cuyo tiempo de acceso es de 15ns.

La velocidad de la adquisición de datos depende de la frecuencia de muestreo la cual es seleccionada por la PC a través de multiplexores (74HC151); sin embargo, nótese en la tabla anterior que hasta llegar a una FM = 20KHZ la frecuencia del ADC y del contador binario MOD 160 (74LS393 que direcciona las localidades de la memoria RAM) son iguales:

FM ADC = FM ADD (Tiempo de adquisición real).

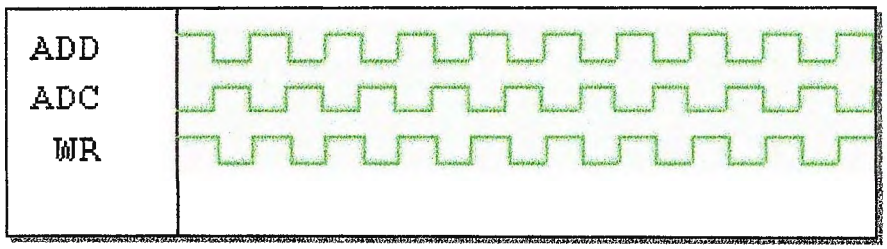
Luego se combinan diferentes frecuencias de muestreo para el ADC y el contador (Tiempo de adquisición equivalente). En realidad las dos formas de adquisición son en tiempo real, la diferencia es que el ADC realiza las conversiones a una velocidad 1000 veces mayor que la escritura en la memoria RAM. La razón principal de realizar esta metodología es porque el LTC1406 no puede responder a frecuencias menores de 10KHZ y se trató de no llegar a los limites del integrado, por ésto el ADC realiza las conversiones a velocidades altas, pero son almacenadas en la memoria a una velocidad 1000 veces menor. Para que exista una sincronía se utiliza en combinación los dos multiplexores (IC11 e IC12), un arreglo de compuertas para conmutación y un FLIP- FLOP (IC15A) con el propósito de lograr la señal de escritura de la memoria RAM. La de escritura proviene del terminal W (IC14C). Las compuertas realizan la función de conmutar las señales que vienen de los multiplexores, esto es necesario porque en el caso de los divisores de frecuencia MOD2 poseen una entrada de reloj activada con transición negativa; mientras que los divisores MOD 5 poseen una entrada de reloj activada por transición positiva. El control de la conmutación de las compuertas se realiza por medio del bit de control Q4 del registro Y8W dicha función se realiza de manera automática. El bit Q7 del mismo registro tiene la función de permitir la lectura y escritura de la RAM.

El proceso de adquisición inicia de manera automática, la PC envía datos que corresponden a las escalas de T/DIV y V/DIV junto con las condiciones de control de escritura (Q7 = 0). El ADC, comienza a realizar las conversiones que provienen de la etapa de VOLT/DIV. Cada muestra efectuada por el convertidor es almacenada en una localidad de memoria. Los datos del ADC son llevados hasta la memoria RAM por medio de una sincronización de un buffer de tres estados 74244 (IC16). La RAM es direccionada en forma secuencial a través de una sincronización entre una base de tiempo (que corresponde a la escala seleccionada de TIME/DIV), un contador binario MOD160 y el circuito de control de W/R (escritura / lectura). Después de terminado el muestreo de la señal, la memoria RAM habrá almacenado 160 BYTES, el contador 74LS393 envía un pulso al temporizador monoestable (74123), el Q` del mismo resetea al contador y permite que un FLIP - FLPP tipo T (arreglo con el 7474 IC15B) bascule (Q` = 0). Esta condición bloquea con un 0 lógico el oscilador principal que genera la frecuencia base, la PC realiza por medio del programa una operación AND a PC4 esperando que ocurra esta condición (PC4 = 0), para iniciar el proceso de lectura de la adquisición almacenada en la RAM.

El inicio de adquisición está determinado por la PC mientras que el fin de adquisición dependerá estrictamente de la FM.

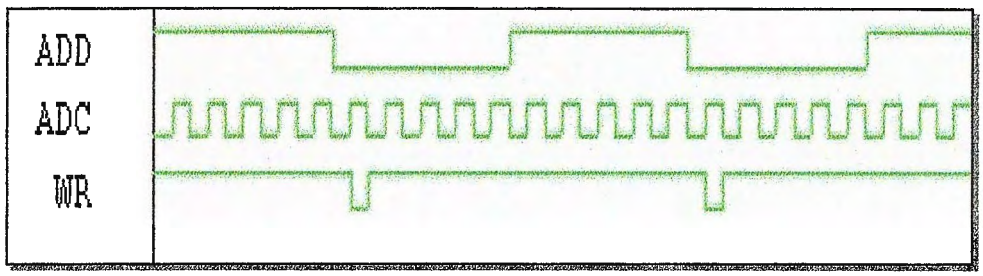
A continuación se presenta un diagrama de tiempo que refleja características de escritura en la memoria RAM y la sincronización que existe con la base de tiempo real y equiváníete:

Diagrama de adquisición en tiempo real:



Como puede observarse en el diagrama de tiempo de adquisición real el LTC1406 comienza la conversión cuando la señal de reloj ADC va de bajada, esto permite que la señal que llega a los contadores binarios (ADD) baya de subida para incrementar la dirección de la RAM en la cual se almacena el dato que sé esta convirtiendo en el ADC. Cuando la señal ADC va de subida el LTC1406 ya tiene disponibles los datos, los cuales son almacenados por medio de la señal de escritura (0 lógico).

Diagrama de adquisición en tiempo equivalente:



En el caso del tiempo de adquisición equivalente podrá notarse que el LTC1406 realiza N conversiones (en total son 1000), de las cuales sólo se almacena una en la RAM. La señal ADD (que se utiliza para incrementar las direcciones de la RAM) y ADC poseen la misma fase por lo que cuando el convertidor a completado las 1000 conversiones la señal ADC y ADD van de bajada, luego la transición positiva de ADC permite que los datos del convertidor estén disponibles y son almacenados en la memoria RAM a través de la señal de escritura W (0 lógico). Se puede observar que el proceso es similar al tiempo real, primero se realiza la conversión, luego se almacena en RAM y se incrementa el contador. El proceso se repite en forma periódica hasta completar los 160BYTES.

Proceso de lectura para el adquisidor de datos.

El proceso de lectura de la RAM es muy sencillo y se logra en forma directa por la PC. La PC envía el dato "E7" al registro Y8W, Q7 coloca el sistema en condición de lectura mientras que las líneas Q4 - Q6 seleccionan la entrada D6 (que proviene del bit de control Q3) del multiplexor (IC12). La PC envía 160 pulsos a este terminal para incrementar la localidad de dirección de la memoria RAM a través del contador (tanto el software como el hardware poseen el contador modulo 160), luego el contador vuelve a cero y el F-F tipo T (IC15B) bascula nuevamente, esto habilita el oscilador de frecuencia base y deja listo al sistema de adquisición para un nuevo muestreo. Para la lectura de los datos de la RAM se utiliza un buffer 74244 (IC17), mientras que el otro buffer (IC16) se encuentra en alta impedancia. La PC almacena los datos en una matriz del tipo arreglo para que finalmente sean interpretados por el programa y efectuar el proceso de presentación gráfica de la señal adquirida en la IG. Luego la PC envía nuevamente los bits de control y escala e

inicia nuevamente la adquisición de los datos, este proceso se repetirá de forma periódica.
La razón de utilizar dispositivos de almacenamiento temporal externos a la PC se debe a que, es necesario obtener la información de la señal bajo medición en tiempo real y que no dependa de la base de tiempo de una computadora. Debido a que no todas las PC's cuentan con la misma frecuencia de trabajo y necesitamos una frecuencia constante para la obtención efectiva de las señales en el dominio del tiempo.

Funcionamiento del multímetro.

Al seleccionar el multímetro la PC solicita al usuario que seleccione el dispositivo que desea utilizar: Óhmetro, voltímetro y amperímetro AC/DC. La PC envía los bytes de dato y de control necesarios para hacer operar la función y escala escogida. El funcionamiento del multímetro es completamente similar al de un multímetro convencional (sólo permite la selección de un instrumento).

Condiciones generales:

Cuando el multímetro se energiza por primera ves la IG manda un pulso (_|¯|_) en PT (protección) para reponer el FLIP - FLOP RS como condición inicial, esto permite que el sistema quede listo para detectar un error en el amperímetro, al suceder dicha condición inicial el Led PAMP (protección del amperímetro) se apaga.
Al inicio del programa se colocan de manera automática las escalas del multímetro; además de la protección que cada dispositivo tiene por separado existe una protección auxiliar la cual se logra colocando un fusible a la entrada del multímetro.
Al referirnos a la etapa de entrada del multímetro se observa que: el mismo relé utilizado para la protección del amperímetro corresponde al relé de selección para el voltímetro; la razón por la cual se coloca el contacto normalmente cerrado (NC) en el voltímetro, es porque el voltímetro a sido diseñado para soportar voltajes altos arriba de 500 voltios.

Voltímetro:

Cuando el usuario selecciona el voltímetro, la PC envía al registro Y9W (F3F1) un pulso a través del bit MSB que es Q7 (PT), además se manda un cero lógico mediante el terminal VOL (bit Q6), con el objetivo de desenergizar la bobina del relé L, para que uno de sus contactos permanezca en la posición normalmente cerrada. El voltaje pasará directamente por una red de protección utilizando un divisor de voltaje (cuya ganancia es de 0.02) y un par de diodos zener que se encargan de forzar el voltaje a un nivel de 15.7V, cuando la entrada excede los 750V (realmente la escala máxima del voltímetro es de 500V); sin embargo, se utilizan diodos zener de 15V para permitir que el voltaje de ruptura de estos no afecte la proporcionalidad del voltaje que se mide. El divisor de voltaje es conectado a un amplificador de ganancia unitaria para aislar la carga que produce la Ri del amplificar no inversor (U5D). Las escalas son controladas a través de relés utilizando el registro Y11W, el objetivo de cada resistencia de escala es lograr un voltaje máximo de salida en U5D de 2V a plena escala. La ganancia para cada una de las escalas se muestran a continuación:

Tabla de AV de entrada para VOLTIMETRO:

N° ESCALA	ESCALA	AV = 2V / ESCALA	RE
1	**500mV**	4	100KΩ
2	**5V**	0.4	10KΩ
3	**50V**	40m	1KΩ
4	**500V**	4m	100Ω

La salida de U5D esta conectada a una resistencia de 100KΩ la que es utilizada para permitir que exista un voltaje de 0V cuando el relé L está momentáneamente en el aire, de este mismo punto se obtiene el signo de la señal, el cual es mandado a la PC a través de PC6. Si pc6 = 1, el signo del voltaje de entrada es positivo, si Pc6 = 0, el voltaje de entrada es negativo, para obtener dicho signo se utiliza un detector de cruce por cero, en combinación de un transistor el cual brinda una señal TTL. La salida de U5D está conectada a un rectificador de precisión de onda completa para obtener el valor absoluto del voltaje de entrada (se hace con el objetivo de aumentar la resolución del multímetro ya que el signo vendría a ser un bit agregado de los 8 bit que posee el ADC), la salida del rectificador de precisión se conecta a un diferenciador el que se encarga de efectuar la diferencia que existe entre el voltaje de referencia del LTC 1406 y el valor absoluto que proviene del rectificador de precisión; si a la salida de dicho rectificador (unidad 1C y 1D), tiene 2V esto indica que el instrumento seleccionado se encuentra en plena escala, la salida del diferenciador tendrá un voltaje de: 1.5V + 2V = 3.5V esto provoca que el ADC tenga a su salida el dato máximo FF (el ADC proporciona dicho valor cuando la diferencia entre sus dos entradas AIN+ - AIN- = 3.5V - 2.5V = +1V).
Cuando la salida del rectificador tiene 0V (esto indica que el voltaje de entrada es 0V o es un voltaje demasiado pequeño para la escala seleccionada) a la salida del diferenciador se obtendrá lo siguiente: 1.5V - 0V= 1.5V esta condición causa un código mínimo 00 en el ADC (esto indica que el ADC tendrá el siguiente voltaje diferencial 1.5V - 2.5V = -1V), la función que realiza el diodo a la salida del diferenciador es la de impedir que se filtren voltajes negativos ya que estos voltajes pueden producir daños al ADC, la unidad 3A se encuentra en una configuración de seguidor de voltaje, este dispositivo se a polarizado a +5 y GND y sirve también como protección para el ADC ya que este no puede soportar a su entrada voltajes mayores a 5V y como el rango efectivo es de 1.5V a 3.5V no existe problema alguno.
La señal del reloj del ADC proviene de un divisor de frecuencia de 1.78MHZ, esta señal se conecta también con la entrada del reloj del IC1 (74374) el cual es habilitado a través del terminal OUTPUT ENABLE, por medio de Y5R. La computadora lee cada cierto tiempo y en forma periódica el dato que a sido enganchado por IC1, este dato es directamente proporcional al voltaje que se está midiendo en combinación con la escala seleccionada.
Cuando el oscilador está en bajada (transición negativa) el ADC comienza la conversión, cuando la señal de reloj va de subida (transición positiva) el ADC ya tiene los datos a la salida mientras que esta transición permite que el registro pueda enganchar los datos del ADC para que finalmente sean leídos y que la PC inicie el proceso de presentación.
Cuando el voltímetro esta midiendo voltaje AC el hardware no sufre cambios y la única diferencia es que el programa revisa el signo varias veces y si este esta cambiando periódicamente, determina que se trata de una señal alterna. El funcionamiento del circuito no cambia, pero se toma en cuenta un retardo de tiempo por software para leer 50 datos que vienen del ADC, el programa encontrara cual ha sido el mayor de estos datos y de esta manera calculara el voltaje RMS y el voltaje máximo de la señal de entrada.

Amperímetro:

El funcionamiento del amperímetro es similar al del voltímetro, salvo por la etapa del *"convertidor de corriente a voltaje"*,

Cuando el usuario selecciona el amperímetro la PC envía a través del registro Y9W un uno lógico en el terminal VOL, además envía al registro Y10W (F3 F2), un cero lógico en el terminal AMPO, esto con el objetivo de colocar al FILP - FLOP RS en condición de "reset" para que quede listo para actuar "SET" en cuyo caso ocurra un error en la medición de la corriente. Las escalas del amperímetro se controlan mediante el registro Y9W (F3F1). Los relés que se utilizan para las escalas del amperímetro poseen dos polos y dos contactos, uno de los contactos sirve para seleccionar la resistencia de escala (por donde fluye una mayor corriente) mientras que el otro contacto sirve para asignarle una ganancia a dicha escala a través de un amplificador inversor.

La PC coloca al relé F en su posición normalmente cerrada mientras que el relé L en su posición normalmente abierta, esto permite la selección del amperímetro.

El amperímetro se conecta en serie en el punto donde se desea realizar la medición, cada resistencia de escala se queda con 0.1V en plena escala, dicho valor pasa a un divisor de voltaje que atenúa la tensión con una ganancia de 166m. Este voltaje es aislado a través de un seguidor (U1A) y se conecta a un amplificador inversor (U1B) el cual proporciona un voltaje de 2V a plena escala, este voltaje se puede expresar junto con la corriente que se mide como una ganancia en AMP/VOLT, tal y como se presenta en la siguiente tabla:

Tabla de ganancia para el Amperímetro:

N° ESCALA	ESCALA	ΔV = IEs / 2V	RE= 0.1V/IEs
1	**500μA**	250μA / V	200Ω
2	**5mA**	2.5mA / V	20Ω
3	**50mA**	25mA / V	2Ω
4	**500mA**	250mA / V	0.2Ω
5	**2A**	1A / V	50mΩ

En el amperímetro al igual que el voltímetro se repite la misma metodología de funcionamiento, lo único que cambia es, la protección, en el caso del amperímetro el diferenciador se conecta directamente a un comparador de nivel. El voltaje en el ADC a plena escala es de 3.5V (FF) pero, si se superan los 3.8V el ADC siempre estará expresando su máximo código; sin embargo este nivel de voltaje provoca que el comparador se sature con +V y en consecuencia el T2 se satura hasta lograr un VCE =0 dicho cero pasa a ser sumado con el cero de AMPO esto provoca un 1 lógico en la entrada SET del FLIP - FLPP RS → Q = PC5 = 1 mientras que Q` = 0, esta condición enciende el LED PAMP y coloca en estado de corte a T3 esto provoca que el relé L pase a la posición VOLT (el amperímetro a sido protegido en tiempo real) la PC revisa cada cierto tiempo el estado de PC5 y si este es uno lógico el programa coloca un mensaje "ERROR MULTIMETRO". Cuando el usuario se percata del error que a cometido y habilita nuevamente el multímetro la PC envía nuevamente las condiciones iniciales y resetea al FLIP - FLOP RS enviado un pulso por el terminal PT; esto repone al RS y deja al sistema listo para seguir midiendo la variable corriente en la escala que se encontraba en el momento del error.

El proceso para medir una corriente alterna es el mismo que el descrito para el voltímetro el programa proporciona el valor máximo y RMS de la I que se mide, es importante mencionar que la protección obedece a la corriente máxima de la señal que se mide.

Óhmetro:

Desde el momento de selección de esta función, la PC enviará al EAM datos de control y selección para permitir que conmute a OHM). Al igual que en el voltímetro se deshabilita la protección (esto se debe a que el óhmetro posee su propia protección). Para lograr la selección de R la PC coloca un 1 en VOL por medio del registro Y9W y un 1 al terminal F para activar el relé "F" y que este conmute junto con el relé "L" a la posición OHM. La escala se controla a trabes del registro Y10W de G - K (5MΩ -500Ω).
La estrategia utilizada con el óhmetro fue la de colocar un divisor de voltaje conformado por +5V, la R de escala y la R de prueba, el objetivo es medir el voltaje del divisor para despejar de la siguiente ecuación: El valor de la R de prueba.

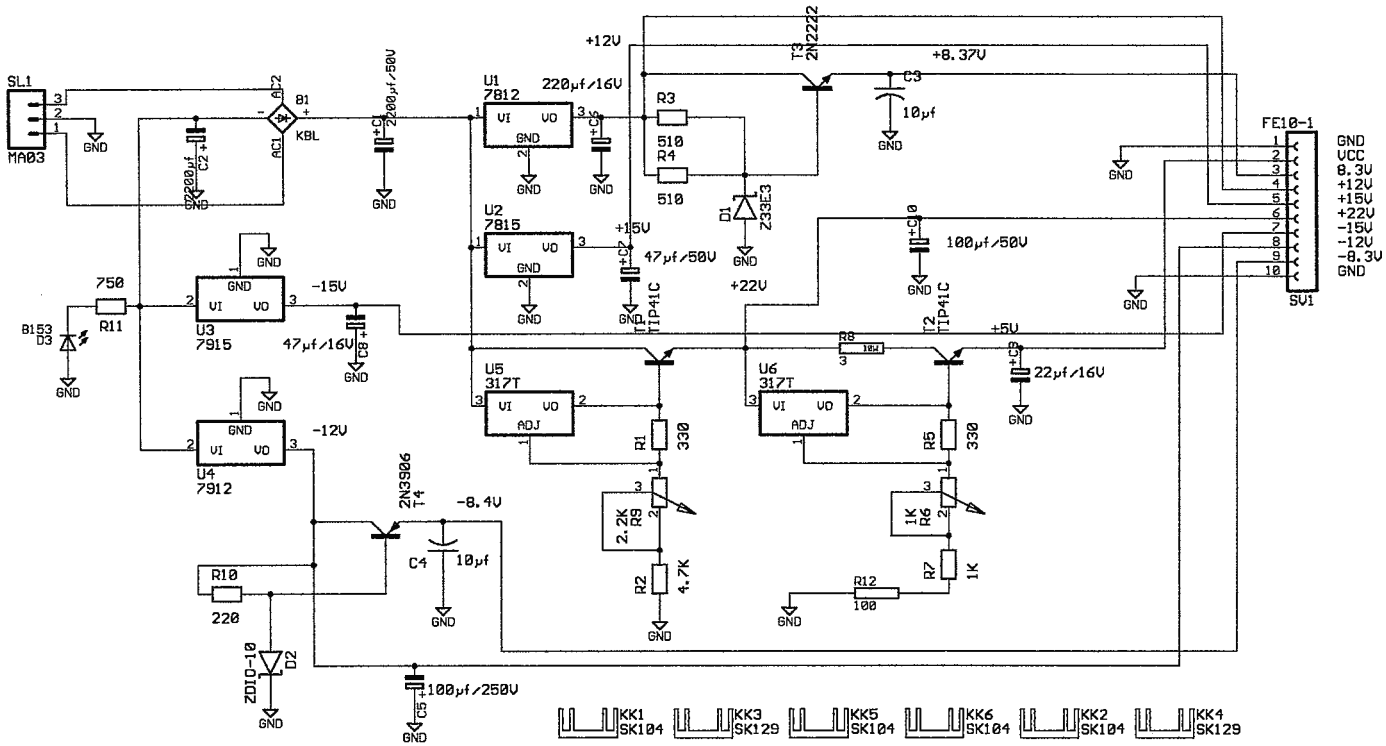
Vx = (+5V * Rtest) / (Rtest + Rescala)
Rtest = Rescala / (5V / Vx –1)

Tabla de Rescala para Óhmetro:

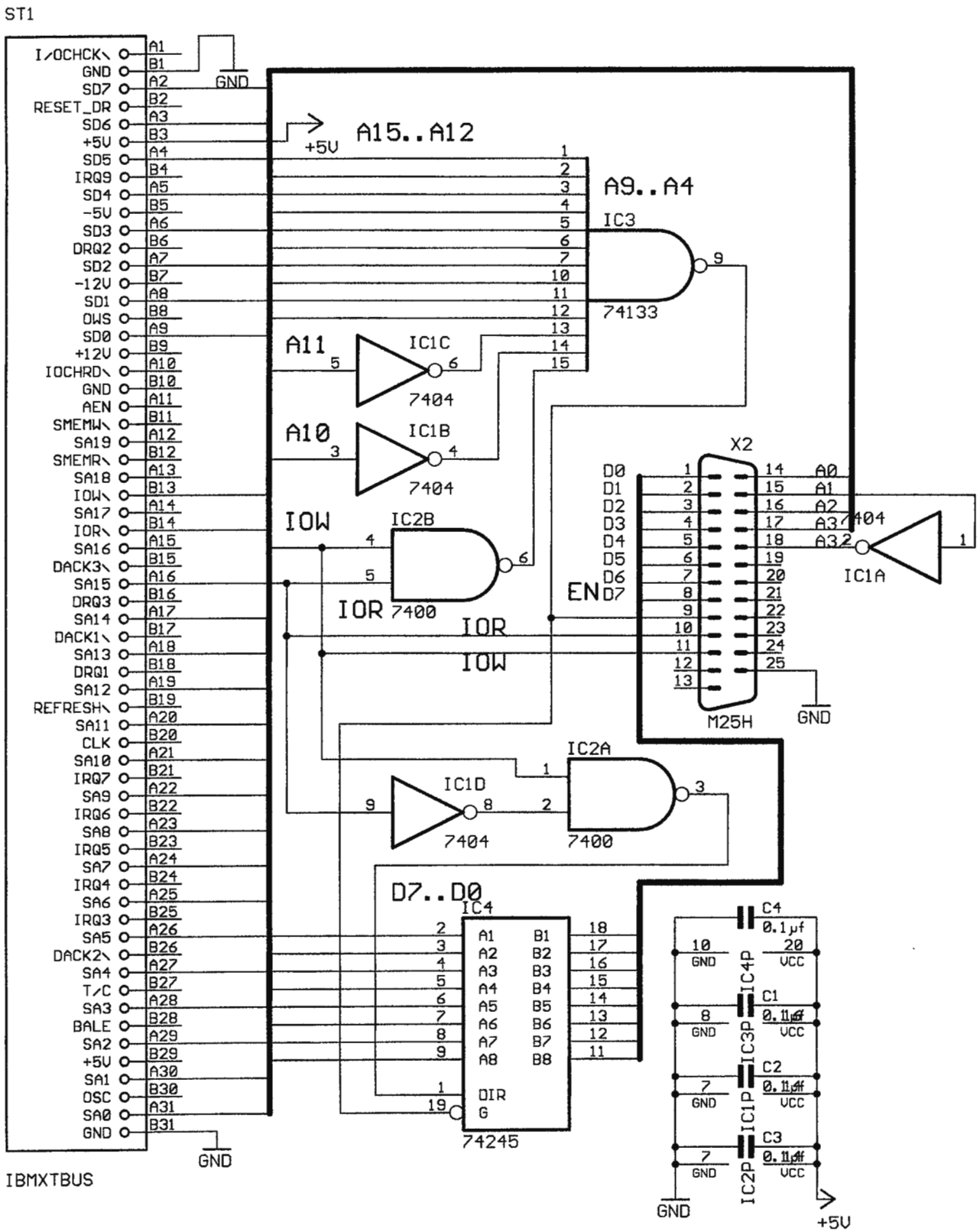
Nº	ESCALA	R ESCALA(Ω)
1	**500Ω**	99
2	**5KΩ**	1.022K
3	**50KΩ**	9.9K
4	**500KΩ**	99.9K

En el punto de entrada del Óhmetro se colocó una pequeña red de protección similar a la utilizada en el voltímetro, esto permite que cuando el usuario tenga seleccionado el óhmetro y accidentalmente coloque un voltaje, los diodos zener recortan la señal para que no pueda afectar a los AMP - OP de la etapa de entrada.

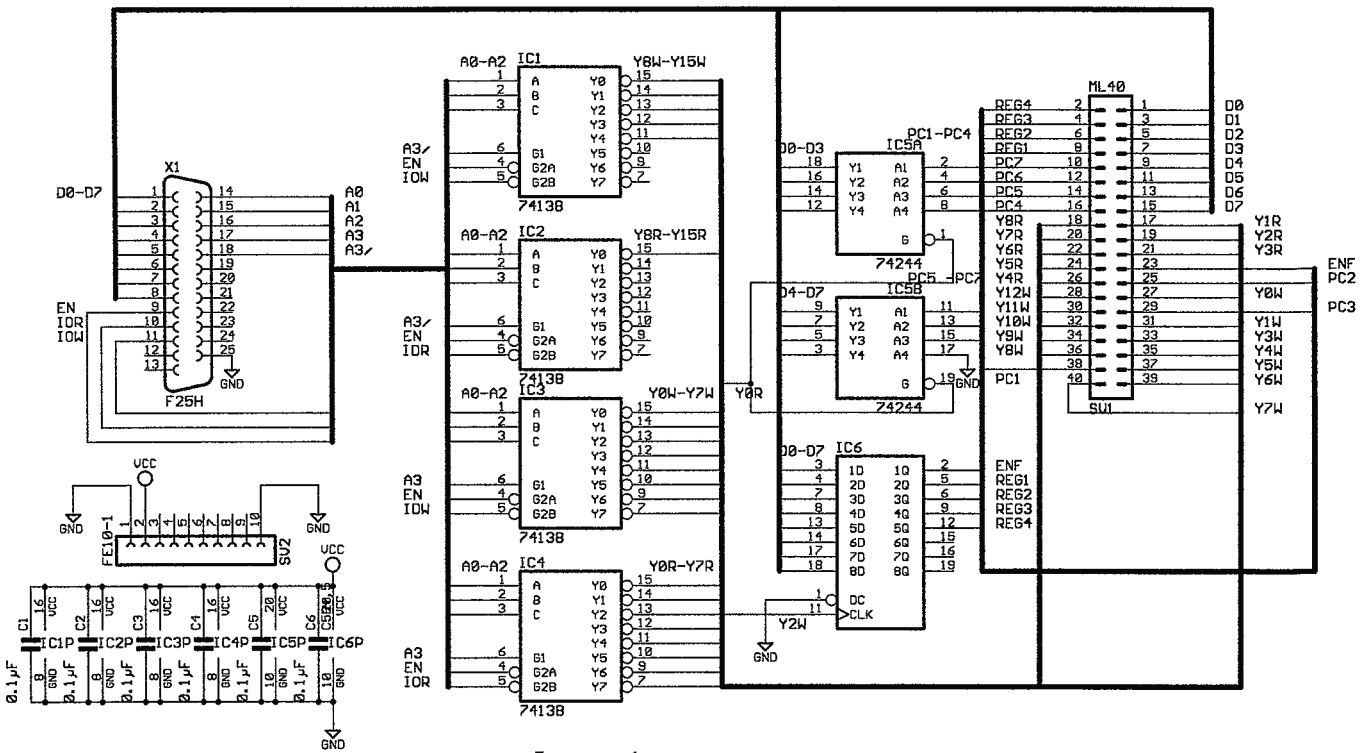
La función de C3 es para mantener una carga temporal (para resistencias en el orden de los MΩ porque se produce ruido y esto provoca inestabilidad en la medición). La U5a se utiliza como un aislador mientras que U5b en configuración de amplificador inversor aplica una ganancia de 0.4 que puede ser ajustada a través de una resistencia variable. El voltaje de salida de U5b a plena escala es de 2V, en este caso el signo no tiene importancia y la PC no lo tomará en cuenta para el calculo de la R de prueba. El procedimiento para obtener el valor binario que corresponde a la resistencia que se mide se logra de la misma forma que se hace con el amperímetro y voltímetro.



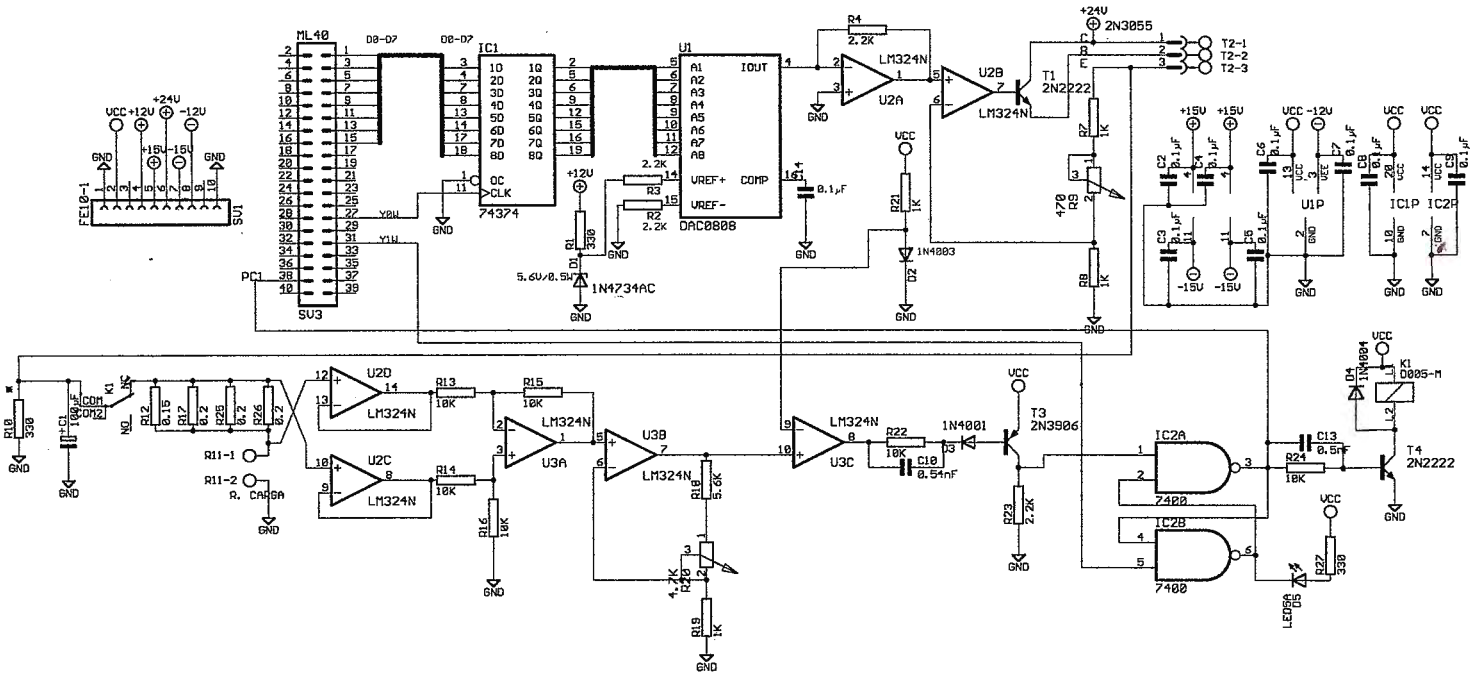
Fuente de alimentación.



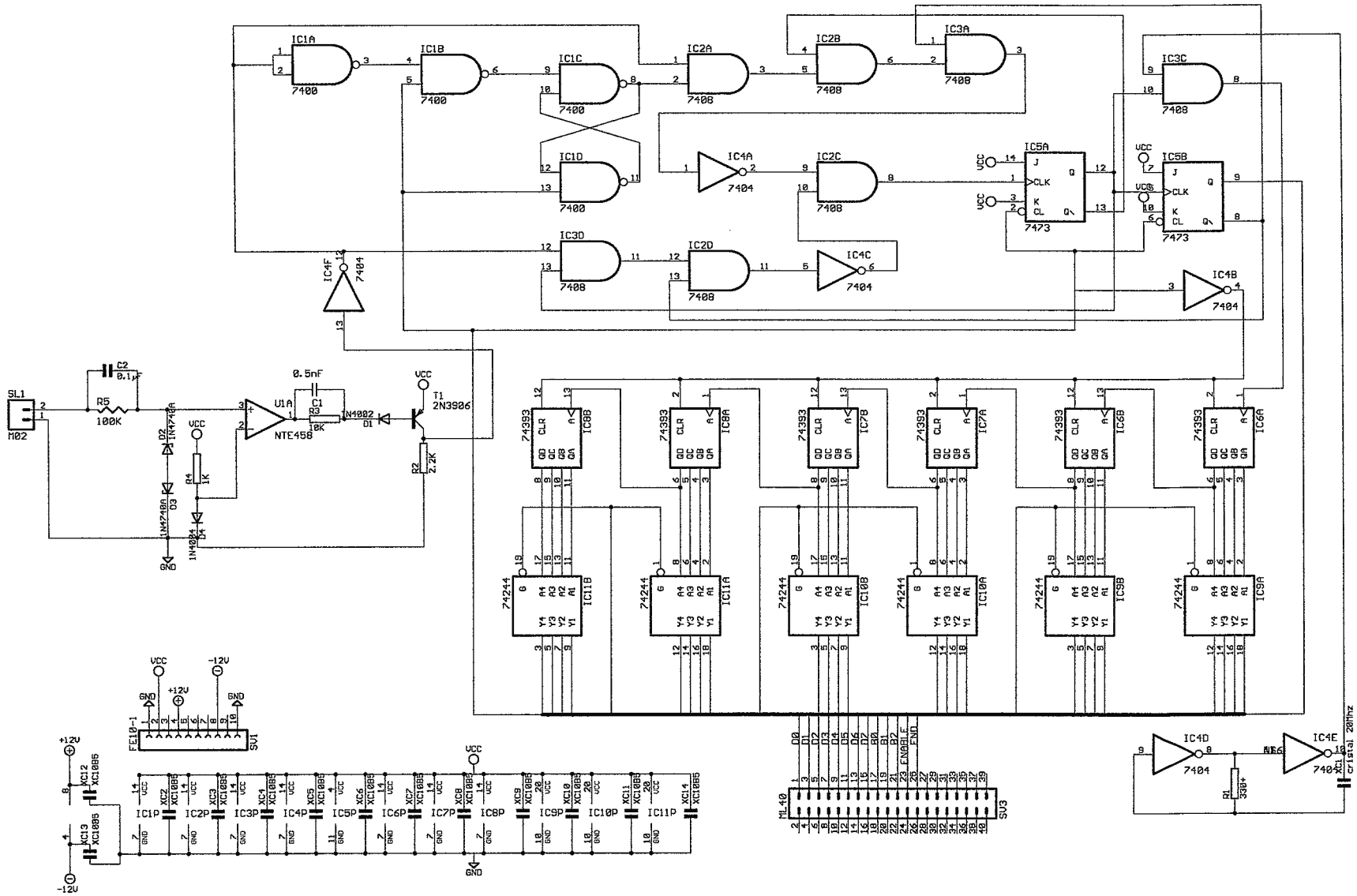
Interfaz de comunicación interna



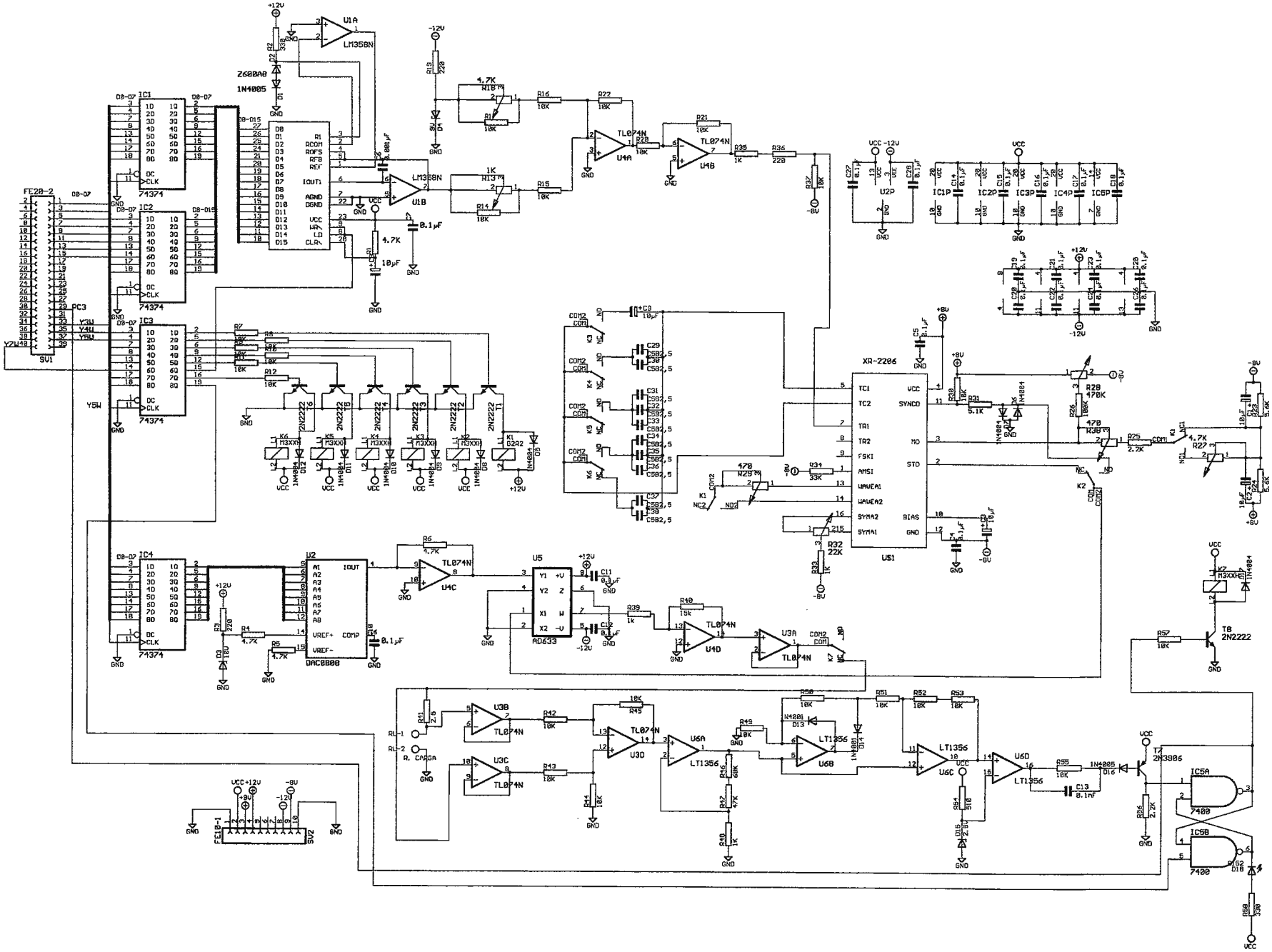
Interfaz externa.



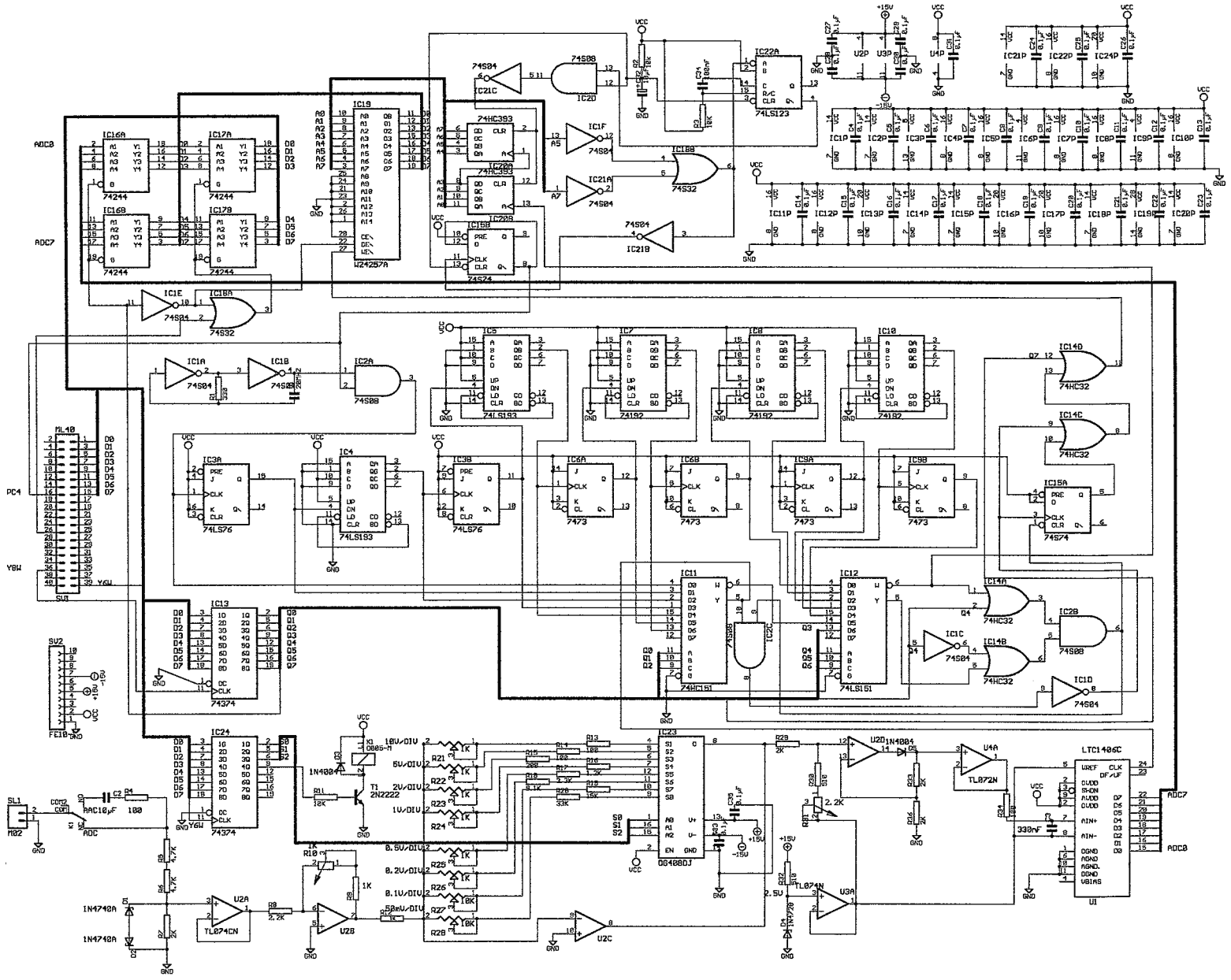
Fuente de voltaje.



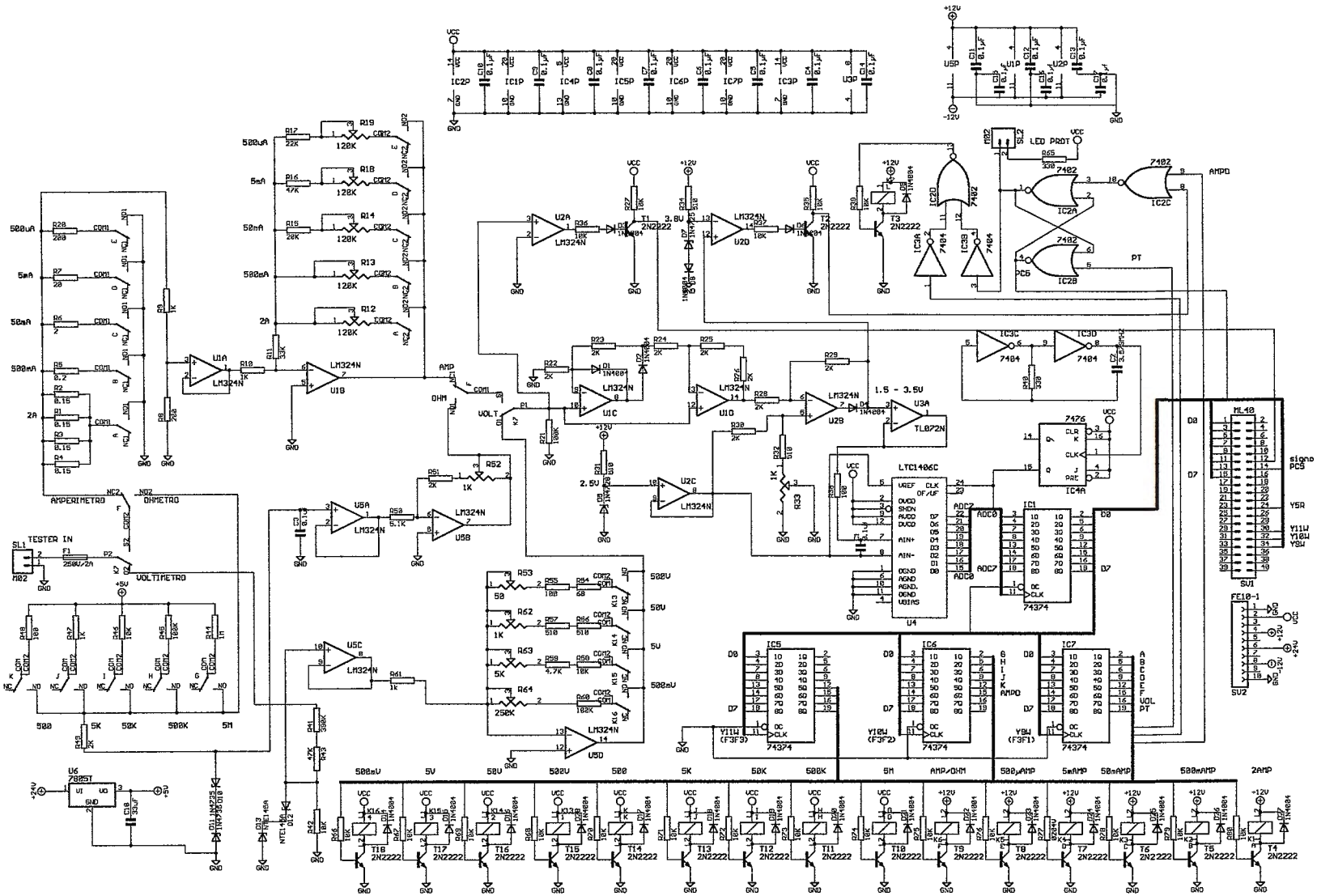
Frecuencímetro



Generador de funciones.



Osciloscopio.


AMPERIMETRO
OHMETRO
VOLTIMETRO
TESTER IN
LTC1406C
LM324N
TL072N
74374
7402
7404
7476
ML40
FE10-1
LED PRDT
AMPO
PT
signo
PCS
Y5R
Y11V
Y10V
Y9V
VOL
PT
500uA
5mA
50mA
500mA
2A
500
5K
50K
500K
5M
AMP/OHM
500uAMP
5mAMP
50mAMP
500mAMP
2AMP
500uV
5V
50V
500V

Multimetro.

CAPITULO 4

Técnicas de control por Software.

Determinación de la frecuencia para el frecuencímetro.

Cuando el frecuencímetro terminó la conversión y la computadora lee los tres BYTES de los contadores, estos valores tienen que agruparse en una sola variable para poder determinar la frecuencia medida.

El primer dato que contiene la parte menos significativa del dato no es alterado, el segundo y el tercer dato es multiplicado por 100 y 10000 respectivamente (en hexadecimal):

primero = primero
segundo = segundo * 0x100
tercer0 = tercero * 0x10000

Nciclos= primero + segundo + tercero

Teniendo él número de ciclos la frecuencia se determina por:

Fmedida = Fbase / Nciclos

Donde:
Fbase: La frecuencia de muestra (20MHz).
N ciclos: El valor binario que fue leído de los contadores.

El valor que poseen los contadores binarios a la salida corresponde al número de ciclos que ingresaron a la entrada de reloj de los contadores para un ciclo de la señal de entrada.

La frecuencia base es un indicador de la capacidad del frecuencímetro, por esta razón la frecuencia máxima a medir seria igual a la frecuencia base, lastimosamente esta condición no se cumple para el frecuencímetro del EAM, ya que para lograrlo se necesitaría un comparador con una velocidad de respuesta muy alta.

Determinación del voltaje de salida de la fuente:

A continuación se presenta la Fórmula para poder obtener el voltaje de salida deseado:

Fórmula:

fuente =(fuentev / 22.1875e-03) / 2.1856

Donde:

- Fuentev: contiene el voltaje digitado por el usuario en decimal.
- La constante 22.1875e-03: valor de la resolución del DAC.(5.657V/255 bit).
- Constante 2.1856: constante del amplificador de salida de la fuente.
- Fuente: Es el valor binario que se manda al registro del DAC, para poder lograr el voltaje deseado.

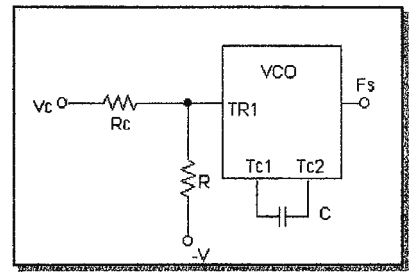
Para que el usuario pueda introducir un voltaje tiene que presionar el botón de la fuente y digitar el voltaje, si dicho voltaje es muy elevado la fuente pondrá el valor máximo que ella puede entregar que es 12V. Si el usuario comete un error al digitar el voltaje solo tiene que presionar el botón de ESC o el botón de DELETE. La fuente se protege al entregar una corriente de 1.5Amp como máximo. En ese momento deja de alimentar al circuito y se enciende un Led en el AEM al mismo tiempo la bandera de protección en la fuente es puesta en 1 lógico, al ser leída por el programa esta presenta el siguiente mensaje: **protección en la fuente** y se elimina solo al presionar el botón de la misma.

Determinación de la frecuencia y amplitud para el GEN – FUN.

Control de la frecuencia:

En el generador de funciones se utiliza un VCO (oscilador controlado por voltaje) en combinación con un DAC (convertidor digital – análogo de 16 Bits), para generar la frecuencia de la señal (Fs). El valor binario a la entrada de DAC es directamente proporcional a la frecuencia de salida. La Fs del VCO es inversamente proporcional al voltaje de control (Vc) que proviene del DAC Y viene dado por la siguiente ecuación:

$$Fs=(1 / R*C)\ [(1 + R / Rc)\ (1 - Vc/3)]$$



Factores que influyen en la frecuencia.

El voltaje de control para el XR-2206 presenta las siguientes características con relación Fs: Vc puede variar entre 0V a 3V con respecto a -V (-8.25V), si se desea lograr una Fs(min) en la escala seleccionada Vc = - 8.25V + 3V = - 5.25V, mientras que para una Fs(max) el voltaje de control sería el siguiente: Vc = -8.25 – 0 = -8.25. notese que se pretende lograr una diferencia neta de 3V (sí se supera este valor en Vc el IC no funcionara correctamente).

El DAC posee un voltaje de referencia de 3V, el rango de voltaje de salida tiene que oscilar de 0V a 3V(corresponde al rango de Vc). La resolución del DAC(LTC1557) es como sigue:

$$RESL = Vref / 2^n-1 = 3V / 65.535K = \mathbf{45.777\mu V / Bit}$$

Se puede apreciar que mediante está resolución se obtiene un control bastante completo de la Fs. El Vo(DAC) se determina de la siguiente manera:

$$Vo(DAC) = N° * RESL$$

Siendo N° el decimal equivalente de la entrada binaria del DAC; este valor es muy importante por ser el valor hexadecimal que la IG necesita mandar al DAC para que este logre un Vc proporcional a Fs. Al sustituir el Vo(DAC) en la ecuación de Fs y despejar N° se tiene:

$$\mathbf{N° = - [((Fs^* R^*C) - 1) Rc / R + 1] 3 / RESL}$$

Cuando el usuario ingresa un valor de frecuencia en HZ, éste es comparado con una base de datos para determinar de manera automática la escala que le corresponde, luego convierte dicho valor en su equivalente hexadecimal para final mente mandarlo al EAM junto con la escala.

- **Rangos de frecuencia para cada escala:**

Escala	Capacitor	Fs(min) Hz	Fs(max) Hz
1	C1 = 10.96µF	9	88
2	C2 = 1.187µF	89	818
3	C3 = 125.0nF	819	7.15K
4	C4 = 12.23nF	7.151K	65K
5	C5 = 1.50nF	65.1K	524K

Control de la amplitud:

El dispositivo encargado de controlar la amplitud de la señal es el DAC0808 en combinación con el multiplicador análogo. El XR-2206 proporciona una señal senoidal y triangular constante cuya amplitud es de 1VPP (X1 del AD633), la señal cuadrada posee una amplitud de 1.2Vpp.

El DAC0808 tiene una resolución de: RESL = 10V / 255 = 39.2156mV / Bit. Y un Voltaje de salida constante en Y1 que se define de la siguiente ecuación:

$$\mathbf{Vo(DAC0808) = N° * RESL}$$

N° representa el valor hexadecimal que la IG necesita mandar al DAC0808 para lograr la magnitud deseada de la señal. La función que se obtiene a la salida del multiplicador esta dada por la siguiente ecuación:

$$\mathbf{W = (X1^* Y1) / 10}$$

La ganancia final que se le asigna a la señal es de 15, no obstante como la señal cuadrada posee una amplitud diferente, el programa se encarga de compensar dicha diferencia y lo hace de forma automática al seleccionar una señal:

❶ Para señales senoidal y triangular: N° = (Vpp / 39.21568627e-03) / 1.5

❷ Para señales cuadradas: N° = (Vpp / 39.21568627e-03)/1.875

Para lograr que las ecuaciones se cumplan, antes es necesario calibrar paso por paso diferentes etapas del circuito como: offset, Vc, amplitud de señal senoidal y triangular, hasta lograr que el dato enviado por la PC sea compatible.

Proceso para determinar las escalas del multímetro.

Amperímetro:

Cuando el usuario selecciona el amperímetro tiene acceso a las diferentes escalas de este para poder realizar las mediciones. Dependiendo de la escala seleccionada, así será su resolución.

Tabla de Resolución para AMPERIMETRO:

N° Escala	ESCALA	RESOLUCION(A/bit)	R. de escala
1	**2Amp**	7.84313m	0.05Ω
2	**500mA**	1.9607m	0.2Ω
3	**50mA**	196.07µ	2Ω
4	**5Ma**	19.6078 µ	20Ω
5	**500µA**	1.95078 µ	200Ω

Para poder presentar la corriente medida se utilizan tres Fórmulas:

Fórmula para corriente DC:

Adctest = Adctest * Iresolución *factor* signo

Donde:

- Adctest: variable que contiene el dato.
- Factor: constante que se utiliza para desplazar el punto decimal
- Signo: indica si la corriente es positiva o negativa (positivo=1, negativo=-1)

Fórmulas para corriente AC:

En AC me puede presentar la corriente máxima y la corriente RMS

Corriente RMS: adctest = (adctestmax/√2) * Iresolución * factor

Corriente Máxima: **adctest = adctestmax * Iresolución * factor**

Donde:

Adctestmax: Es el máximo valor de la corriente.

Voltímetro:

El voltaje en DC está determinado por la siguiente ecuación:

Adctest = ((Adctest * Vresolución)/ Factor) * signo

Donde:

- Adctest: Contiene el valor del voltaje medido.
- Vresolución: La resolución cambia de pendiendo de la escala seleccionada.
- Factor: constante que sirve para desplazar el punto decimal por motivos de presentación en pantalla.
- Signo: Indica si la corriente es positiva o negativa (positivo=1, negativo=-1); este Valor es proporcionado por una bandera.

Cuando es seleccionado el Voltímetro AC este proporciona el Voltaje RMS de la señal y el Voltaje máximo de dicha señal.
El voltaje máximo esta determinado por la siguiente ecuación:

Voltaje máximo: Adctest= (Adctestmax * Vresolución)/ factor

Voltaje RMS: Adctest=((adctestmax/√2)* Vresolución)/ factor

Tabla de Resolución para VOLTIMETRO:

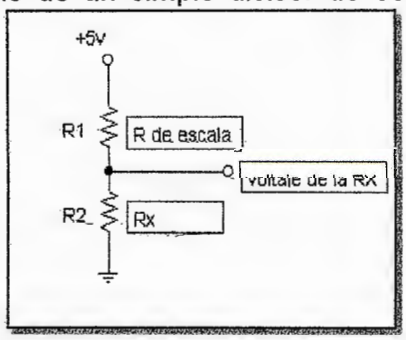
N°	ESCALA	RESOLUCION(v/bit)	FACTOR	R. de escala
1	**500V**	1.960784	1	100Ω
2	**50V**	0.1960784	1	1KΩ
3	**5V**	19.6078m	1	10KΩ
4	**500mV**	1.960784m	0.001	100KΩ

Óhmetro:

La forma de medir la resistencia es por medio de un simple divisor de voltaje, donde ya se conocen las siguientes interrogantes:

1- El voltaje de alimentación del divisor

2- El valor de una de las resistencias que le llamaremos R de escala (esta resistencia cambia de acuerdo a la escala seleccionada).



El objetivo es medir el voltaje de la resistencia de prueba (Rx); conociendo el voltaje de RX se despeja de la ecuación la única incógnita, que es el valor de Rx.

La ecuación es la siguiente:

```
RX = ((RE * (adctes t* 7.843177255e-03/0.48))/
        (5.0 - (adctest * 7.843177255e-03/0.48)))/factor;
```

donde:

RE: es la resistencia de la escala seleccionada.

Adctest: Es el dato que envió el ADC.

Constantes:

7.843177255e-03 = constante de la resolución del ADC.

0.48 = Atenuación de entrada del circuito.

Factor: constante que sirve para desplazar el punto decimal por motivos de presentación en pantalla.

Tabla de Resolución para Óhmetro:

N°	ESCALA	RESISTENCIA DE ESCALA(Ω)	FACTOR	RESOLUCION del ADC(V/bit)
1	**500Ω**	99	1	7.843177255e-03
2	**5KΩ**	1.022K	1000	
3	**50KΩ**	9.9K	1000	
4	**500KΩ**	99.9K	1000	
5	**5MΩ**	1.027M	1000000	

Registros de control para el Multímetro.

- **Registro Y9W dirección F3F1(control de escalas de amperímetro y selección de instrumento).**

- **Escalas del amperímetro:**

Q4	Q3	Q2	Q1	Q0	Escala
0	0	0	0	1	***2 A***
0	0	0	1	0	***500mA***
0	0	1	0	0	***50mA***
0	1	0	0	0	***5mA***
1	0	0	0	0	***500µA***

- **Selección de dispositivo:**

protección	Q7	Q6 (Volt)	Q5(I/Ω)	Función
Off	0	0	X	***Voltimetro***
On	0	1	0	***Amperimetro***
Off	0	1	1	***Óhmetro***
	_---_	X	X	***Fin de protección***

- **Registros Y10W dirección F3F2 (escalas para el óhmetro) :**

Q4	Q3	Q2	Q1	Q0	Escala
0	0	0	0	1	**5M**
0	0	0	1	0	**500K**
0	0	1	0	0	**50K**
0	1	0	0	0	**5K**
1	0	0	0	0	**500**

Ampo	Protección
0	On
1	OFF

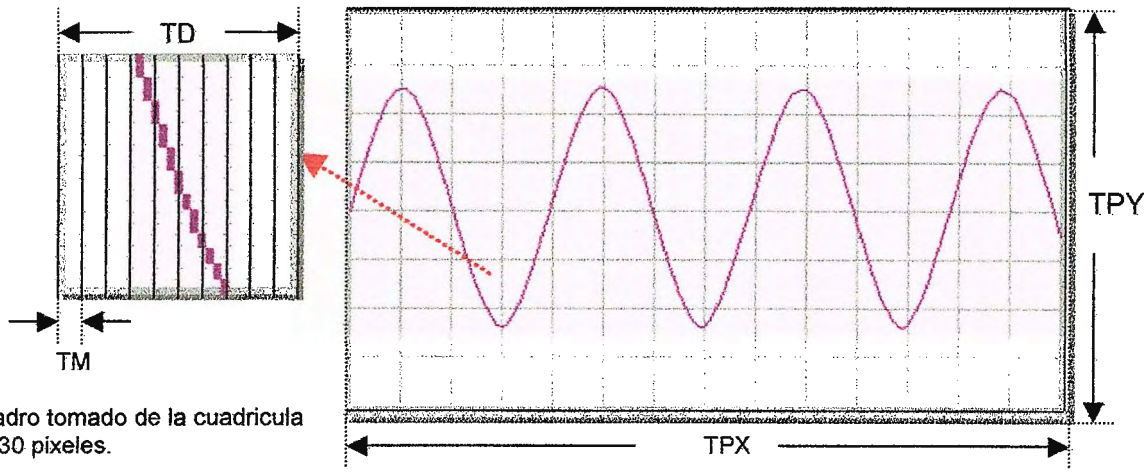
- **Registro Y11W dirección F3F3(escala para el voltímetro):**

Q7	Q6	Q5	Q4	Escalas
0	0	0	1	**500V**
0	0	1	0	**50V**
0	1	0	0	**5V**
1	0	0	0	**500mV**

Determinación de las escalas TIME/DIV y VOLT/DIV:

Criterios importantes de la pantalla:

Pantalla completa del osciloscopio 420 * 240 pixeles (cuadricula).



Cuadro tomado de la cuadricula 30*30 pixeles.

- Para aumentar las dimensiones de presentación de la cuadricula del osciloscopio, se coloco una separación de 3 pixeles entre cada punto adquirido.

TD : Tiempo por división.
TC : Tamaño de un cuadro en pixeles.
*TC = (10 muestras) * 3pixeles = 30 pixeles por cada cuadro:*
En este caso se tomo como parámetro convencional *10 muestras* por cada cuadro. Mientras mayor sea dicho parámetro, mucha mayor información se lograra apreciar en la pantalla; no obstante, al disminuirlo, el sistema será más rápido y se puede aumentar la frecuencia máxima a medir, pero sacrificando la calidad de la señal en la pantalla.

N°CY: Número de cuadros en Y = 8.
N°CX: Número de cuadros en X = 14 (estimado para una presentación de muy buena calidad).
N°PC: Número de pixeles efectivos por cuadro = 10.
*TPY = TC * 8 = 30 * 8 = 240 pixeles (total de pixeles en Y).*
*TPX =TC * 14 = 30 * 14 = 420pixeles (Total de Pixeles en eje X).*

Definición de términos VOLT/DIV:

*VPP (max) = N°CY * N°PC = 8 * 10 = 80VPP.*
Δ3 = 0.125 (ganancia producida por la atenuación de Vin).
RIN: Resistencia de entrada para el amplificador inversor.
RE: Resistencia de escala (seleccionable por un multiplexor análogo) = **ΔT/Δ3 * RIN**.

Escalas de VOLT / DIV:

Escala	VOLT/DIV	Vin(VPP) en plena escala	ΔT=1V/VP	RE(Ω)
1	**10V**	80V	0.025	200
2	**5V**	40V	0.050	400
3	**2V**	16V	0.125	1K
4	**1V**	8V	0.250	2K
5	**0.5V**	4V	0.500	4K
6	**0.2V**	1.6V	1.25	10K
7	**0.1V**	0.8V	2.5	20K
8	**50mv**	0.4V	5	40K

Definición de términos TIME/DIV:

TD = TIME / DIV.

TM: Tiempo de separación entre cada muestra (período para la frecuencia base de la escala seleccionada).

Se toman 10 muestras para cada TD (equivalente a un cuadro de la cuadricula). Cada punto corresponde a un dato convertido por el ADC del adquisidor.
Uno de los datos técnicos más importantes de este parámetro (TM) para un osciloscopio digital es: **Fmax = 1 / TM.** Parámetro fácil para determinar si se conoce la frecuencia máxima de muestreo o utilizando las cotas verticales, de la interfaz gráfica. Esta información es de carácter comercial y dependerá de la velocidad de respuesta que tienen los dispositivos utilizados en el adquisidor de datos.
TM= (TIME / DIV) /10.
Tta: Tiempo total aparente (para propósitos de presentación y efectividad de los datos).
Tta = TD * 14
Ttr: Tiempo total real (es el tiempo real que le toma al adquisidor de datos realizar el muestreo).
Ttr= TD * 16 (los 20 datos sobrantes son descartados por la dudosa efectividad de los mismos).

Escalas de TIME / DIV:

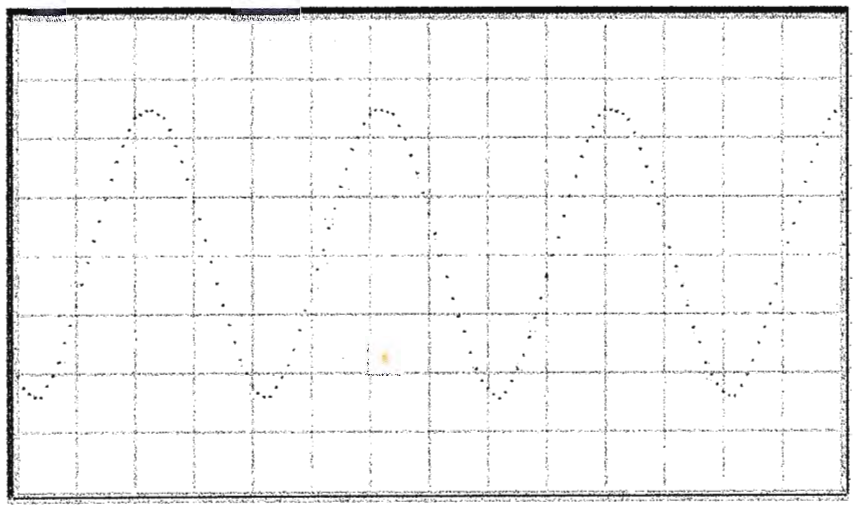
Escala	TIME/ DIV	FS = 1/TD (HZ)	TM = TD/10	FM = 1/TM (HZ)
1	**500ns**	2M	50ns	20M
2	**1.0µs**	1M	100ns	10M
3	**5.0µs**	200K	500ns	2M
4	**10µs**	100K	1µs	1M
5	**50µs**	20K	5µs	200K
6	**0.1ms**	10K	10µs	100K
7	**0.5ms**	2K	20µs	20K
8	**1.0ms**	1K	500µs	10K
9	**5.0ms**	200	0.1ms	2K
10	**10ms**	100	0.5ms	1K
11	**50ms**	20	1ms	200
12	**100ms**	10	10ms	100

Nota: Todas las escalas se han determinado utilizando valores comerciales que usualmente se adoptan en este genero de osciloscopios.

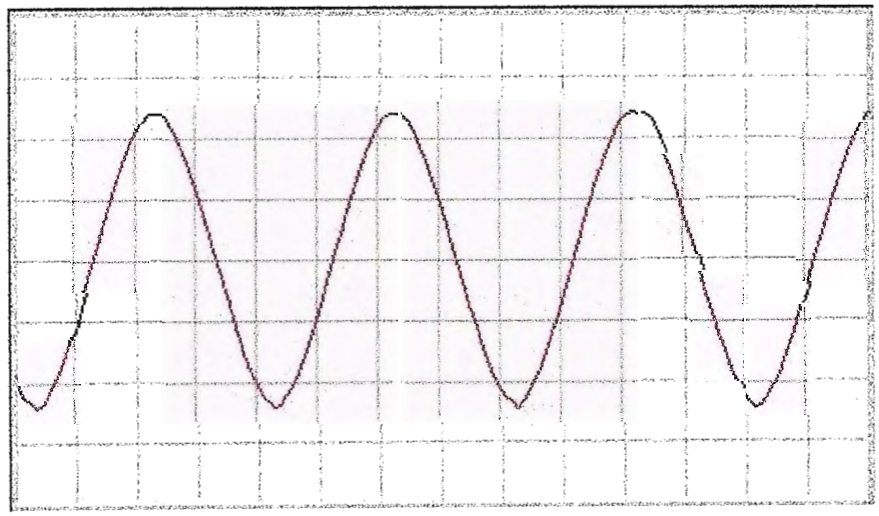
Proceso de presentación en pantalla:

Inicialmente la IG (interfaz gráfica) manda al EAM la señal de inicio de adquisición la que a su vez se combina con las escalas de VOLT/DIV y TIME/DIV. El AD (adquisidor de datos) toma 160 BYTES y los almacena en tiempo real en una memoria RAM luego el AD manda a la PC una señal de fin de adquisición. La PC lee los 160 BYTES ($A0_{16}$) y se almacenan en una matriz del tipo arreglo en la IG. El programa se encarga de eliminar los primeros 20 BYTES (son datos dudosos que el fabricante del ADC recomienda eliminarlos). El programa ubica 140 puntos en la pantalla cuyas coordenadas corresponde al valor instantáneo de voltaje tomado por el AD y de las escalas TIME/DIV y VOLT/DIV tal como se muestra a continuación:

- **Despliegue de puntos en la pantalla:**



- **Unión de puntos para formar la gráfica:**



- El factor de escala VOLT/DIV y TIME/DIV ya viene desde el AD y es generado (por hardware) mediante el MUX/DEMUX análogo y los divisores de frecuencia respectivamente.

Después de colocar los puntos, se procede a unirlos mediante líneas para formar la gráfica completa. El proceso de adquisición y presentación se repite periódicamente; tomando en cuenta que se genera un retardo de tiempo para que el usuario pueda apreciar la señal en pantalla. Las coordenadas de la pantalla tomando en cuenta el área de presentación (420 * 240 pixeles) son las siguientes:

- ⇨ Parte superior izquierda (100,100)
- ⇨ Parte inferior izquierda (100, 340)
- ⇨ Parte superior derecha (520, 100)
- ⇨ Parte inferior derecha (520,340).

En el programa se genera un contador denominado "add" el cual se incrementa de uno en uno hasta llegar a 140, dicho contador representa la coordenada X (tiempo) del punto a colocar en pantalla y esta dado por la siguiente ecuación:

$$X = 100 + add * 3$$

Nota: La constante 3 (separación entre pixeles) multiplica el valor del contador.
La coordenada Y (voltaje) representa la magnitud de la señal y se determina mediante la siguiente ecuación:

$$Y = 340 - RESL\ (Dato)$$

La variable "Dato" corresponde a uno de los 140 Bytes efectivos de la adquisición. RESL es la resolución de las coordenadas en Y de la pantalla, entre el número máximo en decimal y el (2^n-1 = 2^8 – 1= 255) que proporciona el ADC del adquisidor:

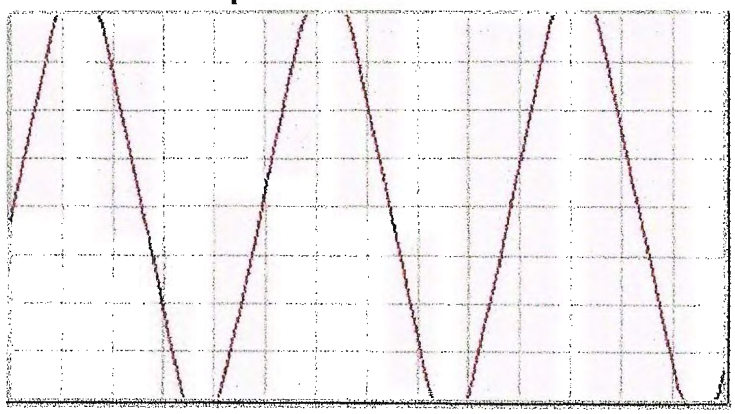
$$RESL = 240 / 255 = 0.94117647$$

Por ejemplo: Si el dato correspondiente al add = 0, es = 127 (7F); esto ubicaría el punto en las siguientes coordenadas:

$$X = 100 + 0 * 3 = 100$$
$$Y = 340 - 0.94117647\ (127) = 220.47$$

El valor en Y es aproximado a 220 ya que las coordenadas en pixeles son enteras. Nuestro punto a gráficar seria: (100, 220); dicho de otra manera, la coordenada corresponde a: (0,120) justo al centro de la pantalla. De igual manera un Dato = 255 ubicará el punto en la parte superior de la pantalla y un Dato = 0 colocaría el punto en la parte inferior de la pantalla. Si tratamos de asociar estos valores con las escalas de VOLT/DIV nos indica que estamos a plena escala; es decir, que el valor pico de entrada supera o iguala el valor seleccionado de la escala de amplitud.
Es válido considerar que en un osciloscopio cuando se supera la escala seleccionada la señal en pantalla comienza a ser recortada, sin embargo esta condición puede producir averías en el ADC si no se tiene una red de protección.

- **Señal superior a la escala VOLT/DIV.**



Determinación del período y magnitud de la señal.

Cotas verticales y horizontales.

Un método moderno y fácil para determinar la frecuencia y amplitud de una señal es el uso de las cotas.
La interfaz gráfica del EAM brinda al usuario una eficaz manera de tomar datos de una señal bajo medición. Con el uso de las cotas se vuelve más sencillo determinar parámetros de la señal, a diferencia de un osciloscopio convencional que utiliza las divisiones y el usuario multiplica esta información por la escala. La interfaz gráfica realiza todos estos procesos en forma automática sin necesidad de hacer mas cálculos para el usuario. Las ventajas que brinda el uso de cotas en un osciloscopio digital son las siguientes:

1- Facilidad y comodidad de uso.
2- No se necesita realizar cálculos extras.
3- Exactitud y precisión en la determinación de la información.
4- Rapidez y efectividad en la toma de datos.
5- Permite que los datos queden almacenados a un después de realizar las mediciones.

La única desventaja es en términos de diseño, la que se puede atribuir a la complejidad de implementar este método en el sistema operativo, pero los beneficios que ofrece al usuario hace que valga la pena adoptar esta metodología.

- Cuadricula: La cuadricula puede utilizarse únicamente para referenciar ciertas señales, pero de preferencia se recomienda utilizar las cotas por los beneficios antes citados. En algunos casos se puede hacer uso de cotas y cuadricula de manera simultanea. Además la cuadricula puede ser un método muy bueno para asociar las cotas con las escalas del osciloscopio.

Cotas verticales (período de la señal):

Las cotas verticales sirven para determinar el período de la señal, así como también la frecuencia de la misma. Tanto el EAM como la IG saben que escala de TIME/DIV sé esta utilizando mientras que el programa conoce las coordenadas para cada cota (C1 y C2) y la separación de las mismas, luego el programa determina la diferencia absoluta que separa a C1 de C2 de la siguiente forma:

$$DF = | C2 - C1 |$$

La distancia que separa a las cotas C1 de C2 se multiplica por el TIME/DIV y se divide entre el numero de pixeles que separa a una división (30 pixeles), esto nos da el tiempo neto que existe entre C1 y C2. Si colocamos las cotas justo donde corresponde un ciclo de señal, el programa estará calculando el período(PS) y la frecuencia (FS) de la señal.

$$Ps = (DF * TIME/DIV) / 30Pixeles$$

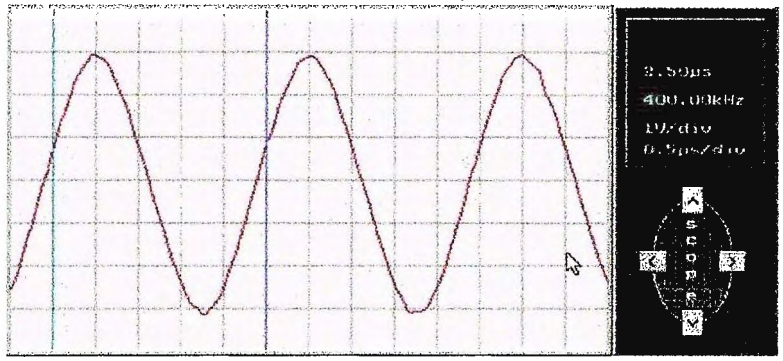
$$FS = 1/Ps$$

El uso correcto de las cotas puede brindar funciones importantes como por ejemplo: encontrar el tiempo en que ocurre un determinado evento en particular.

Por ejemplo: Si la escala de TIME/DIV es de 0.5µs y la ubicación de C1 y C2 es como se muestra en la gráfica se tiene que:

C1 =130pixeles, C2 = 280Pixeles → DF = |280 –130| = 150Pixeles.

- **Gráfica para determinar el período de la señal:**



DF representa la separación absoluta de las cotas en Pixeles, luego el programa calcula el tiempo que separa a las cotas:

Ps = (150Pixeles * 0.5µs) / 30 Pixeles = 2.5µs

Nótese que este resultado representa el período de la señal y es determinado de forma automática por la IG. Otra manera de obtener este valor es utilizando la cuadricula, como puede observarse son 5 cuadros de división para un ciclo, esto es:

5DIV * 0.5µs/DIV = 2.5µs

La frecuencia de la señal es calculada por la IG Fs = 1 / 2.5µs = **400KHZ**, si se logra observar estos son los valores que aparecen en la IG y corresponden exactamente al valor esperado. La exactitud depende de la resolución, esta se determina usando la siguiente ecuación: RESL = (TIME/DIV) / 30pixeles.

- **Tabla de Resolución para cada escala TIME/DIV:**

Escala	TIME/DIV	RESL (Seg/Pixel)
1	500ns	**16.666667n**
2	1.0µs	**33.333333n**
3	5.0µs	**166.666667n**
4	10µs	**333.333333n**
5	50µs	**1.666667µ**
6	0.1ms	**3.3333333µ**
7	0.5ms	**16.6666667µ**
8	1.0ms	**33.33333333µ**
9	5.0ms	**166.6666667µ**
10	10ms	**333.333333µ**
11	50ms	**1.66666667m**
12	100ms	**3.333333m**

Cotas horizontales (amplitud de la señal):

El funcionamiento de las cotas horizontales es similar, la diferencia esta en las unidades de trabajo (voltios). En este caso se determina también la separación de las cotas en pixeles (DF) y se multiplica por la resolución, la que depende de la escala de VOLT/DIV:

DF = |C3 – C4|, RESL = (VOLT/DIV) / 30pixeles → Ampl = DF * RESL

Las cotas C3 y C4 pueden utilizarse para medir un voltaje determinado entre dos puntos cualquiera de la señal que se presenta en pantalla y con mucha mayor exactitud que al utilizar la cuadricula (osciloscopio convencional), sin embargo se puede combinar la cuadricula (como líneas de referencia) con las cotas para un caso en particular.
La exactitud depende de la resolución, esta se determina usando la siguiente ecuación:

RESL = (VOLT/DIV) / 30pixeles.

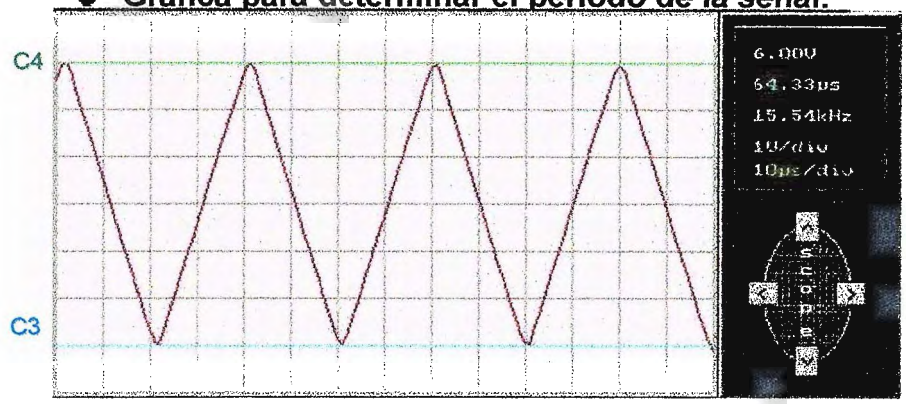
- **Tabla de Resolución para cada escala VOLT/DIV**

Escala	VOLT/DIV	RESL (Volt/Pixel)
1	10V	333.333333m
2	5V	166.666667m
3	2V	66.666667m
4	1V	33.333333m
5	0.5V	16.666667m
6	0.2V	6.666667m
7	0.1V	3.333333m
8	50mV	1.666667m

Por ejemplo: Si la escala de VOLT/DIV es de 1V, la ubicación de C3 = 310 y C4 = 130 se tiene que:

C3 = 310pixeles, C4 = 130Pixeles → DF = |130 – 310| = 180Pixeles.

- **Gráfica para determinar el período de la señal:**



Ampl = (180Pixeles * 1V) / 30 Pixeles = 6VPP

Este resultado representa la amplitud de la señal y es determinado de forma automática por la IG. Otra manera de obtener este valor es utilizando la cuadricula, como puede observarse son 6 cuadros de división lo que nos lleva a:

Ampl = 6DIV * 1V/DIV = 6VPP

Es importante mencionar que la IG y el AD del EAM están diseñados para medir sin problema señales cuya frecuencia es de 20MHZ (separación mínima entre dos puntos tomados por el AD) y amplitud de 80VPP; sin embargo, existen otros dispositivos que no

pueden responder a tales frecuencias, atenuando las señales hasta el punto que no son medibles (como es el caso de AMP-OP y multiplexores análogos). No obstante, el osciloscopio ha sido probado a frecuencias de 3MHZ y se ha logrado medir con cierto grado de inexactitud.

- **La interpolación:**

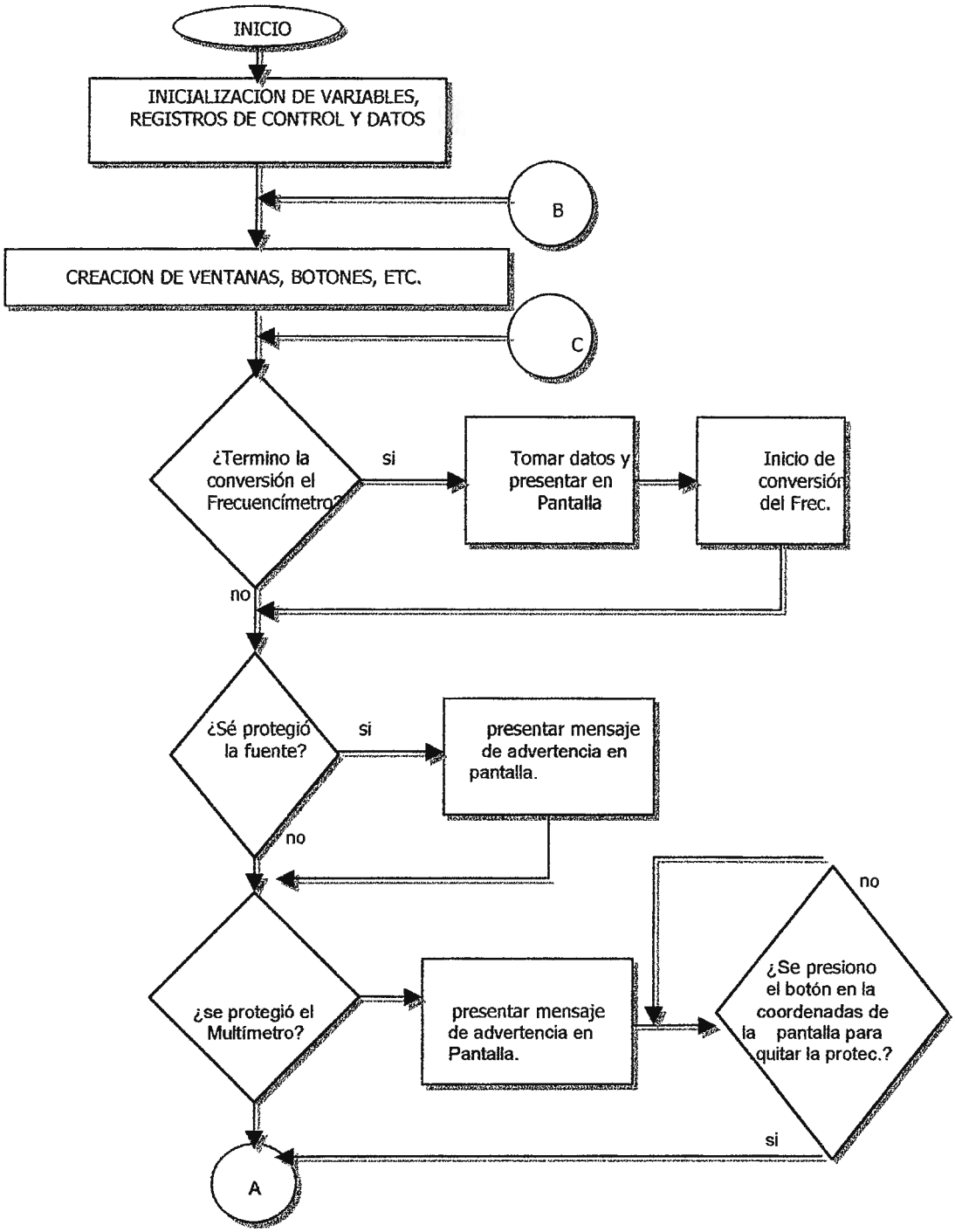
El AD del EAM realiza un muestreo en ***tiempo real***, la presentación de la señal en pantalla se realiza sin necesidad de adicionar puntos intermedios entre cada dato adquirido es decir; sin utilizar interpolación. Este método lo utilizan algunos osciloscopios digitales para aumentar su capacidad de medición.

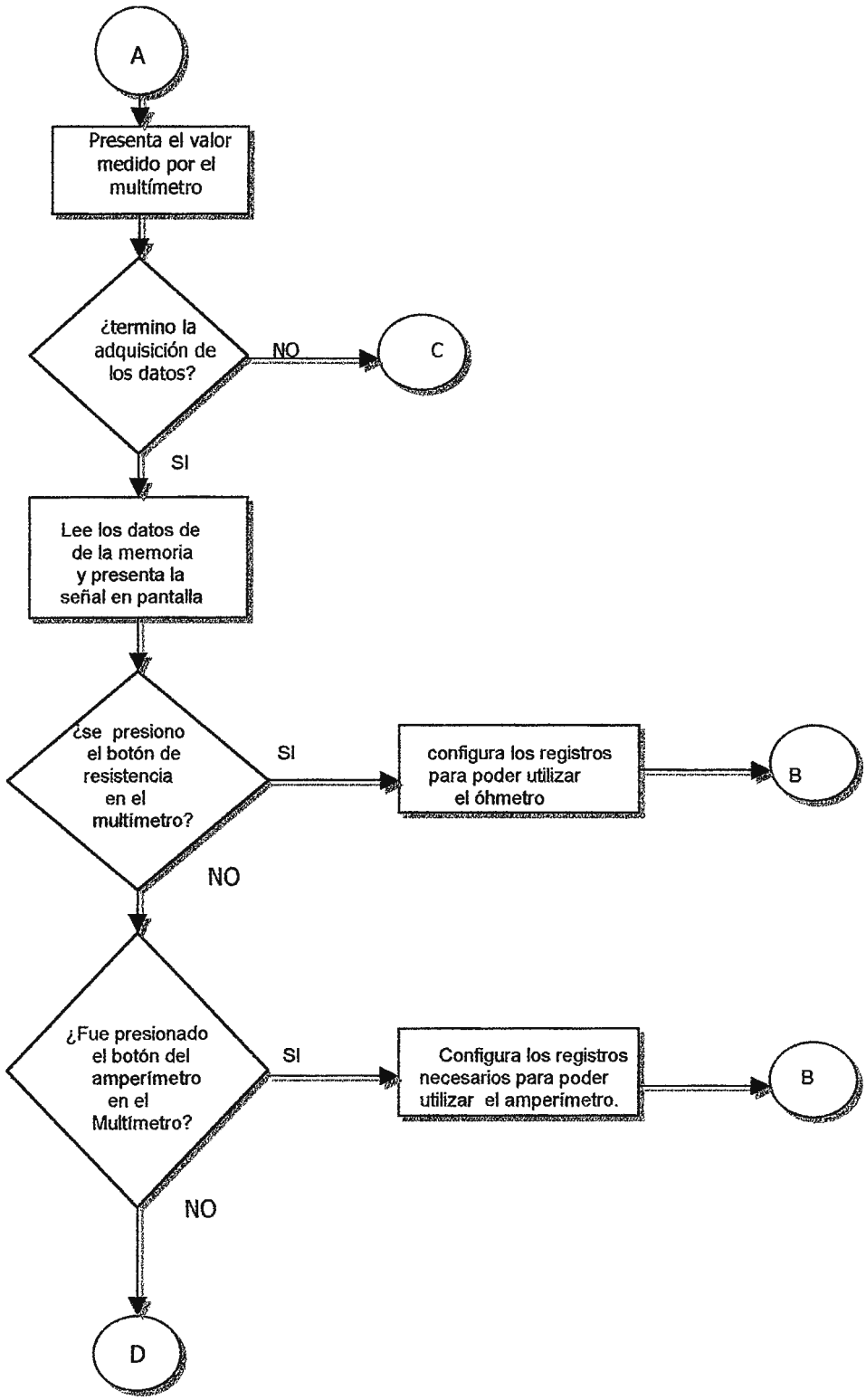
Flujograma general del programa.

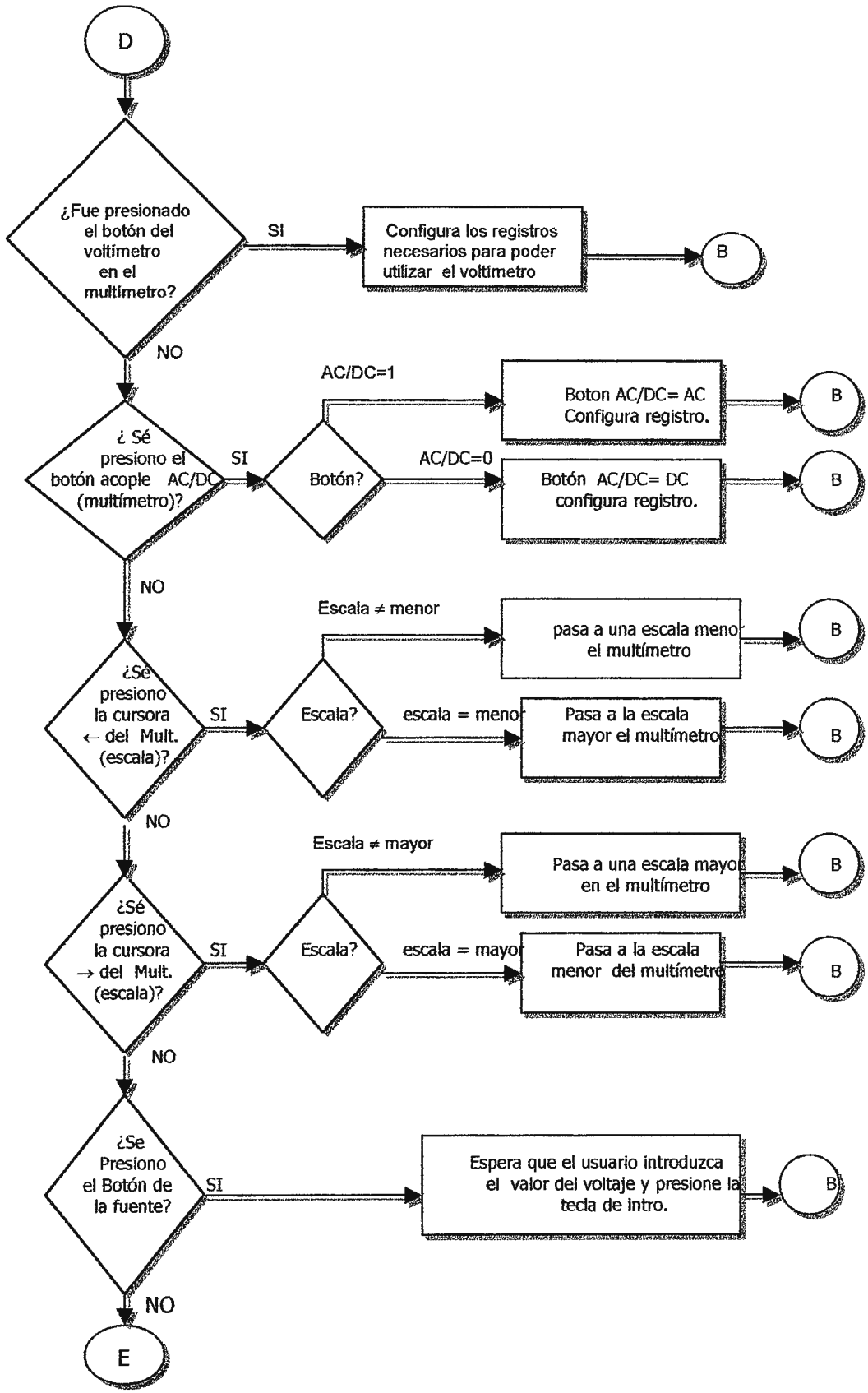
El usuario puede digitar la frecuencia, el voltaje, tipo de señal, puede medir la resistencia, la corriente, el voltaje, la frecuencia, a demás puede apreciar la forma de una señal; todo esto, debido a un conjunto de instrucciones ordenadas, que pueden realizar funciones de control, comunicación y presentación, el programa es esencial y el equipo no podrá funcionar sin esta parte de gran importancia.

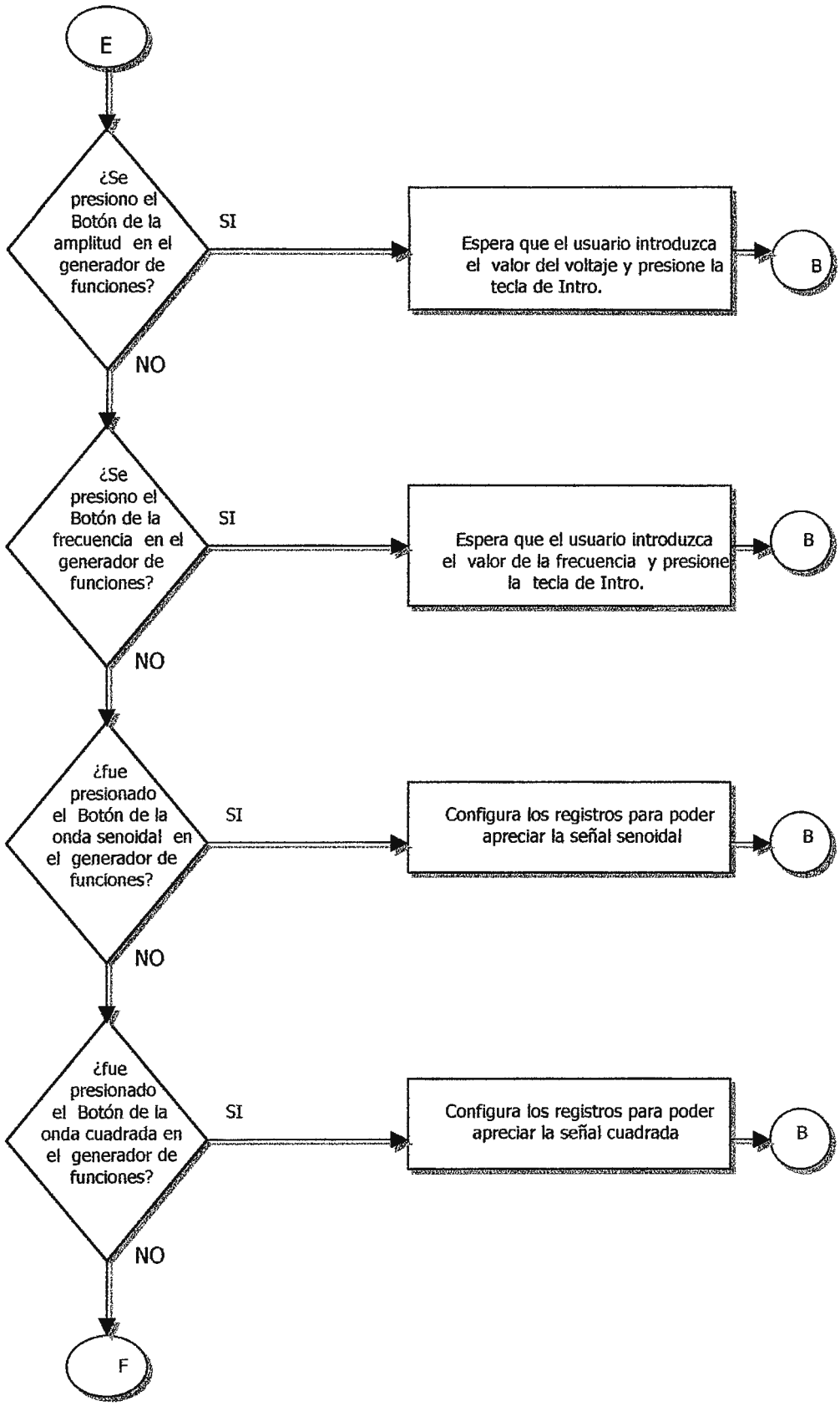
Dado el tamaño y complejidad del programa se presenta un Flujograma bastante general que da a entender la lógica seguida para poder lograr el funcionamiento de este proyecto.

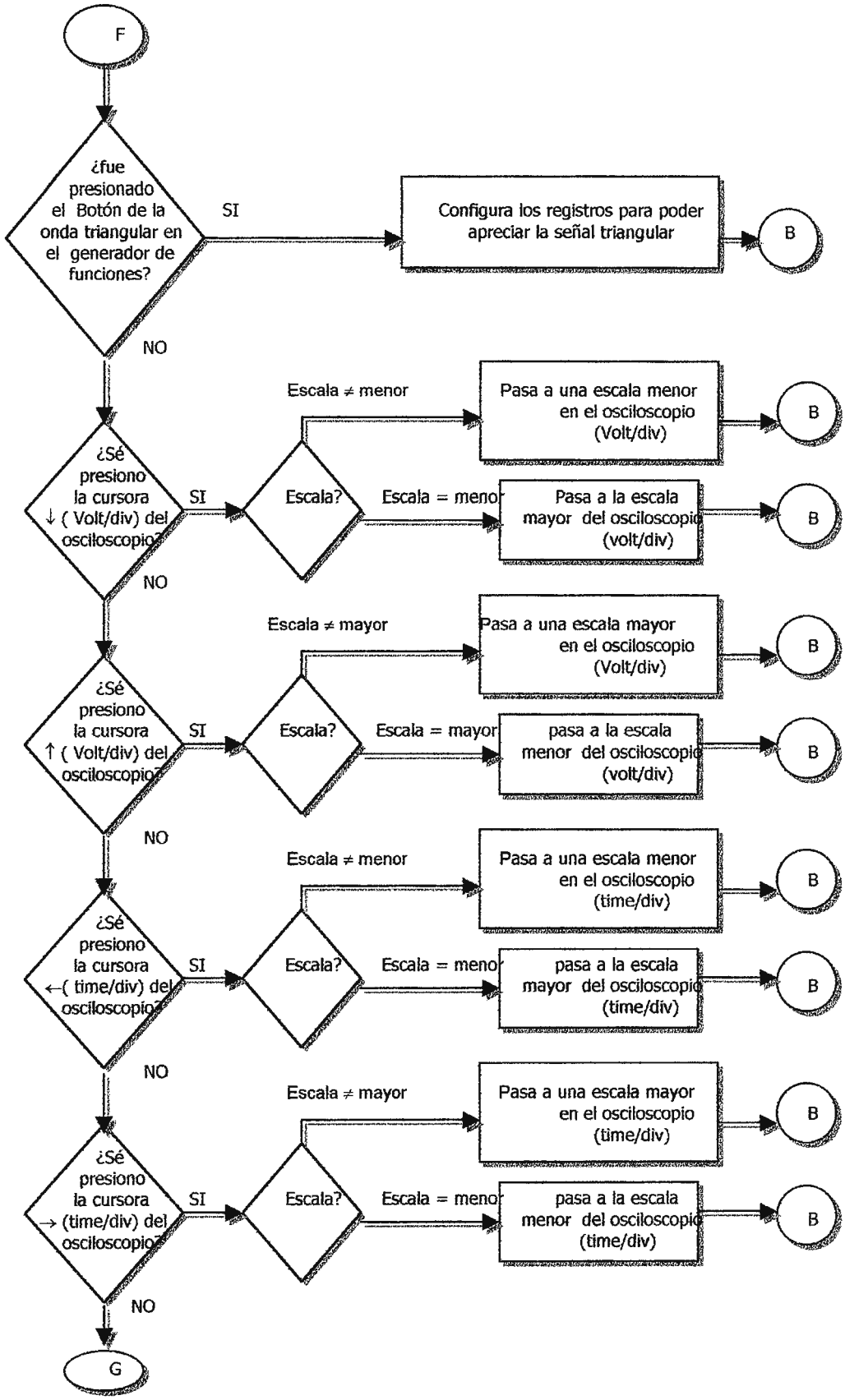
Flujograma general.

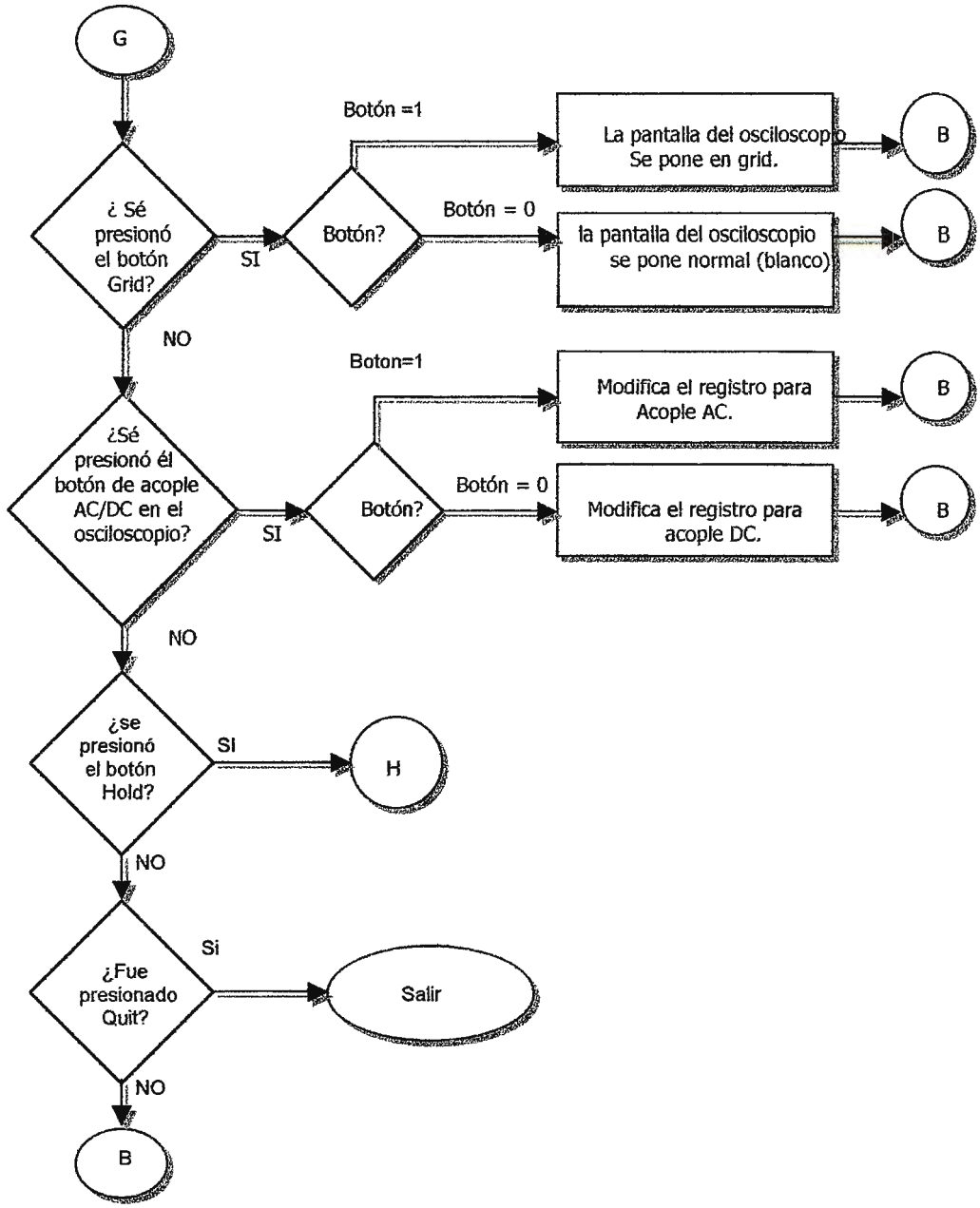


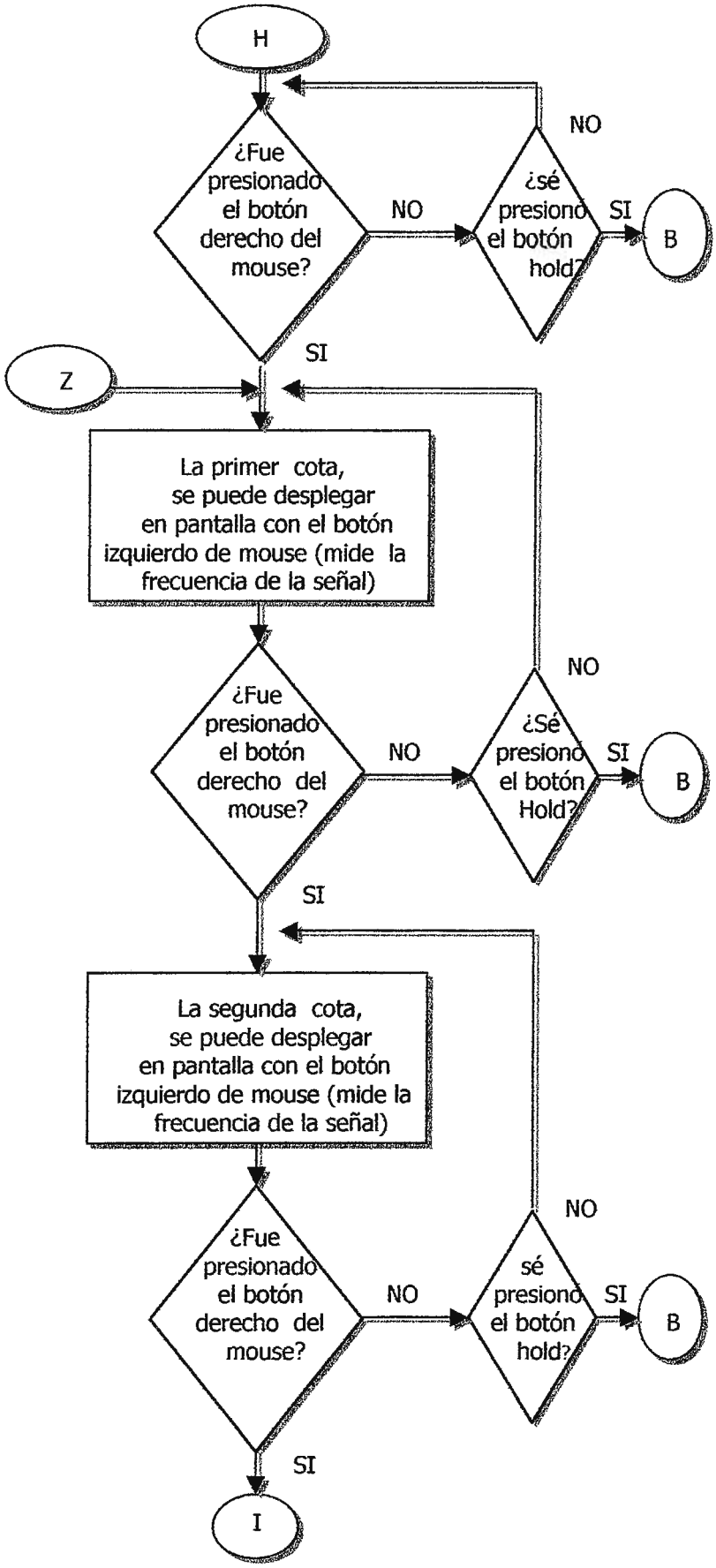

A
Presenta el valor medido por el multímetro
¿termino la adquisición de los datos?
NO
C
SI
Lee los datos de de la memoria y presenta la señal en pantalla
¿se presiono el botón de resistencia en el multimetro?
SI
configura los registros para poder utilizar el óhmetro
B
NO
¿Fue presionado el botón del amperímetro en el Multímetro?
SI
Configura los registros necesarios para poder utilizar el amperímetro.
B
NO
D

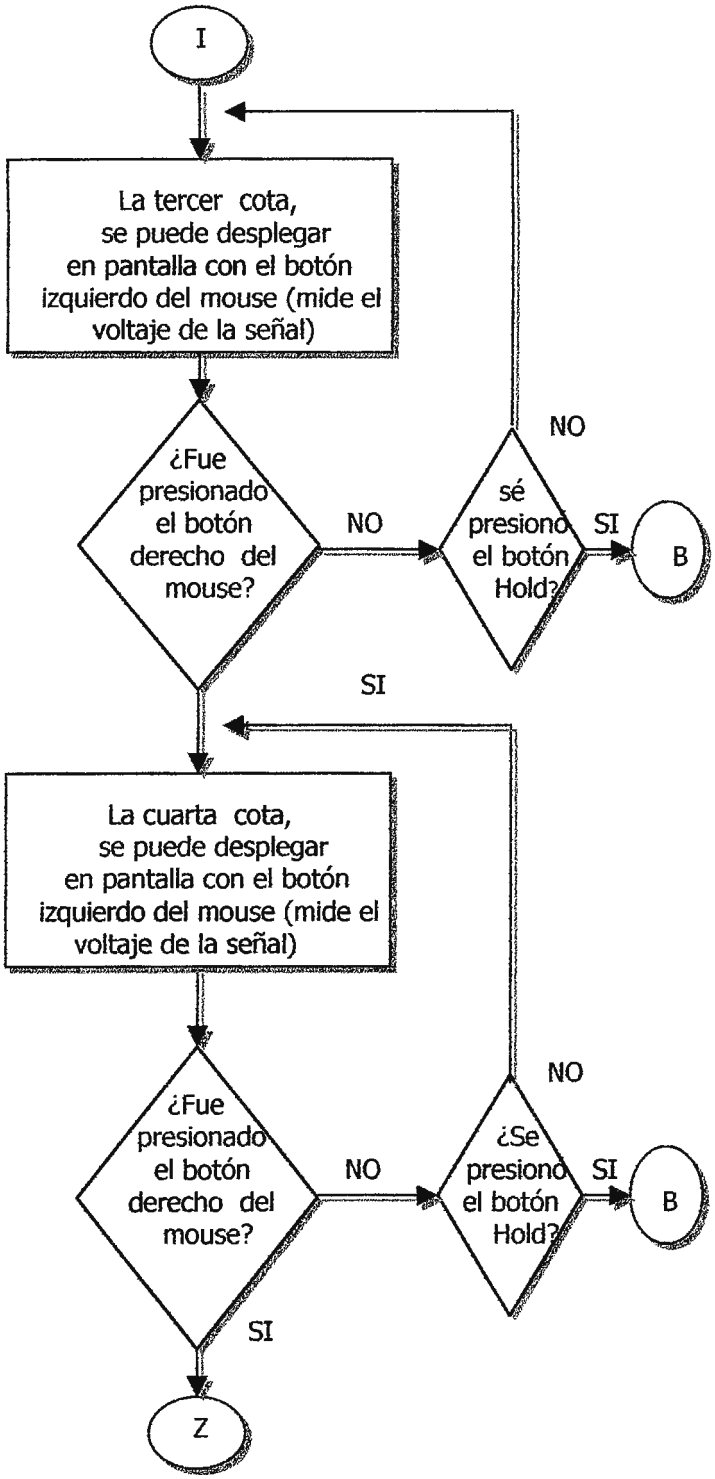

D
¿Fue presionado el botón del voltímetro en el multímetro?
SI
Configura los registros necesarios para poder utilizar el voltímetro
B
NO
¿ Sé presiono el botón acople AC/DC (multímetro)?
SI
Botón?
AC/DC=1
Boton AC/DC= AC Configura registro.
B
AC/DC=0
Botón AC/DC= DC configura registro.
B
NO
¿Sé presiono la cursora ← del Mult. (escala)?
SI
Escala?
Escala ≠ menor
pasa a una escala menor el multímetro
B
escala = menor
Pasa a la escala mayor el multímetro
B
NO
¿Sé presiono la cursora → del Mult. (escala)?
SI
Escala?
Escala ≠ mayor
Pasa a una escala mayor en el multímetro
B
escala = mayor
Pasa a la escala menor del multímetro
B
NO
¿Se Presiono el Botón de la fuente?
SI
Espera que el usuario introduzca el valor del voltaje y presione la tecla de intro.
B
NO
E


E
¿Se presiono el Botón de la amplitud en el generador de funciones?
SI
Espera que el usuario introduzca el valor del voltaje y presione la tecla de Intro.
B
NO
¿Se presiono el Botón de la frecuencia en el generador de funciones?
SI
Espera que el usuario introduzca el valor de la frecuencia y presione la tecla de Intro.
B
NO
¿fue presionado el Botón de la onda senoidal en el generador de funciones?
SI
Configura los registros para poder apreciar la señal senoidal
B
NO
¿fue presionado el Botón de la onda cuadrada en el generador de funciones?
SI
Configura los registros para poder apreciar la señal cuadrada
B
NO
F


F
¿fue presionado el Botón de la onda triangular en el generador de funciones?
SI
Configura los registros para poder apreciar la señal triangular
B
NO
¿Sé presiono la cursora ↓ (Volt/div) del osciloscopio?
SI
Escala?
Escala ≠ menor
Pasa a una escala menor en el osciloscopio (Volt/div)
Escala = menor
Pasa a la escala mayor del osciloscopio (volt/div)
NO
¿Sé presiono la cursora ↑ (Volt/div) del osciloscopio?
SI
Escala?
Escala ≠ mayor
Pasa a una escala mayor en el osciloscopio (Volt/div)
Escala = mayor
pasa a la escala menor del osciloscopio (volt/div)
NO
¿Sé presiono la cursora ←(time/div) del osciloscopio?
SI
Escala?
Escala ≠ menor
Pasa a una escala menor en el osciloscopio (time/div)
Escala = menor
pasa a la escala mayor del osciloscopio (time/div)
NO
¿Sé presiono la cursora → (time/div) del osciloscopio?
SI
Escala?
Escala ≠ mayor
Pasa a una escala mayor en el osciloscopio (time/div)
Escala = menor
pasa a la escala menor del osciloscopio (time/div)
NO
G


G
¿ Sé presionó el botón Grid?
SI
Botón?
Botón =1
La pantalla del osciloscopio Se pone en grid.
B
Botón = 0
la pantalla del osciloscopio se pone normal (blanco)
B
NO
¿Sé presionó él botón de acople AC/DC en el osciloscopio?
SI
Botón?
Boton=1
Modifica el registro para Acople AC.
B
Botón = 0
Modifica el registro para acople DC.
B
NO
¿se presionó el botón Hold?
SI
H
NO
¿Fue presionado Quit?
Si
Salir
NO
B


H
¿Fue presionado el botón derecho del mouse?
NO
¿sé presionó el botón hold?
NO
SI
B
SI
Z
La primer cota, se puede desplegar en pantalla con el botón izquierdo de mouse (mide la frecuencia de la señal)
¿Fue presionado el botón derecho del mouse?
NO
¿Sé presionó el botón Hold?
NO
SI
B
SI
La segunda cota, se puede desplegar en pantalla con el botón izquierdo de mouse (mide la frecuencia de la señal)
¿Fue presionado el botón derecho del mouse?
NO
sé presionó el botón hold?
NO
SI
B
SI
I


I
La tercer cota, se puede desplegar en pantalla con el botón izquierdo del mouse (mide el voltaje de la señal)
¿Fue presionado el botón derecho del mouse?
NO
sé presionó el botón Hold?
NO
SI
B
SI
La cuarta cota, se puede desplegar en pantalla con el botón izquierdo del mouse (mide el voltaje de la señal)
¿Fue presionado el botón derecho del mouse?
NO
¿Se presionó el botón Hold?
NO
SI
B
SI
Z

Programa de la interfaz gráfica.

```
# include <graphics.h>
# include <stdlib.h>
# include <ctype.h>
# include <math.h>
# include <stdio.h>
# include <conio.h>
# include <dos.h>
# include <bios.h>
# define COTA1 3        //COLOR DE LA COTA1
# define COTA2 9        //COLOR DE LA COTA2
# define COTA3 10       //COLOR DE LA COTA3
# define COTA4 11       //COLOR DE LA COTA4
# define IZQ 546       //POSICIÓN X DE LOS VALORES A IMPRIMIR
                         (VOLTAJE,TIEMPO  Y FRECUENCIA)
# define ARRIB 120      //POSICIÓN Y DE LOS VALORES A IMPRIMIR (VOLTAJE)
# define ENMEDIO 140    //POSICIÓN Y DE LOS VALORES A IMPRIMIR (TIEMPO)
# define ABAJO 160      //POSICIÓN Y DE LOS VALORES A IMPRIMIR (FRECUENCIA)
# define COLO 11        //COLOR DE LAS LETRAS
# define FOND 00        //COLOR DE FONDO
# define TAMA 1        //TAMAÑO DE LAS LETRAS
# define LETRA 0        //TIPO DE LETRA
# define COLO1 0        //color de las letras (escalas) del multímetro
# define COLOF 0        //COLOR DE LAS LETRAS DE LA FUENTE
# define GRAF 05
# define DY 60
# define DX 205
# define DY1 50
# define DX1 460
 main()
{

  float time=0,der=0,time1=0,der1=0,escal,escal1,inter=0,pasa=0;
  float tiempo=0,dere=0,Rresolucion;
  char input[80],entrada1[80],inpu[80],entradaf[80],entradaf1[80];
  char *hey[ ]={"µs","ms"};
  char *hey1[ ]={"V","mV"};
  char *hey2[ ]={"Hz","kHz","MHz"};
  char *Iesc1[ ]={"*","Amp","mA","mA","mA","æA"};
  char *Iesc[ ]={"*","2.0Amp","500mA","50.0mA","5.00mA","500æA"};
  char *Vesc[ ]={"*","500V","50.0V","5.00V","500mV"};
  char *Vesc1[ ]={"*","V","V","V","mV"};
  char *Resc[ ]={"*","500.00","5.000K","50.00k","500.0k","5.000M"};
  char *Resc1[ ]={"*"," OHM","K OHM","k OHM","K OHM","M OHM"};
  int esctim=0x59,escvol=0x07,uni=0,uni1=0,uni2=0,uni3=0,uni4=0,change=0;
  int d=0, coorx ,coory ,rey ,color ,sol ,tercel ,n[240],memoria=99 ,col,una=0;
  int e=0,coorx1,coory1,rey1,color1,sol1,tercel1,k[240],memoria1=99,col1;
  int i=0,coorx2,coory2,rey2,color2,sol2,tercel2,s[520],memoria2=342,col2;
  int l=0,coorx3,coory3,rey3,color3,sol3,tercel3,w[520],memoria3=342,col3;
```

```
int cont=0, salir,escala=0x03,escala1=0x06,variable=0x5e,R=0,I=0,V=0,ret;
int senal=1,pc5,mc4066=0x10,Rescala=5,Iescala=5,Iescala1=5,Vescala=1,
                RESC=0x30;
float dato, y, adctest=0, adctest1=1, Iresolucion,Vresolucion,factor=0;
char amplitud[80];
char longitud[80];
char*timediv[ ]={"0.5µs/div","1µs/div","5µs/div","10µs/div","50µs/div","0.1ms/div"
                    ,"0.5ms/div","1ms/div","5ms/div","10ms/div","50ms/div","100ms/div"};
char *voltdiv[]={"10V/div","5V/div","2V/div","1V/div","0.5V/div"
                              ,"0.2V/div","0.1V/div","50mV/div"};
int linea,x,c=0,count,posx=100,posy,cambio=0,ac=0,Rescala1=5,pc1=0x0;
int driver = DETECT, mode, hold=0,xb,yb,ab,bb,cb,colorb,lb,lyb,mb, bbg,
                ybg,bbg1,ybg1,botg1,botong1;
int RAM,pc4=0x08,add,atos,m[256],pxb,pyb,pxbg,pybg,pxb1,pyb1,boton,bot,
                grid=0,b[10],pulsador1=0;
int pulsador2=0,pulsador3=0,conamp=3,conamp1=3,conlon=6,conlon1=6,
                del=0x00,botong,botg,pxbg1,pybg1;
int key, modifiers,tec=0,veces=0,punto=0,veces1=0,prue=0,xbt,bbt,botont,bott;
int cero=0,mismo=0,diferente=0,seltest=0x1,signo=0,acopac,posac=0,negac=0;
int primero1,segundo1,tercero1,cuarto1,quinto1,sexto1,septimo1,
                          octavo1,noveno1,decimo1;
float primero,segundo,tercero,cuarto,quinto,sexto,septimo,octavo,rms=0;
float decimal,numero,ampligen,ampligen1,adctestmax=0, adctestmax1=0,
                RE=0,Vescala1=1;
float fuentev,primerf,segundof,tercerf,Nciclos,Fmedida=0,Fmedida1=0x0;
int pc2=0x00,intermedio1f,intermedio2f,intermedio3f,unidf=0,unidf1=0;
int onda=3, Y5w, F1, F2,genn;
float genf, resl, genc;
char key2[80], key3[80], key4[80];
union REGS regs;

/* initialize graphics and local variables */
initgraph(&driver, &mode, "");
cleardevice ();
outp(0xf3f0,variable); //Selección de escala time/div.
outp(0xf3fe,escala);  //Selección vol/div (1V/div y acople DC(0.5ms) al inicio
outp(0xf3f2,RESC);
outp(0xf3f1,0x80);
outp(0xf3f1,seltest); //escala y Selección del V R I
outp(0xf3f9, 0x00); //quita la protección de la fuente
outp(0xf3fa, 0x00);
outp(0xf3fe, 0x00); //amplitud de la señal del generador de funciones
outp(0xf3fd, 0x20);//escritura y cargar en el dac
outp(0xf3fd, 0xa0);
setfillstyle(1,7);
bar (0,0,640,480);
setcolor(0);
ellipse(76,50,90,270,50,80);
ellipse(565,50,270,90,50,80);
ellipse(564,50,270,90,50,80);
setcolor(0);
```

```
line(36,1,604,1);
line(36,96,604,96);
setcolor(15);
line(36,0,604,0);
ellipse(75,50,90,270,50,80);
ellipse(74,50,90,270,50,80);
ellipse(566,50,270,90,50,80);

setfillstyle(1,1);   /*textura de las opciones */
floodfill(150,20,0);
setcolor(8);
line (0,97,640,97);
line (0,343,640,343);

setfillstyle(9,9);   /*textura de las opciones */
floodfill(1,101,8);

/*ventana del tester */
setcolor(0);
rectangle(65,360,295,460);
rectangle(335,360,565,460);
rectangle(65,360,295,375);
rectangle(335,360,565,375);
setfillstyle(1,9);
floodfill(66,361,0);
floodfill(366,361,0);
setfillstyle(1,15);
floodfill(75,380,0);
setfillstyle(9,9);
floodfill(440,400,0);
setcolor(0);

outtextxy(106,366,"Multímetro Digital");
outtextxy(370,366,"Selección de escala");
setcolor(15);
rectangle(336,376,564,459);
outtextxy(105,365,"Multímetro Digital");
outtextxy(369,365,"Selección de escala");
setcolor(7);
line(66,376,294,376);
line(66,377,293,377);
line(66,377,66,459);
line(67,377,67,459);
setcolor(8);
line(66,459,294,459);
line(67,458,294,458);
line(294,376,294,459);
line(293,377,293,459);


//sombra de las opciones izquierda
```

```
setcolor(15);
line (0,98,95,98);
line (0,97,96,97);
line (0,98,0,343);
line (1,98,1,342);
setcolor(8);
line (96,97,96,343);
line (95,98,95,343);
line (0,343,95,343);
line (1,342,95,342);

//sombra de las opciones derecha
setcolor(15);
line (524,97,640,97);
line (524,98,638,98);
line (524,98,524,342);
line (525,98,525,343);
setcolor(8);
line (639,98,639,343);
line (638,98,638,343);
line (524,343,639,343);
line (525,342,639,342);

/*CUADRO DE MENSAJES*/
setfillstyle(1,0);
bar(533,106,630,210);
setcolor(0);
rectangle(532,105,631,212);
setcolor(15);
line(533,106,630,106);
line(533,106,533,210);
setcolor(8);
line(533,211,630,211);
line(534,210,630,210);
line(630,211,630,106);
line(629,211,629,107);

setcolor(10);
sprintf (amplitud, "%s", voltdiv [conamp]);
outtextxy(IZQ,180,amplitud);
sprintf(longitud,"%s",timediv[conlon]);
outtextxy(IZQ,195,longitud);


 //coloca el fondo blanco en la pantalla de trab.
setfillstyle(1,7);          //borde de color gris claro en la pantalla de trab.
bar (97,97,523,343);
setcolor(0);                //linea blanca en el borde de la pantalla de trab.
rectangle (97,96,523,343);

//sombra de la pantalla del scope
```

```
setcolor(15);
line (97,343,523,343);

 /*titulo del programa */

setcolor(0);
line(200,2,205,12);
line(435,2,430,12);
line(205,12,430,12);

setfillstyle(1,9);
floodfill(206,8,0);
setcolor(15);
outtextxy(210, 3, "Equipo auxiliar de medición ");
setcolor(0);
rectangle(38,4,197,10);
rectangle(438,4,601,10);
setfillstyle(1,11);
floodfill(39,6,0);
floodfill(600,6,0);

/*CUADRO DE MENSAJES*/
setfillstyle(1,0);
bar(115,20,290,85);
setcolor(0);
rectangle(114,19,291,86);
setcolor(15);
line(115,20,290,20);
line(115,20,115,85);
setcolor(8);
line(115,86,290,86);
line(116,85,290,85);
line(290,86,290,21);
line(289,86,289,22);


 /*Frecuencímetro */

//lineas de separación:
setcolor(7);
line(370,17,370,88);
setcolor(8);
line(369,15,369,90);
line(368,17,368,88);


setfillstyle(1,15);
bar(390,20,550,50);
setcolor(0);
rectangle(389,19,551,51);
```

```
setcolor(7);
line(390,20,550,20);
line(390,21,549,21);
line(390,20,390,50);
line(391,20,391,49);
setcolor(8);
line(390,51,550,51);
line(391,50,550,50);
line(550,20,550,51);
line(549,21,549,50);

/* Fuente */
setfillstyle(1,15);
bar(390,60,550,90);
setcolor(0);
line(98,343,522,343);
rectangle(389,59,551,91);
setcolor(7);
line(390,60,550,60);
line(390,61,549,61);
line(390,60,390,90);
line(391,60,391,89);
setcolor(8);
line(390,91,550,91);
line(391,90,550,90);
line(550,60,550,91);
line(549,61,549,90);


inicio:
settextstyle (1,0,1);
/*botones para cambio de escala */
setcolor(15);
ellipse(450,433,0,360,75,15);
setfillstyle(1,15);
bar(409,418,492,448);
setcolor(COLO1);
if (I==1)
outtextxy (430,430,Iesc[ Iescala ]);
if(R==1)
outtextxy (430,430,Resc[ Rescala ]);
if(V==1)
outtextxy (430,430,Vesc[Vescala]);
setfillstyle(1,7);
floodfill(405,422,15);
floodfill(495,422,15);
setcolor(15);
ellipse(450,433,120,180,74,14); //luz blanca del botón
ellipse(450,433,120,180,73,13);
ellipse(450,433,270,120,74,14);
ellipse(450,433,270,120,73,13);
```

```
setcolor(8);
ellipse(450,433,180,236,74,14); //sombra del botón
ellipse(450,433,180,236,73,13);
ellipse(450,433,180,236,75,15);
ellipse(450,433,305,360,75,15);
ellipse(450,433,305,360,74,14);
ellipse(450,433,305,360,73,13);
setcolor(15);
outtextxy(395,430,"<");
outtextxy(500,430,">");
setcolor(0);
// rectangle(375,420,410,448); coordenada del botón <
outtextxy(394,429,"<");
// rectangle(490,420,525,448); coordenada del botón >
outtextxy(501,429,">");
rectangle(409,418,492,448);
setcolor(8);
rectangle(410,419,491,447);
rectangle(411,420,490,446);
setcolor(7);
line(491,419,491,447);
line(490,420,490,447);
line(410,447,491,447);
line(411,446,491,446);

/*BOTÓNES DEL TESTER*/
for(xbt=0;xbt<=3;xbt++)
        { bbt= xbt * 40;
          setcolor(0);
          rectangle(370+bbt,385,410+bbt,405);    //cordenadas de los botones  R I V
          setfillstyle(1,7);
          bar(371+bbt,386,409+bbt,404);
          setcolor(15);
          rectangle(371+bbt,386,409+bbt,404);
          rectangle(372+bbt,387,408+bbt,403);
          setcolor(8);
          line(371+bbt,404,409+bbt,404);
          line(372+bbt,403,408+bbt,403);
          line(409+bbt,404,409+bbt,387);
          line(408+bbt,403,408+bbt,388);
        }

for(cb=1;cb<=2;cb++)
        {
          colorb=-cb*15+30;
          settextstyle(12,0,1);
          setcolor(colorb);
          outtextxy(389-cb,394-cb,"R");
          outtextxy(429-cb,394-cb,"I");
          outtextxy(469-cb,394-cb,"V");
          outtextxy(504-cb,394-cb,"DC");
```

```
        }

if(R==1)
      {
          lescala=5;
          lescala1=5;
          seltest = seltest & 0xe0;        //11100000
          seltest = seltest | 0x10;       //00010000
          outp(0xf3f1, seltest);
          seltest = seltest | 0x60;        //01100000
          outp(0xf3f1,seltest);
          RESC=RESC |  0x20;              //00100000
          outp(0xf3f2, RESC);
          mc4066 = mc4066 | 0x2;           //00000010
          outp(0xf3f3,mc4066);
          mc4066 = mc4066 | 0x04;             //00000100
          outp(0xf3f3,mc4066);
          setcolor(7);
          outtextxy(388,393,"R");
          setcolor(11);
          outtextxy(387,392 ,"R");
      }

if(I==1)
      {
          mc4066= mc4066 & 0xf5;           //11110101
          outp(0xf3f3, mc4066);
          seltest = seltest & 0x9f;        //10011111
          seltest = seltest | 0x40;             //01000000
          outp(0xf3f1, seltest);
          delay(10);
          RESC= RESC &  0xdf;              //11011111
          outp(0xf3f2,RESC);
          mc4066 = mc4066 | 0x1;           //00000001
          outp(0xf3f3,mc4066);
          mc4066 = mc4066 | 0x04;             //00000100
          outp(0xf3f3,mc4066);
          setcolor(7);
          outtextxy(428,393,"I");
          setcolor(11);
          outtextxy(427,392 ,"I");
        }
if(V==1)
        {
          lescala=5;
          lescala1=5;
          seltest = seltest & 0xe0;        //11100000
          seltest = seltest | 0x10;       //00010000
          outp(0xf3f1, seltest);
          seltest= seltest & 0xbf;         //10111111
```

```
            outp(0xf3f1,seltest);
            RESC=RESC | 0x20;              //00100000
            outp(0xf3f2, RESC);
            mc4066 = mc4066 & 0xfb;              //11111011
            outp(0xf3f3,mc4066);
            mc4066 = mc4066 | 0x08;              //00001000
            outp(0xf3f3,mc4066);
            setcolor(7);
            outtextxy(468,393,"V");
            setcolor(11);
            outtextxy(467,392 ,"V");
        }
if(ac==1)
        {
            setcolor(7);
            outtextxy(504,394,"DC");
            outtextxy(503,393,"DC");
            outtextxy(502,392,"DC");
            setcolor(11);
            outtextxy(502,392,"AC");
        }
/*BOTONES GEN  FUN*/
for(ybg=0;ybg<=2;ybg++)
        {
            bbg= ybg * 25;
            setcolor(0);
            rectangle(50,18+bbg,100,38+bbg);
            setfillstyle(1,7);
            bar(51,19+bbg,99,37+bbg);
            setcolor(15);
            rectangle(51,19+bbg,99,37+bbg);
            rectangle(52,20+bbg,98,36+bbg);
            setcolor(8);

           /*SOMBRA DEL BOTÓN*/
           line(51,37+bbg,99,37+bbg);
           line(52,36+bbg,98,36+bbg);
           line(99,37+bbg,99,19+bbg);
           line(98,36+bbg,98,20+bbg);
        }

/*Botones de tipo de señal */

for(ybg1=0;ybg1<=2;ybg1++)
        {
                bbg1= ybg1 * 25;
                setcolor(0);
                rectangle(305,18+bbg1,355,38+bbg1);
                setfillstyle(1,7);
                bar(306,19+bbg1,354,37+bbg1);
                setcolor(15);
```

```
            rectangle(306,19+bbg1,354,37+bbg1);
            rectangle(307,20+bbg1,353,36+bbg1);
            setcolor(8);

            /*SOMBRA DEL BOTÓN*/
            line(306,37+bbg1,354,37+bbg1);
            line(307,36+bbg1,353,36+bbg1);
            line(354,37+bbg1,354,19+bbg1);
            line(353,36+bbg1,353,20+bbg1);

      }

for(cb=1;cb<=2;cb++)
      {
            colorb=-cb*15+30;
            settextstyle(12,0,1);
            setcolor(colorb);
            outtextxy(316-cb,26-cb,"Seno");
            outtextxy(316-cb,52-cb,"Cuad");
            outtextxy(316-cb,77-cb,"Trig");

      }
//tipo de señal
setfillstyle(7,8);
bar(235,30,285,75);
setcolor(15);
rectangle(235,30,285,75);
if(senal ==1)
      {
            setcolor(14);      //seno
            ellipse(250,53,3,180,10,13);
            ellipse(270,53,177,360,10,13);
      }
if(senal ==2)
      {
            setcolor(14);      //cuadrada
            line(240,65,240,40);
            line(240,40,260,40);
            line(260,40,260,65);
            line(260,65,280,65);
            line(280,65,280,40);
      }
if(senal==3)
      {
            setcolor(14);
            line(240,53,250,40);
            line(250,40,270,65);
            line(270,65,280,53);
      }
for(cb=1;cb<=2;cb++)
      {
```

```
            colorb=-cb*15+30;
            settextstyle(12,0,1);
            setcolor(colorb);
            outtextxy(58-cb,26-cb,"Gen-F");
            outtextxy(58-cb,52-cb,"Amplt");
            outtextxy(60-cb,77-cb,"Frec");
        }
/*Botón para Selección de la fuente */
setcolor(0);
rectangle(560,20,600,50);
setfillstyle(1,7);
bar(561,21,599,49);
setcolor(15);
rectangle(561,21,599,49);
rectangle(562,22,598,48);
setcolor(0);
outtextxy(565,31,"Volt");
setcolor(8);

/*SOMBRA DEL BOTON*/
line(561,49,599,49);
line(562,48,599,48);
line(599,49,599,21);
line(598,48,598,22);

/*Botón para Selección del F */
setcolor(0);
rectangle(560,60,600,90);
setfillstyle(1,7);
bar(561,61,599,89);
setcolor(15);
rectangle(561,61,599,89);
rectangle(562,62,598,88);
setcolor(0);
outtextxy(573,72,"FC");
setcolor(8);

/*SOMBRA DEL BOTON*/
line(561,89,599,89);
line(562,88,599,88);
line(599,89,599,61);
line(598,88,598,62);

/* cursoras */
setfillstyle(7,8);
fillellipse (580,277,27,42);
setcolor(15);
ellipse(580,277,0,360,28,43);
setcolor(0);
rectangle(572,310,588,330);
rectangle(572,224,588,244);
```

```
 rectangle(542,269,562,285);
 rectangle(598,269,618,285);
 setfillstyle(1,7);
 bar(573,311,587,329);
 bar(573,225,587,243);
 bar(543,270,561,284);
 bar(599,270,617,284);
 setcolor(15);
 rectangle(573,311,587,329);
 rectangle(573,225,587,243);
 rectangle(543,270,561,284);
 rectangle(599,270,617,284);
 setcolor(8);
 line(573,329,587,329);
 line(573,243,587,243);
 line(543,284,561,284);
 line(599,284,617,284);
 line(587,329,587,311);
 line(587,243,587,225);
 line(561,284,561,270);
 line(617,284,617,270);
 setcolor(0);
 outtextxy(549,274,"<");
 outtextxy(605,274,">");
 settextstyle (0,1,1);
 outtextxy(585,315,"<");
 outtextxy(585,230,">");
 settextstyle (2,0,6);
 setcolor(15);
 outtextxy(576,242+5,"s");
 outtextxy(576,253+5,"c");
 outtextxy(576,265+5,"o");
 outtextxy(576,277+5,"p");
 outtextxy(576,292+5,"e");

 /*BOTONES oscope */
for (yb=0;yb<=3;yb++)
      {
            bb= yb * 60;
            setcolor(0);
            rectangle(15,110+bb,65,140+bb);
            setfillstyle(1,7);
            bar(16,111+bb,64,139+bb);
            setcolor(15);
            rectangle(16,111+bb,64,139+bb);
            rectangle(17,112+bb,63,138+bb);
            setcolor(8);

            /*SOMBRA DEL BOTÓN*/
            line(16,139+bb,64,139+bb);
            line(17,138+bb,63,138+bb);
```

```
            line(64,139+bb,64,111+bb);
            line(63,138+bb,63,112+bb);
        }

/*NOMBRES DE LOS BOTONES*/
for(cb=1;cb<=2;cb++)
        {
            colorb=-cb*15+30;
            settextstyle(12,0,1);
            setcolor(colorb);
            outtextxy(26-cb,125-cb,"Grid");
            outtextxy(23-cb,185-cb,"AC/DC");
            outtextxy(26-cb,245-cb,"Hold");
            outtextxy(26-cb,305-cb,"Quit");
        }

/*ON/OFF de los leds */
for(lb=0;lb<=2;lb++)
        {
            lyb= lb * 60;
             setcolor(8);
            circle(80,125+lyb,5);
            setfillstyle(1,10);
            if(b[lb+1] != lb+1)
                {
                    floodfill(83,123+lyb,8);
                    putpixel(78,124+lyb,15);
                }
        }

setfillstyle(1,15);
if(grid==0)                     //coloca el fondo blanco en la pantalla de trab.
        bar (101,101,519,339);

salto:
regs.x.ax=0x1;
int86(0x33,&regs,&regs);
mb=0;
while(mb==0)                            /*ADQUISICION DE DATOS*/
        {
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
            pxb=regs.x.cx;
            pyb=regs.x.dx;
            pxbg=regs.x.cx;
            pybg=regs.x.dx;
            pxbg1=regs.x.cx;
            pybg1=regs.x.dx;
            outp(0xf3f0,variable);
            rep:
```

```
regs.x.ax=0x1;
int86(0x33,&regs,&regs);
regs.x.ax=0x3;
int86(0x33,&regs,&regs);
del = 0x1 & regs.x.bx;
pxb1=regs.x.cx;
pyb1=regs.x.dx;
 if(del == 0x01)
        {
                mb=del;
                pxb=pxb1;
                pyb=pyb1;
        }
 pc1=inp(0xf3f8);
 pc1= pc1 & 0x1;
/* frecuencímetro */
delay(10);
outp(0xf3fa, 0x01);
pc2=inp(0xf3f8);
pc2=pc2 & 0x2;

if ( pc2 ==0x2)
        {
                //pc2=0x00;
                intermedio1f=inp(0xf3f9);
                intermedio2f=inp(0xf3fa);
                intermedio3f=inp(0xf3fb);
                outp(0xf3fa, 0x00);//se limpia para poner en impedancia los
                                buffer y así se evitan conflictos con
                        otra información
                primerf=intermedio1f;
                segundof=intermedio2f;
                segundof= segundof* 0x100;
                tercerf= intermedio3f;
                tercerf= tercerf*0x10000;
                Nciclos=primerf+segundof+tercerf;
                Fmedida=20000000/Nciclos;


                if (Fmedida >=1000)
                        {
                                Fmedida=Fmedida/1000;
                                unidf=1;
                        }
                else
                unidf=0;
                if (Fmedida != Fmedida1)
                        {
                                setcolor(15);
                sprintf(entradaf1,"%3.2f%s",Fmedida1,hey2[unidf1]);
                                outtextxy(IZQ-110,ARRIB-47,entradaf1);
```

```
                                setcolor(COLOF);
                        sprintf(entradaf,"%3.2f%s",Fmedida,hey2[unidf]);
                                settextstyle(LETRA, 0,TAMA);
                                outtextxy(IZQ-110,ARRIB-47,entradaf);
                                Fmedida1= Fmedida;
                                unidf1=unidf;
                            }
                        outp(0xf3fa, 0x00);
                        pc2=0x00;

                }

/*protección de la fuente */
if(pc1==0x1)
        {
                /*setfillstyle(1,0);
                bar(390,60,550,90);  */
                setfillstyle(1,15);
                bar(DX1 +85,DY1 - 25,DX1 -65,DY1 -2);
                settextstyle(0,0,1);
                setcolor(1);
                outtextxy(400,30,"corto en la fuente");
        }
pc5=inp(0xf3f8);
pc5=inp(0xf3f8);
pc5=pc5 & 0x10;
if(pc5==0x10)  // poner para utilizar el multímetro
        {
                sound(500);
                delay(40);
                sound(800);
                delay(40);
                if(una == 0)
                        {
                                settextstyle(0,0,2);
                                setfillstyle(1,15);
                                setcolor(15);
                                bar(70,380,270,450);
                        }
                una=1;
                setcolor(0);            //MULTIMETRO
                settextstyle(0,0,2);
                outtextxy(138,390,"Error");
                outtextxy(100,425,"Multímetro");
                if(pxb > 65 & pxb < 295 & pyb > 360 & pyb < 460)
                        {
                                una=0;
                                regs.x.ax =0x02;
                                int86 (0x33, &regs, &regs);
                                settextstyle(0,0,2);
                                setcolor(15);
```

```
                        outtextxy(138,390,"Error");
                        outtextxy(100,425,"Multímetro");
                        settextstyle(1,0,1);
                        seltest = seltest & 0xbf;     // 10111111
                        outp(0xf3f1,seltest);
                        seltest = seltest | 0xc0;     //11000000
                        outp(0xf3f1,seltest);
                        seltest = seltest & 0x7f;     //01111111
                        outp(0xf3f1,seltest);
                        regs.x.ax =0x01;
                        int86 (0x33, &regs, &regs);
                        adctest1=1;
                }
            else
            goto rep;
        }
nosound();
if(l==1)
        {
            if (lescala==1)
                {
                        factor=1;
                        lresolucion=7.843137255e-03;
                }
            if(lescala==2)
                {
                        lresolucion=1.960784314e-03;
                        factor=1000;
                }
            if (lescala==3)
                {
                        factor=1000;
                        lresolucion=196.0784314e-06;
                }
            if (lescala==4)
                {
                        factor=1000;
                        lresolucion=19.60784314e-06;
                }
            if (lescala==5)
                {
                        factor=1000000;
                        lresolucion=1.950784314e-06;
                }
            if(ac==0)
                {
                        posac=0;
                        negac=0;
                        for(acopac=0;acopac<=10;acopac+=1)
                                {
                                        for(ret=0;ret<=50e02;ret+=1)
```

```
                        outtextxy(138,390,"Error");
                        outtextxy(100,425,"Multímetro");
                        settextstyle(1,0,1);
                        seltest = seltest & 0xbf;     // 10111111
                        outp(0xf3f1,seltest);
                        seltest = seltest | 0xc0;     //11000000
                        outp(0xf3f1,seltest);
                        seltest = seltest & 0x7f;     //01111111
                        outp(0xf3f1,seltest);
                        regs.x.ax =0x01;
                        int86 (0x33, &regs, &regs);
                        adctest1=1;
                }
            else
            goto rep;
        }
nosound();
if(l==1)
        {
            if (lescala==1)
                {
                        factor=1;
                        lresolucion=7.843137255e-03;
                }
            if(lescala==2)
                {
                        lresolucion=1.960784314e-03;
                        factor=1000;
                }
            if (lescala==3)
                {
                        factor=1000;
                        lresolucion=196.0784314e-06;
                }
            if (lescala==4)
                {
                        factor=1000;
                        lresolucion=19.60784314e-06;
                }
            if (lescala==5)
                {
                        factor=1000000;
                        lresolucion=1.950784314e-06;
                }
            if(ac==0)
                {
                        posac=0;
                        negac=0;
                        for(acopac=0;acopac<=10;acopac+=1)
                                {
                                        for(ret=0;ret<=50e02;ret+=1)
```

```
                                        {
                                        }
                                    // adctest= inp(0xf3fd);
                                     signo=inp(0xf3f8);
                                     signo=signo & 0x20;
                                    if (signo==0x20)
                                            posac=1;
                                    if(signo==0x00)
                                    negac=1;
                                }
                if((posac==0 & negac==1 ) | (posac==1 & negac==0))
                                {
                                     if(posac==1)
                                            signo=1;
                                     else
                                            signo=-1;
                                     adctest= inp(0xf3fd);
                                     adctest= inp(xf3fd);
                        adctest=(adctest * lresolucion)*factor* signo;
                                }
                else
                        adctest=0x00;
                 delay(40);
                 if(adctest1!=adctest)
                        {
                                regs.x.ax =0x02;
                                int86 (0x33, &regs, &regs);
                                settextstyle(0,0,2);
                                setcolor(15);
                        sprintf(key4,"%4.2f%s", adctest1,lesc1[lescala1]);
                                outtextxy(120,410,key4);
                                setcolor(1);
                        sprintf(key4,"%4.2f%s", adctest,lesc1[lescala]);
                                outtextxy(120,410,key4);
                                adctest1=adctest;
                                lescala1=lescala;
                                regs.x.ax =0x01;
                                int86 (0x33, &regs, &regs);
                                settextstyle(0,0,1);
                        }

                }
if (ac==1)
    {
            posac=0;
            negac=0;
            rms=0;
            for(acopac=0;acopac<=50;acopac+=1)
                    {
                            for(ret=0;ret<=50e02;ret+=1)
                                    {
```

```
                        }
                    // adctest= inp(0xf3fd);
                    adctest= inp(0xf3fd);
                    if (adctest>=rms)
                        rms=adctest;
                        signo=inp(0xf3f8);
                        signo=signo & 0x20;
                    if (signo==0x20)
                        posac=1;
                    if(signo==0x00)
                        negac=1;
                }
            if(posac==1 & negac==1)
                {
                    adctestmax=rms;
                    adctestmax=(adctestmax * lresolucion)*factor;
                    adctest=rms/sqrt(2);
                    adctest=(adctest * lresolucion)*factor;
                }
        else
                {
                    adctest=0;
                    adctestmax=0;
                }


    }
delay(40);
if(adctest1!=adctest)
    {
        regs.x.ax =0x02;
        int86 (0x33, &regs, &regs);
        settextstyle(0,0,2);
        setcolor(15);
        sprintf(key4,"%4.2f%s RMS", adctest1,lesc1[lescala1]);
        outtextxy(90,390,key4);
        sprintf(key4,"%4.2f%s MAX", adctestmax1,lesc1[lescala1]);
        outtextxy(90,430,key4);
        setcolor(1);
        sprintf(key4,"%4.2f%s RMS", adctest,lesc1[lescala]);
        outtextxy(90,390,key4);
        sprintf(key4,"%4.2f%s MAX", adctestmax,lesc1[lescala]);
        outtextxy(90,430,key4);
        adctest1=adctest;
        lescala1=lescala;
        adctestmax1=adctestmax;
        regs.x.ax =0x01;
        int86 (0x33, &regs, &regs);
        settextstyle(0,0,1);
    }
```

```
        }

if(R==1)
        {
                f(Rescala==1)
                        {
                                RE=100;
                                factor=1;
                        }
                if(Rescala==2)
                        {
                                RE=1.022e03;
                                factor=1000;
                        }
                if(Rescala==3)
                        {
                                RE=9.9e03;
                                factor=1000;
                        }
                if(Rescala==4)
                        {
                                factor=1000;
                                RE=99.9e03;
                        }
                if(Rescala==5)
                        {
                                factor=1000000;
                                RE=1.027e06;
                        }

                adctest= inp(0xf3fd);
                adctest= inp(0xf3fd);
                if(adctest == 0x0)
                    {
                                sound(1500);

                    }
                if(adctest != 0x0)
                                nosound();

                adctest=(adctest*7.843177255e-03 );
                adctest=((RE * (adctest/0.48))/(5.0 - (adctest/0.48)))/factor;
                delay(40);
                if(adctest1!=adctest)
                        {
                                regs.x.ax =0x02;
                                int86 (0x33, &regs, &regs);
                                settextstyle(0,0,2);
```

```
                        setcolor(15);
                        sprintf(key4,"%3.2f%s", adctest1,Resc1[Rescala1]);
                        outtextxy(100,408,key4);
                        setcolor(1);
                        sprintf(key4,"%3.2f%s " adctest,Resc1[Rescala]);
                        outtextxy(100,408,key4);
                        adctest1=adctest;
                        Rescala1=Rescala;
                        regs.x.ax =0x01;
                        int86 (0x33, &regs, &regs);
                        settextstyle(0,0,1);
                }
        }
if(V==1)
        {
                if(Vescala==1)
                        {
                                Vresolucion=1.960784314;
                                factor=1;
                        }
                if(Vescala==2)
                        {
                                Vresolucion=0.196078431;
                                factor=1;
                        }
                if(Vescala==3)
                        {
                                Vresolucion=19.60784314e-03;
                                factor=1;
                        }
                if(Vescala==4)
                        {
                                Vresolucion=1.960784314e-03;
                                factor=0.001;
                        }


                if(ac==0)
                        {
                                posac=0;
                                negac=0;
                                for(acopac=0;acopac<=10;acopac+=1)
                                        {
                                                for(ret=0;ret<=50e02;ret+=1)
                                                        {
                                                        }
                                                // adctest= inp(0xf3fd);
                                                 signo=inp(0xf3f8);
                                                 signo=signo & 0x20;
                                                if (signo==0x20)
                                                        posac=1;
```

```
                        if(signo==0x00)
                                negac=1;
                }
            if((posac==0 & negac==1 ) | (posac==1 & negac==0))
                {
                        if(posac==1)
                                signo=1;
                        else
                                signo=-1;
                         adctest= inp(0xf3fd);
                         adctest= inp(0xf3fd);
                  adctest=((adctest * Vresolucion)/factor)* signo;
                }
            else
                adctest=0x00;
            delay(40);
            if(adctest1!=adctest)
                {
                        regs.x.ax =0x02;
                        int86 (0x33, &regs, &regs);
                        settextstyle(0,0,2);
                        setcolor(15);
                sprintf(key4,"%4.2f%s", adctest1,Vesc1[Vescala1]);
                       outtextxy(120,410,key4);
                        setcolor(1);
                sprintf(key4,"%4.2f%s", adctest,Vesc1[Vescala]);
                        outtextxy(120,410,key4);
                        adctest1=adctest;
                        Vescala1=Vescala;
                        regs.x.ax =0x01;
                        int86 (0x33, &regs, &regs);
                        settextstyle(0,0,1);
                }

        }

if (ac==1)
        {
            posac=0;
            negac=0;
            rms=0;
            for(acopac=0;acopac<=50;acopac+=1)
                {
                    for(ret=0;ret<=50e02;ret+=1)
                        {
                        }
                    adctest= inp(0xf3fd);
                    if (adctest>=rms)
                            rms=adctest;
                    signo=inp(0xf3f8);
                    signo=signo & 0x20;
```

```
                    if (signo==0x20)
                            posac=1;
                    if(signo==0x00)
                            negac=1;
                }
            if(posac==1 & negac==1)
                {
                        adctestmax=rms;
                        adctestmax=(adctestmax * Vresolucion)/factor;
                        adctest=rms/sqrt(2);
                        adctest=(adctest * Vresolucion)/factor;
                }
            else
                {
                        adctest=0;
                        adctestmax=0;
                }
            delay(40);
            if(adctest1!=adctest)
                {
                        regs.x.ax =0x02;
                        int86 (0x33, &regs, &regs);
                        settextstyle(0,0,2);
                        setcolor(15);
                    sprintf(key4,"%4.2f%s RMS", adctest1,Vesc1[Vescala1]);
                        outtextxy(90,390,key4);
                        sprintf(key4,"%4.2f%s MAX",
        adctestmax1,Vesc1[Vescala1]);
                        outtextxy(90,430,key4);
                        setcolor(1);
                    sprintf(key4,"%4.2f%s RMS", adctest,Vesc1[Vescala]);
                        outtextxy(90,390,key4);
                    sprintf(key4,"%4.2f%s MAX", adctestmax,Vesc1[Vescala]);
                        outtextxy(90,430,key4);
                        adctest1=adctest;
                        Vescala1=Vescala;
                        adctestmax1=adctestmax;
                        regs.x.ax =0x01;
                        int86 (0x33, &regs, &regs);
                        settextstyle(0,0,1);
                    }
        }

    }


pc4=inp(0xf3f8);
pc4=inp(0xf3f8);
// goto quit;       //quit
if(pc4 != 0x08)   //08
        pc4 = pc4 & 0x08;  //00000100
```

```
 if(pc4 == 0x08)
 goto rep;
 delay(1);
 regs.x.ax=0x2;
 int86(0x33,&regs,&regs);
 outp(0xf3f0,0xe7); //11100111  ////////////////////////
 RAM=0xe7;
 posx=100;
 for(add=0; add <= 160; add++)
        {
                atos=inp(0xf3fc);
                atos=inp(0xf3fc);
                m[add]=atos;
                RAM = RAM | 0x08;
                outp(0xf3f0, RAM);
                RAM = RAM & 0xf7;
                outp(0xf3f0, RAM);
        }

setfillstyle(1,15);
bar (100,100,520,340);
if(grid==1)
        {
                setfillstyle(1,12);
                floodfill(83,123,8);
                setcolor(7);
                for(linea=0;linea<=420;linea+=30)
                line (100+ linea,100,100 + linea,340);
                for (linea=0;linea<=240;linea+=30)
                line (100,100+ linea,520,100+ linea);           //coloca el fondo blanco en la
                                                                pantalla de trab.
        }
 for(add=0; add <= 2; add++)
          {
        y = 340 - 0.94117647 *  m[add+10];   //10*3*8=240; 240/255=0.94117647
                posy = y;               //punto anterior de la posición y
                for(add=10;add<=150;add++)          //lleva el conteo de cuantas veces he
                                                    leído la memoria
                        {
                        y = 340 - 0.94117647 * m[add];    //se saca la coordenada y
                                x = 100 +(add-10)*3;

                                if(y>=100 & y <= 340)   //gráfica únicamente si la
                                        coordenada y es mayor o igual de
                                        100 y menor o igual de 340
                                        {

                                        setcolor(GRAF) //se incrementa por simulacro
                                          line(posx, posy ,x, y);
                                          //putpixel(x,y,4);              //uno los puntos
                                            posx=x;             //actualizo la coordenada x
```

```
                                        posy=y;             //actualizo la coordenada y

                                    }
                            }
                        setcolor(15);
                        line(100,100,520,100);
                        line(100,340,520,340);
                }
regs.x.ax =0x01;
int86 (0x33, &regs, &regs);
delay(125);
regs.x.ax =0x02;
int86 (0x33, &regs, &regs);
}

regs.x.ax =0x01;
int86 (0x33, &regs, &regs);

  /*CUAL BOTON SÉ PRECIONO*/
  regs.x.ax=0x2;
  int86(0x33,&regs,&regs);
  boton=0;
  botong=0;
  botong1=0;
  botont=0;
  for(yb=0;yb<=3;yb++)
      {
          ybg=yb;
          ybg1=yb;
          xbt=b;
          boton+=1;
          botong+=1;
          botong1+=1;
          botont+=1;
          bb= yb * 60;
          bbg = ybg * 25;
          bbg1 = ybg1 * 25;
          bbt = xbt * 40;
          /*CUÁL TECLA SÉ PRECION?*/
          // botones del tester R I V?
         if( pxb > 370 + bbt & pxb < 410 +bbt & pyb > 385 & pyb < 405)
             {
                    if(botont < 5)
                            {

                                    bott=botont;
                                    setcolor(8);
                                    rectangle(371+bbt,386,409+bbt,404);
                                    rectangle(372+bbt,387,408+bbt,403);
                                    setcolor(15);
                                    line(371+bbt,404,409+bbt,404);
```

```
                                        line(372+bbt,403,408+bbt,403);
                                        line(409+bbt,404,409+bbt,387);
                                        line(408+bbt,403,408+bbt,388);
                                        regs.x.ax=0x1;
                                        int86(0x33,&regs,&regs);
                                        mb=1;
                                        while(mb == 1)
                                                {
                                                        regs.x.ax=0x3;
                                                        int86(0x33,&regs,&regs);
                                                        mb = 0x1 & regs.x.bx;
                                                }
                                        regs.x.ax=0x2;
                                        int86(0x33,&regs,&regs);
                                }
                }
/*botones de escala para el tester */
if( pxb > 375 & pxb < 410 & pyb > 420 & pyb < 448)   // <
        {
                cambio = 23;
                if(R==1)
                        {
                          Rescala-=1;
                          if(Rescala == 0)
                                  Rescala = 5;

                        }
                if(I==1)
                        {
                          Iescala+=1;
                          if(Iescala == 6)
                                  Iescala = 1;
                        }
                (V==1)
                        {
                          Vescala+=1;
                          if(Vescala == 5)
                                  Vescala = 1;

                        }
                setcolor(8);
                ellipse(450,433,124,180,75,15); //luz blanca del botón
                ellipse(450,433,124,180,74,14);
                ellipse(450,433,124,180,73,13);
                setcolor(15);
                ellipse(450,433,180,235,75,15); //sombra del botón
                ellipse(450,433,180,235,74,14);
                ellipse(450,433,180,235,73,13);
                regs.x.ax=0x1;
                int86(0x33,&regs,&regs);
                mb=1;
```

```
while(mb == 1)
        {
                regs.x.ax=0x3;
                int86(0x33,&regs,&regs);
                mb = 0x1 & regs.x.bx;
        }
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
if (R==1)
          {
                if(Rescala==1)
                    {
                        RESC = RESC & 0xe0;        //11100000
                        RESC = RESC | 0x10;        //00010000
                        outp(0xf3f2, RESC);
                    }
                 if(Rescala==2)
                    {
                         RESC = RESC & 0xe0;        //11100000
                         RESC = RESC | 0x08;        //00001000
                         outp(0xf3f2, RESC);

                    }
                if(Rescala==3)
                    {
                        RESC = RESC & 0xe0;        //11100000
                        RESC = RESC | 0x04;        //00000100
                        outp(0xf3f2, RESC);

                    }
                if(Rescala==4)
                    {
                        RESC = RESC & 0xe0;        //11100000
                        RESC = RESC | 0x02;        //00000010
                        outp(0xf3f2, RESC);
                    }
                if(Rescala==5)
                    {
                         RESC = RESC & 0xe0;        //11100000
                         RESC = RESC | 0x01;        //00000001
                         outp(0xf3f2, RESC);

                    }
            }
   if(I==1)
          {
                     RESC= RESC |  0x20;            //00100000
                     outp(0xf3f2,RESC);
                    if(Iescala==1)
                          {
                              seltest = seltest & 0xe0;        //11100000
```

```
                                        seltest  = seltest | 0x01;          //00000001
                                        outp(0xf3f1, seltest);
                                    }

            if(Iescala==2)
                    {
                        seltest = seltest & 0xe0;        //11100000
                        seltest = seltest | 0x02;        //00000010
                        outp(0xf3f1, seltest);

                    }

            if(Iescala==3)
                    {
                        seltest = seltest & 0xe0;        //11100000
                        seltest = seltest | 0x04;        //00000100
                        outp(0xf3f1, seltest);
                    }

            if(Iescala==4)
                    {
                        seltest = seltest & 0xe0;        //11100000
                        seltest = seltest | 0x08;        //00001000
                        outp(0xf3f1, seltest);

                    }
            if(Iescala==5)
                    {
                        seltest = seltest & 0xe0;        //11100000
                        seltest = seltest | 0x10;        //00010000
                        outp(0xf3f1, seltest);
                    }
            delay(10);
            RESC= RESC &  0xDF;             //11011111
            outp(0xf3f2,RESC);
        }
if (V==1)
{
if(Vescala==1)
        {
            mc4066 = mc4066 & 0x0f;        //00001111
            mc4066        = mc4066 | 0x10;        //00010000
            outp(0xf3f3, mc4066);
        }
if(Vescala==2)
        {
            mc4066 = mc4066 & 0x0f;        //00001111
            mc4066 = mc4066 | 0x20;        //00100000
            outp(0xf3f3, mc4066);
        }
if(Vescala==3)
```

```
            {
                mc4066 = mc4066 & 0x0f;         //00001111
                mc4066 = mc4066 | 0x40;         //01000000
                outp(0xf3f3, mc4066);
            }
    if(Vescala==4)
            {
                mc4066 = mc4066 & 0x0f;         //00001111
                mc4066 = mc4066 | 0x80;         //10000000
                outp(0xf3f3, mc4066);
            }

    }

goto inicio;


}

if( pxb > 490 & pxb < 525 & pyb > 420 & pyb < 448)     // >
        {
                if(R==1)
                    {
                        Rescala+=1;
                        if(Rescala==6)
                                Rescala = 1;
                    }

                if(I==1)
                   {
                        Iescala-=1;
                        if(Iescala==0)
                                Iescala = 5;
                   }
                if(V==1)
                  {
                        Vescala-=1;
                        if(Vescala== 0)        // 4escalas
                                Vescala = 4;
                  }
                setcolor(8);        //luz blanca del botón
                ellipse(450,433,310,55,75,15);
                ellipse(450,433,310,54,74,14);
                ellipse(450,433,310,54,73,13);
                setcolor(15);     //sombra del botón
                ellipse(450,433,306,360,75,15);
                ellipse(450,433,306,360,74,14);
                ellipse(450,433,308,360,73,13);
                regs.x.ax=0x1;
                int86(0x33,&regs,&regs);
                mb=1;
```

```
while(mb == 1)
        {
                regs.x.ax=0x3;
                int86(0x33,&regs,&regs);
                mb = 0x1 & regs.x.bx;
        }
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
if(R==1)
        {
                if(Rescala==1)
                    {
                        RESC = RESC & 0xe0;       //11100000
                        RESC = RESC | 0x10;       //00010000
                        outp(0xf3f2, RESC);
                    }
                if(Rescala==2)
                    {
                        RESC = RESC & 0xe0;       //11100000
                        RESC = RESC | 0x08;       //00001000
                        outp(0xf3f2, RESC);
                    }
                if(Rescala==3)
                    {
                        RESC = RESC & 0xe0;       //11100000
                        RESC = RESC | 0x04;       //00000100
                        outp(0xf3f2, RESC);
                    }
                if(Rescala==4)
                    {
                        RESC = RESC & 0xe0;       //11100000
                        RESC = RESC | 0x02;       //00000010
                        outp(0xf3f2, RESC);
                    }
                if(Rescala==5)
                    {
                        RESC = RESC & 0xe0;       //11100000
                        RESC = RESC | 0x01;       //00000001
                        outp(0xf3f2, RESC);
                    }

        }
if(I==1)
        {
            RESC= RESC |  0x20;             //00100000
            outp(0xf3f2,RESC);
            if(Iescala==1)
                {
                    seltest = seltest & 0xe0;       //11100000
                    seltest  = seltest | 0x01;      //00000001
                    outp(0xf3f1, seltest);
```

```
                    }
        if(Iescala==2)
            {
                seltest = seltest & 0xe0;       //11100000
                seltest = seltest | 0x02;       //00000010
                outp(0xf3f1, seltest);
            }
        if(Iescala==3)
            {
                seltest = seltest & 0xe0;       //11100000
                seltest = seltest | 0x04;       //00000100
                outp(0xf3f1, seltest);
            }
        if(Iescala==4)
            {
                seltest = seltest & 0xe0;       //11100000
                seltest = seltest | 0x08;       //00001000
                outp(0xf3f1, seltest);
            }
        if(Iescala==5)
            {
                seltest = seltest & 0xe0;       //11100000
                seltest = seltest | 0x10;       //00010000
                outp(0xf3f1, seltest);
            }
        delay(10);
        RESC= RESC &  0xDF;             //11011111
        outp(0xf3f2,RESC);

    }
if(V==1)
    {
        if(Vescala==1)
            {
                mc4066 = mc4066 & 0x0f;         //00001111
                mc4066 = mc4066 | 0x10;         //00010000
                outp(0xf3f3, mc4066);
            }
        if(Vescala==2)
            {
                mc4066 = mc4066 & 0x0f;         //00001111
                mc4066 = mc4066 | 0x20;         //00100000
                outp(0xf3f3, mc4066);
            }
        if(Vescala==3)
            {
                mc4066 = mc4066 & 0x0f;         //00001111
                mc4066 = mc4066 | 0x40;         //01000000
                outp(0xf3f3, mc4066);
            }
        if(Vescala==4)
```

```
                {
                    mc4066 = mc4066 & 0x0f;         //00001111
                    mc4066 = mc4066 | 0x80;         //10000000
                    outp(0xf3f3, mc4066);
                }

            }

        goto inicio;



        }

 /*botón de la fuente de voltaje */

if( pxb > 560 & pxb < 600 & pyb > 20 & pyb < 50)
        {
            outp(0xf3f9, 0x00); //quita la protección de la fuente
            setcolor(8);
            rectangle(561,21,599,49);
            rectangle(562,22,598,48);
            setcolor(15);
            line(561,49,599,49);
            line(562,48,599,48);
            line(599,49,599,21);
            line(598,48,598,22);
            regs.x.ax=0x1;
            int86(0x33,&regs,&regs);
            mb=1;
            while(mb == 1)
                {
                    regs.x.ax=0x3;
                    int86(0x33,&regs,&regs);
                    mb = 0x1 & regs.x.bx;
                }
            regs.x.ax=0x01;
            int86(0x33,&regs,&regs);
            ////
             sal:
            setfillstyle(1,15);
            bar(DX1 +85,DY1 - 25,DX1 -65,DY1 -2);
        veces=0,veces1=0,punto=0,cero=0,prue=0,mismo=0,diferente=0,primero1=0;
            segundo1=0,tercero1=0,cuarto1=0,quinto1=0,sexto1=0;
            primero=0,segundo=0,tercero=0,cuarto=0,quinto=0,sexto=0;
            settextstyle(1,0,1);
            setcolor(COLOF);
            outtextxy(DX1,DY1-20,"0V");
            while(tec==0)
                {
                    while (bioskey(1) == 0);
```

```
        {
                // function 0 returns the key that is waiting
                key = bioskey(0);
        }

// use function 2 to determine if shift keys were used
// print out the character read
 if (isdigit(key & 0xff  ))
        {
                if(key==0x5230)
                decimal=0;

                if(key==0x4f31)
                        decimal=1;

                if(key==0x5032)
                        decimal=2;

                if(key==0x5133)
                        decimal=3;

                if(key==0x4b34)
                        decimal=4;

                if(key==0x4c35)
                        decimal=5;

                if(key==0x4d36)
                decimal=6;

                if(key==0x4737)
                        decimal=7;

                if(key==0x4838)
                        decimal=8;

                if(key==0x4939)
                        decimal=9;

                 if(key==0x5230 & diferente!=1)
                        {
                                mismo=1;
                        }
                 if(key!=0x5230)
                        {
                                diferente=1;
                                mismo=0;
                        }

if(punto==0 & (veces>=0 & veces <=1))
        {
```

```
                        cuarto = tercero;
                        tercero = segundo;
                        segundo = primero;
                        primero = decimal;
                }
            if (punto==1 & (veces1>=0 & veces1<=1))
                {
                        if( veces1==0)
                                {
                                        prue=1;
                                        quinto=decimal;
                                }
                        if(veces1==1)
                                        sexto=decimal;
                }

            if(mismo==0)
                {
                        cero++;
                        veces++;
                        veces1++;
                }
                        }
if(key==0x5300| key==0x11b )
        goto sal;

if(key==0x532e )
        {
            punto++;
            veces=0;
            veces1=0;
            mismo=0;
            diferente=1;

        }

if(key==0x1c0d)
        {
            setcolor(15);
fuentev= primero + segundo*10 + tercero*100 +cuarto *1000
                +quinto/10 +sexto/100;
if(fuentev>=12.01)
        {
              setfillstyle(1,15);
              bar(DX1 + 85,DY1 - 25,DX1 -65,DY1 -2);
              setcolor(COLOF);
              outtextxy(DX1-40,DY1-20,"1");
              outtextxy(DX1-30,DY1-20,"2");
              outtextxy(DX1-20,DY1-20,".");
              outtextxy(DX1-10,DY1-20,"0");
              outtextxy(DX1,DY1-20,"0");
```

```
                outtextxy(DX1+10,DY1-20,"V");
                fuentev=12;
        }
fuentev =(fuentev / 22.1875e-03) / 2.1856;
outp(0xf3f8,fuentev);
goto sal1;
        }
if(key==0x532e  & punto==2)
        if(prue!=1)
                {
                        segundo1=0;
                        primero1=0;
                        veces=0;
                        veces1=0;
                        punto=1;
                }

 if(punto==1 & veces==0 & cero==0 )
        primero1=0x5230;
 if (isdigit(key & 0xff  ) | key==0x532e )
        if( punto==0 & (veces>=0 & veces <=2) |  punto==1 & (veces1>=0 &
        veces1<=2))
                {
                        if(mismo==0)
                                {
                                        sexto1 = quinto1;
                                        quinto1 = cuarto1;
                                        cuarto1 = tercero1;
                                        tercero1 = segundo1;
                                        segundo1 = primero1;
                                        primero1 = key;
                                        setcolor(15);
                                        if(veces==1)
                                        outtextxy(DX1,DY1-20,"0");
                                        sprintf(key3,"%c",sexto1);
                                        outtextxy(DX1-40,DY1-20, key3);
                                        sprintf(key3,"%c",quinto1);
                                        outtextxy(DX1-30,DY1-20, key3);
                                        sprintf(key3,"%c",cuarto1);
                                        outtextxy(DX1-20,DY1-20, key3);
                                        sprintf(key3,"%c",tercero1);
                                        outtextxy(DX1-10,DY1-20, key3);
                                        sprintf(key3,"%c",segundo1);
                                        outtextxy(DX1,DY1-20, key3);
                                        setcolor(COLOF);
                                        sprintf(key3,"%c",primero1);
                                        outtextxy(DX1,DY1-20, key3);
                                        sprintf(key3,"%c",segundo1);
                                        outtextxy(DX1-10,DY1-20, key3);
                                        sprintf(key3,"%c",tercero1);
                                        outtextxy(DX1-20,DY1-20, key3);
```

```
                                        sprintf(key3,"%c",cuarto1);
                                        outtextxy(DX1-30,DY1-20, key3);
                                        sprintf(key3,"%c",quinto1);
                                        outtextxy(DX1-40,DY1-20, key3);


                                }
                        }
        }
sal1:
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
///
goto inicio;
}

/*botón del frecuencímetro */
if( pxb > 560 & pxb < 600 & pyb > 60 & pyb < 90)
        {
                setcolor(8);
                rectangle(561,61,599,89);
                rectangle(562,62,598,88);
                setcolor(15);
                line(561,89,599,89);
                line(562,88,599,88);
                line(599,89,599,61);
                line(598,88,598,62);
                regs.x.ax=0x1;
                int86(0x33,&regs,&regs);
                mb=1;
                while(mb == 1)
                        {
                                regs.x.ax=0x3;
                                int86(0x33,&regs,&regs);
                                mb = 0x1 & regs.x.bx;
                        }
                regs.x.ax=0x2;
                int86(0x33,&regs,&regs);
                goto inicio;
        }

 if(pxbg > 50 & pxbg < 100 & pybg > 18 + bbg  & pybg < 38 + bbg)
        {
                if(botong < 4)
                  {
                      botg=botong;
                      setcolor(8);
                      rectangle(51,19+bbg,99,37+bbg);
                      rectangle(52,20+bbg,98,36+bbg);
                      setcolor(15);
```

```
                    line(51,37+bbg,99,37+bbg);
                    line(52,36+bbg,98,36+bbg);
                    line(99,37+bbg,99,19+bbg);
                    line(98,36+bbg,98,20+bbg);
                }
        }

if(pxbg1 > 305 & pxbg1 < 355 & pybg1 > 18 + bbg1  & pybg1 < 38 + bbg1)
        {
                if(botong < 4)
                {
                    botg1=botong1;
                    onda=botg1;
                    setcolor(8);
                    rectangle(306,19+bbg1,354,37+bbg1);
                    rectangle(307,20+bbg1,353,36+bbg1);
                    setcolor(15);
                    line(306,37+bbg1,354,37+bbg1);
                    line(307,36+bbg1,353,36+bbg1);
                    line(354,37+bbg1,354,19+bbg1);
                    line(353,36+bbg1,353,20+bbg1);
                }
        }
if(pxb>572 & pxb<588 & pyb>310 & pyb<330)   // cursor de abajo
        {
                setcolor(8);
                rectangle(573,311,587,329);
                setcolor(15);
                line(573,329,587,329);
                line(587,329,587,311);
                conamp++;
                if(conamp > 7)
                        conamp=0;
                escala=escala & 0x08;
                escala=conamp | escala;
                outp(0xf3fe,escala);
                if (conamp!=conamp1)
                        {
                                setcolor(0);
                                sprintf(amplitud,"%s",voltdiv[conamp1]);
                                outtextxy(IZQ,180,amplitud);
                                setcolor(10);
                                sprintf(amplitud,"%s",voltdiv[conamp]);
                                outtextxy(IZQ,180,amplitud);
                                conamp1=conamp;
                        }
                regs.x.ax=0x1;
                int86(0x33,&regs,&regs);
                mb=1;
                while(mb == 1)
                        {
```

```
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
        }
regs.x.ax=0x2;
nt86(0x33,&regs,&regs);
goto inicio;

            }

if(pxb>572 & pxb<588 & pyb>224 & pyb<244)   //  cursor de arriba
        {
                setcolor(8);
                rectangle(573,225,587,243);
                setcolor(15);
                line(573,243,587,243);
                line(587,243,587,225);
                conamp--;
                if(conamp < 0)
                        conamp=7;
                 escala=escala & 0x08;
                escala=conamp | escala;
                outp(0xf3fe,escala);
                if (conamp!=conamp1)
                        {
                                setcolor(0);
                                sprintf(amplitud,"%s",voltdiv[conamp1]);
                                outtextxy(IZQ,180,amplitud);
                                setcolor(10);
                                sprintf(amplitud,"%s",voltdiv[conamp]);
                                outtextxy(IZQ,180,amplitud);
                                conamp1=conamp;
                        }
                regs.x.ax=0x1;
                int86(0x33,&regs,&regs);
                mb=1;
                while(mb == 1)
                        {
                            regs.x.ax=0x3;
                            int86(0x33,&regs,&regs);
                            mb = 0x1 & regs.x.bx;
                        }
                 regs.x.ax=0x2;
                 int86(0x33,&regs,&regs);
                 goto inicio;
        }

if(pxb>542 & pxb<562 & pyb>269 & pyb<285)  //cursor  <
        {
                setcolor(8);
                rectangle(543,270,561,284);
```

```
                    setcolor(15);
                    line(543,284,561,284);
                    line(561,284,561,270);
                    conlon--;
                    if(conlon < 0)
                    conlon=11;
                    if(conlon < 7 & conlon >= 0)
                        {
                            variable = 0x58+conlon;
                        }
                    if(conlon ==7)
                        variable=0x09;
                    if(conlon ==8)
                        variable=0x1a;
                    if(conlon ==9)
                        variable=0x2b;
                    if(conlon ==10)
                        variable=0x3c;
                    if(conlon ==11)
                        variable=0x4d;
                    outp(0xf3f0,variable); //Selección de escala time/div.

                    if (conlon!=conlon1)
                        {
                            setcolor(0);
                            sprintf(longitud,"%s",timediv[conlon1]);
                            outtextxy(IZQ,195,longitud);
                            setcolor(10);
                            sprintf(longitud,"%s",timediv[conlon]);
                            outtextxy(IZQ,195,longitud);
                            conlon1=conlon;
                        }
                    regs.x.ax=0x1;
                    int86(0x33,&regs,&regs);
                    mb=1;
                    while(mb == 1)
                        {
                            regs.x.ax=0x3;
                            int86(0x33,&regs,&regs);
                            mb = 0x1 & regs.x.bx;
                        }
                    regs.x.ax=0x2;
                    int86(0x33,&regs,&regs);
                    goto inicio;
                }

if(pxb>598 & pxb<618 & pyb>269 & pyb<285)  //cursor  >
    {
        setcolor(8);
        rectangle(599,270,617,284);
        setcolor(15);
```

```
line(599,284,617,284);
line(617,284,617,270);
 conlon++;
if(conlon > 11)
        conlon=0;

if(conlon < 7 & conlon >= 0)
        {
            variable = 0x58+conlon;
        }
if(conlon ==7)
        variable=0x09;
if(conlon ==8)
        variable=0x1a;
if(conlon ==9)
        variable=0x2b;
if(conlon ==10)
        variable=0x3c;
if(conlon ==11)
        variable=0x4d;
outp(0xf3f0,variable); //Selección de escala time/div.
if (conlon!=conlon1)
        {
                setcolor(0);
                sprintf(longitud,"%s",timediv[conlon1]);
                outtextxy(IZQ,195,longitud);
                setcolor(10);
                sprintf(longitud,"%s",timediv[conlon]);
                outtextxy(IZQ,195,longitud);
                conlon1=conlon;
        }
regs.x.ax=0x1;
int86(0x33,&regs,&regs);
mb=1;
while(mb == 1)
    {
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
    }
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
goto inicio;

    }
if(pxb > 15 & pxb< 65 & pyb>110+bb & pyb<140+bb)
    {
        bot=boton;
        setcolor(8);
        rectangle(16,111+bb,64,139+bb);
        rectangle(17,112+bb,63,138+bb);
```

```
setcolor(15);
line(16,139+bb,64,139+bb);
line(17,138+bb,63,138+bb);
line(64,139+bb,64,111+bb);
line(63,138+bb,63,112+bb);
regs.x.ax=0x1;
int86(0x33,&regs,&regs);

/*PARA EL NUMERO UNO*/

switch(bot){   //scope

case 1:
pulsador1++;
b[bot]= bot;
grid=1;
if ((pulsador1/2) != 0)
    {
        grid=0;
        b[bot]=8;
        pulsador1=0;
    }
setfillstyle(1,12);
floodfill(83,123,8);
setcolor(7);
for(linea=0;linea<=420;linea+=30)
        line (100+ linea,100,100 + linea,340);
for (linea=0;linea<=240;linea+=30)
        line (100,100+ linea,520,100+ linea);
while(mb == 1)
        {
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
        }
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
goto inicio;
break;

case 2:
pulsador2++;
b[bot]= bot;
if ((pulsador2/2) != 0)
    {
        b[bot]=8;
        pulsador2=0;
        escala=escala & 0x07;
    }
else
    escala=escala | 0xf8;
```

```
            outp(0xf3fe,escala);

        setfillstyle(1,12);
        floodfill(83,183,8);
        while(mb == 1)
            {
                regs.x.ax=0x3;
                int86(0x33,&regs,&regs);
                mb = 0x1 & regs.x.bx;
            }
        regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
        goto inicio;
        break;
//////////////////////cotas inicio:
        case 3:             //Entra si se presionó HOLD

        regs.x.ax=0x2;      //apago el Mouse
        int86(0x33,&regs,&regs);
        setfillstyle(1,12);
        floodfill(83,243,8); //led rojo de hold
         regs.x.ax=0x1;          //activa el Mouse
        int86(0x33,&regs,&regs);
        memoria=100;
        memoria1=100;
        memoria2=340;
        memoria3=340;
    for(color=0;color<=240;color++)  //asigno color gris a todas las cotas
          n[color]=7;
        for(color=0;color<=240;color++)
            k[color]=7;
        for(color=0;color<=420;color++)
            w[color]=7;
        for(color=0;color<=420;color++)
            s[color]=7;
        col=0;
        c=0;
        mb=1;
        while(mb == 1)              //detecta coordenadas clic izquierdo
            {
                regs.x.ax=0x3;          //se sale hasta que se deje de
                                        presionar el clic izquierdo
                int 86(0x33,&regs,&regs);
                mb = 0x1 & regs.x.bx;
            }
        col=0;
        while (c==0)
            {
                REGRESO:
                coory=regs.x.dx;      //asigna las coordenadas X,Y
                coorx=regs.x.cx;
```

```
                    int86 (0x33, &regs, &regs);
                    if((regs.x.bx & 0x01) != 0)
                        {
                            coory=regs.x.dx;
                            coorx=regs.x.cx;
            if(coorx > 15 & coorx< 65 & coory>230 & coory<260)    //en caso de
                    que este la opción GRID
                    {

                        mb=1;
                        while(mb == 1)
                        {
                                regs.x.ax=0x3;
                                int86(0x33,&regs,&regs);
                                mb = 0x1 & regs.x.bx;
                        }
            if(grid==1)                //SI ESTA LA OPCION GRID
                {
                    setfillstyle(1,15);    //LIMPIA LA PANTALLA
                    bar (101,101,519,339);
                    setcolor(7);          //CONSTRULLE EL GRID
                    for(linea=0;linea<=420;linea+=30)
                            line (100+ linea,100,100 + linea,340);
                    for (linea=0;linea<=240;linea+=30)
                            line (100,100+ linea,520,100+ linea);
                }
            regs.x.ax=0x2;
            int86(0x33,&regs,&regs);   //MOUSE OFF
            cont=0;
            goto inicio;

        }
    }
if (conlon ==0)                //si esta la escala 0.5æS/div
        {
            escal=16.666666666666e-09;    //(0.5æS/30 pixeles) (escala time)
            inter=1e06;                //Factor de escala múltiplos
             uni=0;
        }
if (conlon ==1)
        {
            escal=33.3333333333333e-09;
             inter=1e06;
             uni=0;
        }
if (conlon ==2)
        {
            escal=166.666666666666e-09;
            inter=1e06;
             uni=0;
        }
```

```
if (conlon ==3)
        {
                escal=333.333333333e-09;
                inter=1e06;
                uni=0;
        }
 if (conlon ==4)
        {
                escal=1.666666667e-06;
                inter=1e06;
                uni=0;
        }
 if (conlon ==5)
        {
                escal=3333.3333333333333e-09;
                inter=1e03;
                uni=1;
        }
 if (conlon ==6)
        {
                escal=16.666666666666666667e-06;
                inter=1e03;
                uni=1;
        }
 if (conlon ==7)
        {
                escal=33.333333333e-06;
                inter=1e03;
                uni=1;
        }
 if (conlon ==8)
        {
                escal=166.66666667e-06;
                inter=1e03;
                uni=1;
        }
 if (conlon ==9)
        {
                escal=333.3333333333e-06;
                inter=1e03;
                uni=1;
        }
if (conlon ==10)
        {
                escal=1.666666667e-03;
                inter=1e03;
                uni=1;
        }
 if (conlon ==11)
        {
                escal=3.3333333333e-03;
```

```
                inter=1e03;
                uni=1;
        }
 if(conamp==0)
        {
                escal1=333.33333333e-03;
                uni1=0;
        }
 if(conamp==1)
        {
                escal1=166.66666666667e-03;
                uni1=0;
        }
 if(conamp==2)
        {
                escal1=66.666666666667e-03;
                uni1=0;
        }
 if(conamp==3)
        {
                escal1=33.33333333e-03;
                uni1=0;
        }
 if(conamp==4)
        {
                escal1=16.666666666667e-03;
                uni1=0;
        }
 if(conamp==5)
        {
                escal1=6.6666666666667e-03;
                uni1=0;
        }
if(conamp==6)
        {
                escal1=3.33333333333333e-03;
                uni1=0;
        }
 if(conamp==7)
        {
                escal1=1.6666666666666667e-03*1e03;
                uni1=1;
        }


regs.x.ax=0x03;
int86 (0x33,&regs, &regs);
if((regs.x.bx & 0x02)==2)
        {
                cont++;
                if (cont ==1)
```

```
                {
                  regs.x.ax=0x03;
                  int86 (0x33,&regs, &regs);
                  salir=regs.x.bx;
                   while (salir ==0x02)
                        {
                           regs.x.ax=0x03;
        //lee posición del Mouse
                           int86 (0x33,&regs, &regs);
                           regs.x.bx =0x00;
                           regs.x.ax=0x03;
        //lee posición del Mouse
                           int86 (0x33,&regs, &regs);
                           salir=regs.x.bx;
                        }

                           goto PRIMERO;
                }
        if (cont ==2)
                {
                  regs.x.ax=0x03;
//lee posición del Mouse
                  int86 (0x33,&regs, &regs);
                  salir=regs.x.bx;
                  while (salir ==0x02)
                        {
                           regs.x.ax=0x03;
        //lee posición del Mouse
                           int86 (0x33,&regs, &regs);
                           regs.x.bx =0x00;
                           regs.x.ax=0x03;
        //lee posición del Mouse
                           int86 (0x33,&regs, &regs);
                           salir=regs.x.bx;
                        }
                                goto SEGUNDO;
                }
        if (cont ==3)
                {
                  regs.x.ax=0x03;
//lee posición del Mouse
                  int86 (0x33,&regs, &regs);
                  salir=regs.x.bx;
                  while (salir ==0x02)
                        {
                           regs.x.ax=0x03;
        //lee posición del Mouse
                           int86 (0x33,&regs, &regs);
                           regs.x.bx =0x00;
                           regs.x.ax=0x03;
        //lee posición del Mouse
```

```
                        int86 (0x33,&regs, &regs);
                        salir=regs.x.bx;
                    }
                            goto TERCERO;
                }

            if (cont ==4)
                {
                  regs.x.ax=0x03;
                  int86 (0x33,&regs, &regs);
                  salir=regs.x.bx;
                  while (salir ==0x02)
                        {
                        regs.x.ax=0x03;
                        int86 (0x33,&regs, &regs);
                        regs.x.bx =0x00;
                        regs.x.ax=0x03;
                        int86 (0x33,&regs, &regs);
                        salir=regs.x.bx;
                    }
                            goto CUARTO;
                }


              if (cont > 4)
                {
                        cont=0;
                        goto REGRESO;
                }
        }
}
 col=0;
 while (d==0)
 {
      PRIMERO:
      regs.x.ax=0x03;
      int86 (0x33,&regs, &regs);
      if((regs.x.bx & 0x01) != 0)
            {

              ++col;
              coory=regs.x.dx;
              coorx=regs.x.cx;

              if(coorx > 15 & coorx< 65 & coory>230 & coory<260)
                {


                        mb=1;
                        while(mb == 1)
                              {
```

```
                                regs.x.ax=0x3;
                                int86(0x33,&regs,&regs);
                                mb = 0x1 & regs.x.bx;
                        }       if(grid==1)
                                {
                                        setfillstyle(1,15);
                                        bar (101,101,519,339);
                                        setcolor(7);
                                        for(linea=0;linea<=420;linea+=30)
                                line (100+ linea,100,100 + linea,340);
                                        for (linea=0;linea<=240;linea+=30)
                                        line (100,100+ linea,520,100+ linea);
                                }
                regs.x.ax=0x2;
                int86(0x33,&regs,&regs);
                cont=0;
                goto inicio;

    }
if (coorx >=100 & coorx <=520 & coory >=100 & coory<=340)
{
    if(coorx !=memoria)
    {
            regs.x.ax=0x02;
            int86 (0x33, &regs, &regs);
            if (col!=2)
                {
                        rey=100;
                        for(color=0;color<=239;color++)
                        {
                        putpixel(memoria,rey,n[color]);
                        rey++;

                        }
                        col=0;
                }

          rey=100;
          for(color=0;color<=239;color++)
          {
                n[color]=getpixel(coorx,rey);

                rey++;
          }
          setcolor (COTA1);
          line (coorx,101,coorx,339);
          regs.x.ax=0x01;
          int86 (0x33, &regs, &regs);
          memoria=coorx;

    }
```

```
    }
if(memoria> memoria1)
 pasa=(memoria- memoria1) * escal;
if(memoria1> memoria)
 pasa=(memoria1- memoria) * escal;
tiempo=(1/pasa);

time=pasa*inter;
if( der!=time )
        {

                if (tiempo>=1e06)
                        {
                        tiempo=tiempo/1e06;
                        uni2=2;
                        goto WER;
                        }
                if (tiempo>= 1e03 & tiempo <1e06)
                        {
                        tiempo=tiempo/1e03;
                        uni2=1;
                        goto WER;
                        }
                if (tiempo >0 & tiempo < 1e03)
                        {
                        uni2=0;
                        }
                        WER:
setcolor(FOND);
settextstyle(LETRA, 0,TAMA);
sprintf(inpu,"%3.2f%s",dere,hey2[change]);
outtextxy(IZQ,ABAJO,inpu);
sprintf(input,"%3.2f%s",der,hey[uni4]);
outtextxy(IZQ,ENMEDIO,input);
setcolor(COLO);
settextstyle(LETRA, 0,TAMA);
sprintf(inpu,"%3.2f%s",tiempo,hey2[uni2]);
outtextxy(IZQ,ABAJO,inpu);
sprintf(input,"%3.2f%s",time,hey[uni]);
outtextxy(IZQ,ENMEDIO,input);
der=time;
dere=tiempo;
change=uni2;
uni4=uni;
        }
 }
  regs.x.ax= 0x03;
  int86 (0x33, &regs, &regs);

  if((regs.x.bx & 0x02) == 0x02)
  goto REGRESO1;
```

```
}
 while (e==0)
 {  SEGUNDO:
      regs.x.ax=0x03;                    //lee posición del Mouse
      int86 (0x33,&regs, &regs);         //entra  si se presiono el botón izquierdo
     if((regs.x.bx & 0x01) != 0)
     {

        ++col1;
        coory1=regs.x.dx;
        coorx1=regs.x.cx;

        if(coorx1 > 15 & coorx1< 65 & coory1>230 & coory1<260)
           {


                        mb=1;
                        while(mb == 1)
                              {
                                    regs.x.ax=0x3;
                                    int86(0x33,&regs,&regs);
                                    mb = 0x1 & regs.x.bx;
                              }
                                    if(grid==1)
                                    {
                                          setfillstyle(1,15);
                                          bar (101,101,519,339);
                                          setcolor(7);
                                          for(linea=0;linea<=420;linea+=30)
                                    line (100+ linea,100,100 + linea,340);
                                          for (linea=0;linea<=240;linea+=30)
                                          line (100,100+ linea,520,100+ linea);
                                    }
                        regs.x.ax=0x2;
                        int86(0x33,&regs,&regs);
                        cont=0;
                        goto inicio;

           }
        if (coorx1 >=100 & coorx1 <=520 & coory1 >=100 & coory1<=340)
                  //para solo marcar la linea dentro del cuadro
        {
           if(coorx1 != memoria1)
           {
                  regs.x.ax=0x02;            //apago el Mouse
                  int86 (0x33, &regs, &regs);
                  if (col1!=2)
                        {
                              rey1 =100;
                              for(color1=0;color1<=239;color1 ++)
                              {
```

```
                        putpixel(memoria1,rey1,k[color1]);
                        rey1++;
                        }
                        col1=0;
                }
            rey1=100;
            for(color1=0;color1<=239;color1++)
            {
                k[color1]=getpixel(coorx1,rey1);

                rey1++;
            }
            setcolor (COTA2);
            line (coorx1,101,coorx1,339);
            regs.x.ax=0x01;     //enciendo el Mouse
            int86 (0x33, &regs, &regs);
            memoria1=coorx1;

        }
    }

if(memoria> memoria1)
 pasa=(memoria- memoria1)* escal;
if(memoria1> memoria)
 pasa=(memoria1- memoria)* escal;
tiempo=(1/pasa);

time=pasa* inter;
if( der!=time )
        {
            if (tiempo>=1e06)
                {
                tiempo=tiempo/1e06;
                uni2=2;
                goto WER1;
                }
            if (tiempo>= 1e03 & tiempo <1e06)
                {

                tiempo=tiempo/1e03;
                uni2=1;
                goto WER1;
                }
            if (tiempo >0 & tiempo < 1e03)
                {
                uni2=0;
                }
                WER1:
setcolor(FOND);
settextstyle(LETRA, 0,TAMA);
sprintf(inpu,"%3.2f%s",dere,hey2[change]);
```

```
outtextxy(IZQ,ABAJO,inpu);
sprintf(input,"%3.2f%s",der,hey[uni4]);
outtextxy(IZQ,ENMEDIO,input);
setcolor(COLO);
settextstyle(LETRA, 0,TAMA);
sprintf(inpu,"%3.2f%s",tiempo,hey2[uni2]);
outtextxy(IZQ,ABAJO,inpu);
sprintf(input,"%3.2f%s",time,hey[uni]);
outtextxy(IZQ,ENMEDIO,input);
der=time;
dere=tiempo;
change=uni2;
uni4=uni;
        }
 }
 regs.x.ax= 0x03;
 int86 (0x33, &regs, &regs);
 if((regs.x.bx & 0x02) == 0x02)
 goto REGRESO2;
}
   while (i==0)                    //para salir del programa botón derecho
{
TERCERO:
regs.x.ax=0x03;                    //lee posición del Mouse
int86 (0x33,&regs, &regs);
if ((regs.x.bx & 0x01) != 0)       //entra  si se presiono el botón izquierdo
        {
            ++col2;
            coory2=regs.x.dx;
            coorx2=regs.x.cx;

            if(coorx2 > 15 & coorx2< 65 & coory2>230 & coory2<260)
              {

                    mb=1;
                    while(mb == 1)
                          {
                                regs.x.ax=0x3;
                                int86(0x33,&regs,&regs);
                                mb = 0x1 & regs.x.bx;
                          }
                                if(grid==1)
                                {
                                      setfillstyle(1,15);
                                      bar (101,101,519,339);
                                      setcolor(7);
                                      for(linea=0;linea<=420;linea+=30)
                                line (100+ linea,100,100 + linea,340);
                                      for (linea=0;linea<=240;linea+=30)
                                      line (100,100+ linea,520,100+ linea);
                                }
```

```
                regs.x.ax=0x2;
                int86(0x33,&regs,&regs);
                cont=0;
                goto inicio;

            }
    if (coorx2 >=100 & coorx2 <=520 & coory2 >=100 & coory2<=340)
            //para solo marcar la linea dentro del cuadro
        {
            if(coory2 != memoria2)
                {
                    regs.x.ax=0x02; //apago el Mouse
                    int86 (0x33, &regs, &regs);
                    if (col2!=2)
                        {
                            rey2 =100;
                        for(color2=0;color2<=420;color2 ++)
                            {
                            putpixel(rey2,memoria2,s[color2]);
                            rey2++;

                            }
                        col2=0;
                }

            rey2=100;
            for(color2=0;color2<=420;color2++)
                {
                    s[color2]=getpixel(rey2,coory2);
                    rey2++;
                }

            setcolor (COTA3);
            line (101,coory2,519,coory2);
            regs.x.ax=0x01;        //enciendo el Mouse
            int86 (0x33, &regs, &regs);
            memoria2=coory2;

        }
    }
        if(memoria3> memoria2)
time1=(memoria3- memoria2)* escal1;
if(memoria2> memoria3)
time1=(memoria2- memoria3)*escal1;
if(der1!=time1)
        {
            setcolor(FOND);
            sprintf(entrada1,"%3.2f%s",der1,hey1[uni3]);
            outtextxy(IZQ,ARRIB,entrada1);
            setcolor(COLO);
            sprintf(entrada1,"%3.2f%s",time1,hey1[uni1]);
```

```
                settextstyle(LETRA, 0,TAMA);
                outtextxy(IZQ,ARRIB,entrada1);
                der1=time1;
                uni3=uni1;
            }

        }
         regs.x.ax= 0x03;
         int86 (0x33, &regs, &regs);
         if((regs.x.bx & 0x02) == 0x02)
         goto REGRESO3;

}

 while (l==0)
 {
     CUARTO:
     regs.x.ax=0x03;
     int86 (0x33,&regs, &regs);
     if ((regs.x.bx & 0x01) != 0)
            {

                ++col3;
                coory3=regs.x.dx;
                coorx3=regs.x.cx;

                if(coorx3 > 15 & coorx3< 65 & coory3>230 & coory3<260)
                    {
                        mb=1;
                        while(mb == 1)
                            {
                                regs.x.ax=0x3;
                                int86(0x33,&regs,&regs);
                                mb = 0x1 & regs.x.bx;
                            }
                                if(grid==1)
                                {
                                    setfillstyle(1,15);
                                    bar (101,101,519,339);
                                    setcolor(7);
                                    for(linea=0;linea<=420;linea+=30)
                                line (100+ linea,100,100 + linea,340);
                                    for (linea=0;linea<=240;linea+=30)
                                    line (100,100+ linea,520,100+ linea);
                                }
                        regs.x.ax=0x2;
                        int86(0x33,&regs,&regs);
                        cont=0;
                        goto inicio;

                    }
```

```
        if (coorx3 >=100 & coorx3 <=520 & coory3 >=100 & coory3<=340)
                {
                        if(coory3 != memoria3)
                                {
                                        regs.x.ax=0x02;
                                        int86 (0x33, &regs, &regs);
                        if (col3!=2)
                                {
                                        rey3 =100;
                                        for(color3=0;color3<=420;color3 ++)
                                                {
                                                        putpixel(rey3,memoria3,w[color3]);
                                                        rey3++;

                                                }
                                        col3=0;
                                }

                        rey3=100;
                        for(color3=0;color3<=420;color3++)
                                {
                                        w[color3]=getpixel(rey3,coory3);
                                        rey3++;
                                }

                        setcolor (COTA4);
                        line (101,coory3,519,coory3);
                        regs.x.ax=0x01;
                        int86 (0x33, &regs, &regs);
                        memoria3=coory3;
                }

  }
if(memoria3> memoria2)
        time1=(memoria3- memoria2)* escal1;
if(memoria2> memoria3)
        time1=(memoria2- memoria3)*escal1;
if(der1!=time1)
        {
                setcolor(FOND);
                sprintf(entrada1,"%3.2f%s",der1,hey1[uni3]);
                outtextxy(IZQ,ARRIB,entrada1);
                setcolor(COLO);
                sprintf(entrada1,"%3.2f%s",time1,hey1[uni1]);
                settextstyle(LETRA, 0,TAMA);
                outtextxy(IZQ,ARRIB,entrada1);
                der1=time1;
                uni3=uni1;
        }

}
```

```
regs.x.ax= 0x03;
int86 (0x33, &regs, &regs);
if((regs.x.bx & 0x02) == 0x02)
goto REGRESO4;
 }
        REGRESO1:
        rey=100;
        for(color=0;color<=240;color++)
            {
                    sol=getpixel(memoria1,rey);
                    if(sol==COTA1)
                        {
                            tercel=k[memoria-100];
                            n[memoria1-100]=tercel;
                            k[color]=sol;
                        }
                    rey++;
            }
      rey=100;
      for(color=0;color<=420;color++)
            {

                    sol=getpixel(rey,memoria3);
                    if(sol==COTA1)
                        {
                            tercel=w[memoria-100];
                            n[memoria3-100]=tercel;
                            w[color]=sol;
                        }
                        rey++;
            }
      rey=100;
      for(color=0;color<=420;color++)
            {

                    sol=getpixel(rey,memoria2);
                    if(sol==COTA1)
                        {
                            tercel=s[memoria-100];
                            n[memoria2-100]=tercel;
                            s[color]=sol;
                        }
                        rey++;
            }

 goto REGRESO;

 REGRESO2:
      rey1=100;
      for(color1=0;color1<=240;color1++)
            {
```

```
                sol1=getpixel(memoria,rey1);
                if(sol1==COTA2)
                    {
                        tercel1=n[memoria1-100];
                        k[memoria-100]=tercel1;
                        n[color1]=sol1;
                    }
                rey1++;
            }
        rey1=100;
        for(color1=0;color1<=420;color1++)
            {
                sol1=getpixel(rey1,memoria3);
                if(sol1==COTA2)
                    {
                        tercel1=w[memoria1-100];
                        k[memoria3-100]=tercel1;
                        w[color1]=sol1;
                    }
                rey1++;
            }

        rey1=100;
        for(color1=0;color1<=420;color1++)
            {
                sol1=getpixel(rey1,memoria2);
                if(sol1==COTA2)
                    {
                        tercel1=s[memoria1-100];
                        k[memoria2-100]=tercel1;
                        s[color1]=sol1;
            }
                rey1++;
    }
goto REGRESO;

REGRESO3:
        rey2=100;
        for(color2=0;color2<=240;color2++)
            {

                sol2=getpixel(memoria,rey2);
                if(sol2==COTA3)
                    {

                        tercel2=n[memoria2-100];
                        s[memoria-100]=tercel2;
                        n[color2]=sol2;
                    }
            rey2++;
```

```
        }
    rey2=100;
    for(color2=0;color2<=240;color2++)
        {
                sol2=getpixel(memoria1,rey2);
                if(sol2==COTA3)
                    {
                        tercel2=k[memoria2-100];
                        s[memoria1-100]=tercel2;
                        k[color2]=sol2;
                    }
                rey2++;
        }
    rey2=100;
    for(color2=0;color2<=420;color2++)
        {
                sol2=getpixel(rey2,memoria3);
                if(sol2==COTA3)
                    {
                        tercel2=w[memoria2-100];
                        s[memoria3-100]=tercel2;
                        w[color2]=sol2;
                    }
                rey2++;
        }

goto REGRESO;

REGRESO4:
    rey3=100;
    for(color3=0;color3<=240;color3++)
        {

                sol3=getpixel(memoria,rey3);
                if(sol3==COTA4)
                    {

                        tercel3=n[memoria3-100];
                        w[memoria-100]=tercel3;
                        n[color3]=sol3;
                    }
              rey3++;
        }
    rey3=100;
    for(color3=0;color3<=240;color3++)
        {
                sol3=getpixel(memoria1,rey3);
                if(sol3==COTA4)
                    {
                        tercel3=k[memoria3-100];
                        w[memoria1-100]=tercel3;
```

```
                                        k[color3]=sol3;
                                }
                        rey3++;
                }
        rey3=100;
        for(color3=0;color3<=420;color3++)
                {
                        sol3=getpixel(rey3,memoria2);
                        if(sol3==COTA4)
                                {
                                        tercel3=s[memoria3-100];
                                        w[memoria2-100]=tercel3;
                                        s[color3]=sol3;
                                }
                        rey3++;
                }

        ///////////////cotas fin

 goto REGRESO;


regs.x.ax=0x02;                                         //enciendo el Mouse
int86 (0x33, &regs, &regs);
goto inicio;
break;
case 4:
delay(120);
goto quit;
break;
 }
 }
 }
regs.x.ax=0x01;                                         //enciendo el Mouse
int86 (0x33, &regs, &regs);


switch(botg1)     //gen fun
        {
                case 1:
                senal=1;
                Y5w = Y5w &  0x7c; //0000 0001
                Y5w = Y5w | 0x01;
                outp(0xf3fd, Y5w);
                delay(20);
                Y5w=Y5w | 0x80;
                outp(0xf3fd, Y5w);
                 //ampligen = ampligen1;
                ampligen = (ampligen / 0.037465116)/1.5;
                outp(0xf3ff,ampligen);
                while(mb == 1)
```

```
                {
                        regs.x.ax=0x3;
                        int86(0x33,&regs,&regs);
                        mb = 0x1 & regs.x.bx;
                }
            regs.x.ax=0x2;
            int86(0x33,&regs,&regs);
            botg1=0;
            goto  inicio;
             break;

case 2:
senal=2;
Y5w = Y5w &  0x7c;
Y5w = Y5w | 0x02; //0101 1111
outp(0xf3fd, Y5w);
delay(20);
Y5w=Y5w | 0x80;
outp(0xf3fd, Y5w);

ampligen = ampligen1;
ampligen = (ampligen / 0.037465116)/1.833 ;
outp(0xf3ff, ampligen);

while(mb == 1)
      {
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
      }
regs.x.ax=0x2;
int86(0x33,&regs,&regs);
 botg1=0;
 goto  inicio;
 break;
 case 3:
 senal=3;
 Y5w = Y5w &  0x7c;
 outp(0xf3fd, Y5w);  //xx1x 0000
delay(20);
 Y5w=Y5w | 0x80;
 outp(0xf3fd, Y5w);

ampligen = ampligen1;
ampligen = (ampligen / 0.037465116)/1.5;
outp(0xf3ff,ampligen);
while(mb == 1)
      {
            regs.x.ax=0x3;
            int86(0x33,&regs,&regs);
            mb = 0x1 & regs.x.bx;
```

```
        }
 regs.x.ax=0x2;
 int86(0x33,&regs,&regs);
 botg1=0;
 goto  inicio;
 break;

  }



 switch(botg)    //gen fun
        {
                case 1:
                while(mb == 1)
                        {
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}     regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
 botg=0;
 goto  inicio;
 break;

case 2:
     while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}
         sali:
 setfillstyle(1,0);
 bar(DX - 85,DY - 22,DX +8,DY -10);
 veces=0,veces1=0,punto=0,cero=0,prue=0,mismo=0,diferente=0,primero1=0;
 segundo1=0,tercero1=0,cuarto1=0,quinto1=0,sexto1=0;
 primero=0,segundo=0,tercero=0,cuarto=0,quinto=0,sexto=0;
 setcolor(11);
 outtextxy(DX,DY-20,"0V");
 while(tec==0)
 {

        while (bioskey(1) == 0);
        {
                // function 0 returns the key that is waiting
                key = bioskey(0);
        }

                // use function 2 to determine if shift keys were used
                // print out the character read
```

```
if (isdigit(key & 0xff ))
{
                        if(key==0x5230)
                                decimal=0;

                        if(key==0x4f31)
                                decimal=1;

                        if(key==0x5032)
                                decimal=2;

                        if(key==0x5133)
                                decimal=3;

                        if(key==0x4b34)
                                decimal=4;

                        if(key==0x4c35)
                                decimal=5;

                        if(key==0x4d36)
                                decimal=6;

                        if(key==0x4737)
                                decimal=7;

                        if(key==0x4838)
                                decimal=8;

                        if(key==0x4939)
                                decimal=9;


        if(key==0x5230 & diferente!=1)
                {
                        mismo=1;
                }
        if(key!=0x5230)
        {
                diferente=1;
                mismo=0;
        }


        if(punto==0 & (veces>=0 & veces <=1))
                {
                        cuarto = tercero;
                        tercero = segundo;
                        segundo = primero;
```

```
                primero = decimal;

        }
    if (punto==1 & (veces1>=0 & veces1<=1))
        {

                if( veces1==0)
                        {
                        prue=1;
                        quinto=decimal;
                        }
                if(veces1==1)
                        sexto=decimal;
        }

        if(mismo==0)
        {
        cero++;
        veces++;
        veces1++;
        }
    }

if(key==0x5300| key==0x11b )
    goto sali;

if(key==0x532e )
    {
        punto++;
        veces=0;
        veces1=0;
        mismo=0;
        diferente=1;

    }

    if(key==0x1c0d)
        {
                setcolor(15);
        ampligen= primero + segundo*10 + tercero*100 +cuarto *1000
                +quinto/10 +sexto/100;
                if(ampligen>=15.01)
                {
                  setfillstyle(1,0);
                  bar(DX - 85,DY - 22,DX +8,DY -10);
                  setcolor(COLO);
                  outtextxy(DX-40,DY-20,"1");
                  outtextxy(DX-30,DY-20,"5");
                  outtextxy(DX-20,DY-20,".");
                  outtextxy(DX-10,DY-20,"0");
                  outtextxy(DX,DY-20,"0");
```

```
                    ampligen=15;
                }
                goto sali1;
            }
            if(key==0x532e  & punto==2)

            if(prue!=1)
            {
            segundo1=0;
            primero1=0;
            veces=0;
            veces1=0;
            punto=1;
            }

if(punto==1 & veces==0 & cero==0 )
    primero1=0x5230;
if (isdigit(key & 0xff  ) | key==0x532e )
    if( punto==0 & (veces>=0 & veces <=2) |  punto==1 & (veces1>=0 & veces1<=2))
        {
            if(mismo==0)
                {

                    sexto1 = quinto1;
                    quinto1 = cuarto1;
                    cuarto1 = tercero1;
                    tercero1 = segundo1;
                    segundo1 = primero1;
                    primero1 = key;
                    setcolor(0);
                    if(veces==1)
                    outtextxy(DX,DY-20,"0");
                    sprintf(key3,"%c",sexto1);
                    outtextxy(DX-40,DY-20, key3);
                    sprintf(key3,"%c",quinto1);
                    outtextxy(DX-30,DY-20, key3);
                    sprintf(key3,"%c",cuarto1);
                    outtextxy(DX-20,DY-20, key3);
                    sprintf(key3,"%c",tercero1);
                    outtextxy(DX-10,DY-20, key3);
                    sprintf(key3,"%c",segundo1);
                    outtextxy(DX,DY-20, key3);
                    setcolor(11);
                    sprintf(key3,"%c",primero1);
                    outtextxy(DX,DY-20, key3);
                    sprintf(key3,"%c",segundo1);
                    outtextxy(DX-10,DY-20, key3);
                    sprintf(key3,"%c",tercero1);
                    outtextxy(DX-20,DY-20, key3);
                    sprintf(key3,"%c",cuarto1);
                    outtextxy(DX-30,DY-20, key3);
```

```
                                        sprintf(key3,"%c",quinto1);
                                        outtextxy(DX-40,DY-20, key3);


                                }
                        }
 }
 sali1:
//
ampligen1=ampligen;
if(senal==1 | senal==3)
{
ampligen = (ampligen / 0.037465116)/1.5;
outp(0xf3ff,ampligen);
//ap=amp;
//printf ("%x\n",ap);
}
if(senal==2)
{
ampligen = (ampligen / 0.037465116)/1.833;
outp(0xf3ff,ampligen);
//ap=amp;
//printf ("%x\n",ap);
}
        regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
 botg=0;
         goto  inicio;
break;

case 3:

        while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}
         salu:
 setfillstyle(1,0);
 bar(DX - 85,DY - 2,DX +8,DY + 8);
 veces=0,veces1=0,punto=0,cero=0,prue=0,mismo=0,diferente=0,primero1=0;
 segundo1=0,tercero1=0,cuarto1=0,quinto1=0,sexto1=0,septimo1=0,octavo1=0;
 noveno1=0,decimo1=0,primero=0,segundo=0,tercero=0,cuarto=0,quinto=0;
 sexto=0,septimo=0,octavo=0;
 setcolor(COLO);
 outtextxy(DX,DY,"0Hz");
 while(tec==0)
 {
```

```
while (bioskey(1) == 0);
{
        // function 0 returns the key that is waiting
        key = bioskey(0);
}

        // use function 2 to determine if shift keys were used
        // print out the character read
if (isdigit(key & 0xff  ))
{
                        if(key==0x5230)
                                decimal=0;

                        if(key==0x4f31)
                                decimal=1;

                        if(key==0x5032)
                                decimal=2;

                        if(key==0x5133)
                                decimal=3;

                        if(key==0x4b34)
                                decimal=4;

                        if(key==0x4c35)
                                decimal=5;

                        if(key==0x4d36)
                                decimal=6;

                        if(key==0x4737)
                                decimal=7;

                        if(key==0x4838)
                                decimal=8;

                        if(key==0x4939)
                                decimal=9;
        if(key==0x5230 & diferente!=1)   //SEGURIDAD POR SI EL CERO SE
                                                PRESIONA PRIMERO
                {
                        mismo=1;
                }
        if(key!=0x5230)  //LEDA EL PERMISO QUE SE IMPRIMA EL CERO
                        SOLO SI SE PRESIONO UN # DIREF. DE 0 Y EL "."
        {
                diferente=1;
                mismo=0;
        }
```

```
    if(punto==0 & (veces>=0 & veces <=5))
        {
            sexto = quinto;
            quinto = cuarto;
            cuarto = tercero;
            tercero = segundo;
            segundo = primero;
            primero = decimal;

        }
    if (punto==1 & (veces1>=0 & veces1<=1))
        {

            if( veces1==0)
                {
                prue=1;
                septimo=decimal;
                }
            if(veces1==1)
                octavo=decimal;
        }

        if(mismo==0)
        {
        cero++;
        veces++;
        veces1++;
        }
    }

if(key==0x5300| key==0x11b )  //TECLA DE SUPR Y ESC
    goto salu;

if(key==0x532e ) // TECLA DEL PUNTO
    {
        punto++;
        veces=0;
        veces1=0;
        mismo=0;
        diferente=1;

    }

    if(key==0x1c0d)  // TECLA DEL ENTER
        {
            setcolor(15);
            genf= primero + segundo*10 + tercero*100 +cuarto *1000
            +quinto*10000 +sexto* 100000 +septimo/10+octavo/100;
            if(genf>=524000.00)
```

```
                    {
                      regs.x.ax=0x2;
                      int86(0x33,&regs,&regs);
                      setfillstyle(1,0);
                      bar(DX - 85,DY-3 ,DX+6 ,DY +10);
                      setcolor(COLO);

                      outtextxy(DX-80,DY,"5");
                      outtextxy(DX-70,DY,"2");
                      outtextxy(DX-60,DY,"4");
                      outtextxy(DX-50,DY,"0");
                      outtextxy(DX-40,DY,"0");
                      outtextxy(DX-30,DY,"0");
                      outtextxy(DX-20,DY,".");
                      outtextxy(DX-10,DY,"0");
                      outtextxy(DX,DY,"0");
                      genf=524000;
                      regs.x.ax=0x1;
                      int86(0x33,&regs,&regs);
                    }

if(genf >= 9 & genf <= 88)
{
genc = 10.96e-6;
resl=44.9e-06;
if(genf >= 20 )
resl=45.15e-06;
if(genf >= 30 )
resl=45.35e-06;
if(genf >= 40 )
resl=45.8e-06;
if(genf >= 50 )
resl=46.2e-06;
if(genf >= 60 )
resl=46.5e-06;
if(genf >= 70 )
resl=46.8e-06;
if(genf >= 80 )
resl=47.0e-06;

Y5w= Y5w & 0xa3;
                //        Q7          Q6       Q5     Q4   Q3   Q2    Q1        Q0
Y5w = Y5w | 0x24; //0010 0100  |  ERROR GEN | cc5=0/c4=1 | LD/da | c3 | c2 | c1 |
square | trg=0/sen=1 |
outp(0xf3fd, Y5w);
}

if(genf >= 89 & genf <= 818)
{
genc = 1.187e-06;
resl= 45.2e-06;
```

```
if(genf >= 150 )
resl=45.5e-06;
if(genf >= 200 )
resl=45.7e-06;
if(genf >= 250)
resl=45.85e-06;
if(genf >= 300)
resl=46.1e-06;
if(genf >= 350)
resl=46.2e-06;
if(genf >= 400)
resl=46.35e-06;
if(genf >= 450)
resl=46.5e-06;
if(genf >= 500)
resl=46.65e-06;
if(genf >= 550)
resl=46.75e-06;
if(genf >= 600)
resl=46.85e-06;
if(genf >= 650)
resl=46.93e-06;
if(genf >= 700)
resl=47.0e-06;
if(genf >= 750)
resl=47.05e-06;
if(genf >= 800)
resl=47.1e-06;
Y5w= Y5w & 0xa3;
                    //        Q7          Q6      Q5    Q4   Q3   Q2    Q1        Q0
Y5w = Y5w | 0x28; //0010 0100  |  ERROR GEN | cc5=0/c4=1 | LD/da | c3 | c2 | c1 |
square | trg=0/sen=1 |
//Y5w = 0x28; //0010 1000
outp(0xf3fd, Y5w);
}
if(genf >= 819 & genf <= 7.15e3)
{
genc = 125.0e-09;
resl= 44.69e-06;
if(genf >= 2e3)
resl=44.77e-06;
if(genf >= 3e3)
resl=44.85e-06;
if(genf >= 5e3)
resl=44.95e-06;
Y5w= Y5w & 0xa3;
                    //        Q7          Q6      Q5    Q4   Q3   Q2    Q1        Q0
Y5w = Y5w | 0x30; //0010 0100  |  ERROR GEN | cc5=0/c4=1 | LD/da | c3 | c2 | c1 |
square | trg=0/sen=1 |
//Y5w = 0x30; //0011 0000
outp(0xf3fd, Y5w);
```

```
}
if(genf >= 7.151e3 & genf <= 65e3)
{
genc = 12.23e-09;
resl= 44.22e-06;
if (genf >= 9e3)
resl=44.07e-06;
if (genf >= 12e3)
resl=44.02e-06;
if (genf >= 14e3)
resl=43.92e-06;
if (genf >= 14e3)
resl=43.77e-06;
if (genf >= 20e3)
resl=43.71e-06;
if (genf >= 23e3)
resl=43.65e-06;
if (genf >= 26e3)
resl=43.5e-06;
if (genf >= 34e3)
resl=43.3e-06;
if (genf >= 40e3)
resl=43.15e-06;
if (genf >= 45e3)
resl=43.03e-06;
if (genf >= 50e3)
resl=42.9e-06;
if (genf >= 55e3)
resl=42.56e-06;
if (genf >= 60e3)
resl=42.48e-06;
Y5w= Y5w & 0xa3;
                    //        Q7          Q6      Q5    Q4  Q3  Q2    Q1        Q0
Y5w = Y5w | 0x60; //0010 0100  |  ERROR GEN | cc5=0/c4=1 | LD/da | c3 | c2 | c1 |
square | trg=0/sen=1 |
//Y5w = 0x60; //0110 0000
outp(0xf3fd, Y5w);
}
if(genf > 65e3 & genf <= 524e03)
{
genc = 1.5e-09;
resl= 45.25e-06;
if (genf >= 180e3)
resl=45.03e-06;
if (genf >= 230e3)
resl=44.8e-06;
if (genf >= 280e3)
resl=44.43e-06;
if (genf >= 330e3)
resl=44.26e-06;
if (genf >= 370e3)
```

```
resl=43.99e-06;
if (genf >= 400e3)
resl=43.73e-06;
if (genf >= 430e3)
resl=43.52e-06;
if (genf >= 460e3)
resl=43.15e-06;
if (genf >= 480e3)
resl=42.7e-06;
if (genf >= 500e3)
resl=42.45e-06;
if (genf >= 515e3)
resl=42.0e-06;
Y5w = 0x20; //0010 0000
outp(0xf3fd, Y5w);
}


genn= - ( (((((genf * 9700 * genc) - 1) * 1218) / 9700) + 1) * 3) /resl ;

F1 = genn & 0x00ff;
F2 = genn & 0xff00;
F2 = F2/0x100;
outp(0xf3fb,F1);
outp(0xf3fc,F2);
Y5w = Y5w & 0xdf;   //1101 1111
outp(0xf3fd, Y5w);
Y5w = Y5w | 0x20;
outp(0xf3fd, Y5w);

                        goto salu1;
                       }
                        if(key==0x532e  & punto==2) //TECLA DEL PUNTO SE SEA
                                                PRECIONADO DEMACIADAS VECES

                        if(prue!=1)
                        {
                        segundo1=0;
                        primero1=0;
                        veces=0;
                        veces1=0;
                        punto=1;
                        }

 if(punto==1 & veces==0 & cero==0 )
        primero1=0x5230;
 if (isdigit(key & 0xff  ) | key==0x532e )
        if( punto==0 & (veces>=0 & veces <=6) |  punto==1 & (veces1>=0 & veces1<=2))
                {
                        if(mismo==0)
                                {       regs.x.ax=0x2;
```

```
int86(0x33,&regs,&regs);
noveno1= octavo1;
octavo1 =septimo1;
septimo1 =sexto1;
sexto1 = quinto1;
quinto1 = cuarto1;
cuarto1 = tercero1;
tercero1 = segundo1;
segundo1 = primero1;
primero1 = key;
setcolor(0);
if(veces==1)
outtextxy(DX,DY,"0");
sprintf(key3,"%c",decimo1);
outtextxy(DX-80,DY, key3);
sprintf(key3,"%c",noveno1);
outtextxy(DX-70,DY, key3);
sprintf(key3,"%c",octavo1);
outtextxy(DX-60,DY, key3);
sprintf(key3,"%c",septimo1);
outtextxy(DX-50,DY, key3);
sprintf(key3,"%c",sexto1);
outtextxy(DX-40,DY, key3);
sprintf(key3,"%c",quinto1);
outtextxy(DX-30,DY, key3);
sprintf(key3,"%c",cuarto1);
outtextxy(DX-20,DY, key3);
sprintf(key3,"%c",tercero1);
outtextxy(DX-10,DY, key3);
sprintf(key3,"%c",segundo1);
outtextxy(DX,DY, key3);
setcolor(COLO);
sprintf(key3,"%c",primero1);
outtextxy(DX,DY, key3);
sprintf(key3,"%c",segundo1);
outtextxy(DX-10,DY, key3);
sprintf(key3,"%c",tercero1);
outtextxy(DX-20,DY, key3);
sprintf(key3,"%c",cuarto1);
outtextxy(DX-30,DY, key3);
sprintf(key3,"%c",quinto1);
outtextxy(DX-40,DY, key3);
sprintf(key3,"%c",sexto1);
outtextxy(DX-50,DY, key3);
sprintf(key3,"%c",septimo1);
outtextxy(DX-60,DY, key3);
sprintf(key3,"%c",octavo1);
outtextxy(DX-70,DY, key3);
sprintf(key3,"%c",noveno1);
outtextxy(DX-80,DY, key3);
sprintf(key3,"%c",decimo1);
```

```
                                        outtextxy(DX-90,DY, key3);
                                        regs.x.ax=0x1;
                                        int86(0x33,&regs,&regs);
                                }
                        }
 }
 salu1:
  regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
 botg=0;
         goto  inicio;
 break;

   }

         //tester

        if(bott==1)

     { I=0;
         V=0;
    R+=1;

        while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}     regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
        bott=0;
        //setcolor(15);
        setfillstyle(1,15);
        bar(70,380,280,455);
        if(R==2)
    {   seltest= seltest & 0x9f;  //10011111
        outp(0xf3f1,seltest);
        delay(10);
        RESC= RESC &  0xdf;          //11011111
        outp(0xf3f2,RESC);
        R=0;
        mc4066 = mc4066 & 0xfd;   //11111101
        outp(0xf3f3,mc4066);

    }
        goto inicio;

    }

    if(bott==2)
   {
```

```
        R=0;
        V=0;
        I+=1;
        //setcolor(15);
        setfillstyle(1,15);
        bar(70,380,280,455);
        while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}     regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
        bott=0;
        if(I==2)
     {
        seltest= seltest & 0x9f;  //10011111
        outp(0xf3f1,seltest);
        I=0;
        mc4066 = mc4066 & 0xfe;    //11111110
        outp(0xf3f3,mc4066);


     }
        goto inicio;

    }
    if(bott==3)
   { R=0;
        I=0;
        V+=1;
        //setcolor(15);
        setfillstyle(1,15);
        bar(70,380,280,455);
        while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}      regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
        if(V==2)
        V=0;
        mc4066= mc4066 & 0xf7;     //11110111
        outp(0xf3f3, mc4066);
        bott=0;
        goto inicio;
    }
    if(bott ==4)
   { ac+=1;
        adctest1=1;
```

```
        //setcolor(15);
        setfillstyle(1,15);
        bar(70,380,280,455);
    if(ac==2)
     ac=0;

while(mb == 1)
{
        regs.x.ax=0x3;
        int86(0x33,&regs,&regs);
        mb = 0x1 & regs.x.bx;
}     regs.x.ax=0x2;
        int86(0x33,&regs,&regs);
         bott=0;
        goto inicio;

   }
         goto salto;

 /*LECTURA DEL MOUSE*/
 regs.x.ax =0x01;
 int86 (0x33, &regs, &regs);
  quit:
  closegraph();
  return 0;
}
```

CAPITULO 5

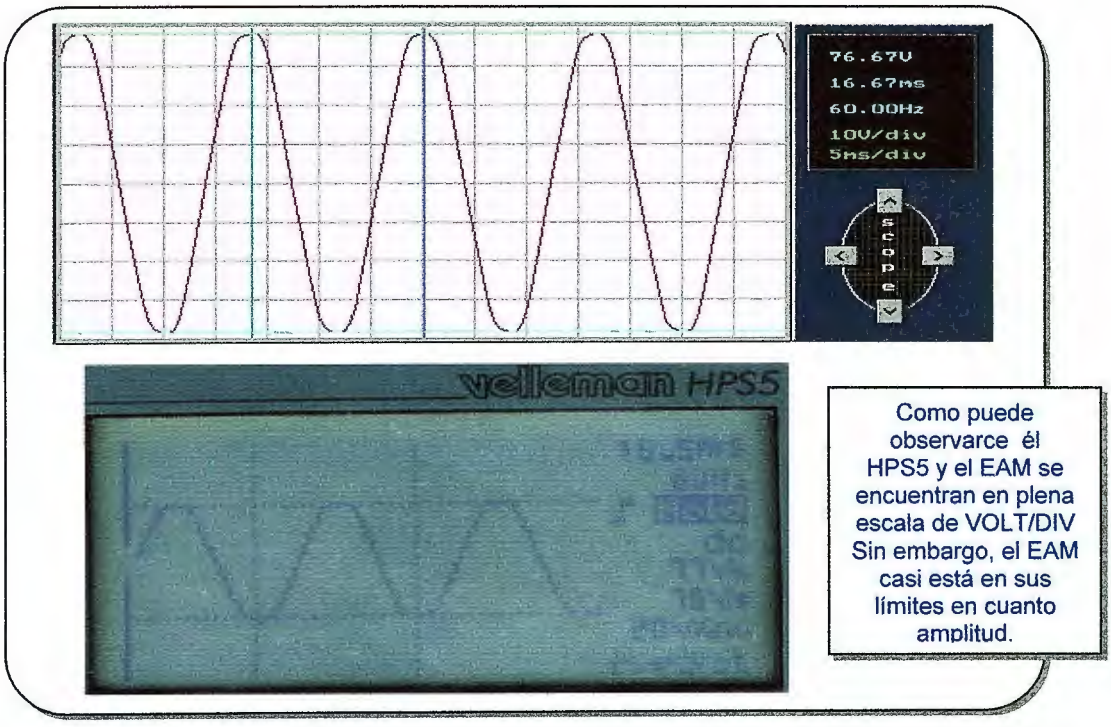
RESULTADOS OBTENIDOS Y SUS POSIBLES MEJORAS.

Pruebas y Análisis de resultado:

A continuación se presenta el análisis de resultado efectuado al equipo auxiliar de medición controlado por computadora. Las pruebas presentadas a continuación fueron efectuadas con diferentes equipos de medición de prestigio, para comprobar y garantizar que se trata de un equipo de medición que puede ser competitivo en cierta medida y al mismo tiempo recalcar cuales son los puntos fuertes y débiles del mismo.
Es necesario comprender que cuando se trata de equipos de instrumentación diferentes estos son diseñados utilizando diversas tácticas, procedimientos y al momento de tomar mediciones poseen un margen de error que los diferencia.

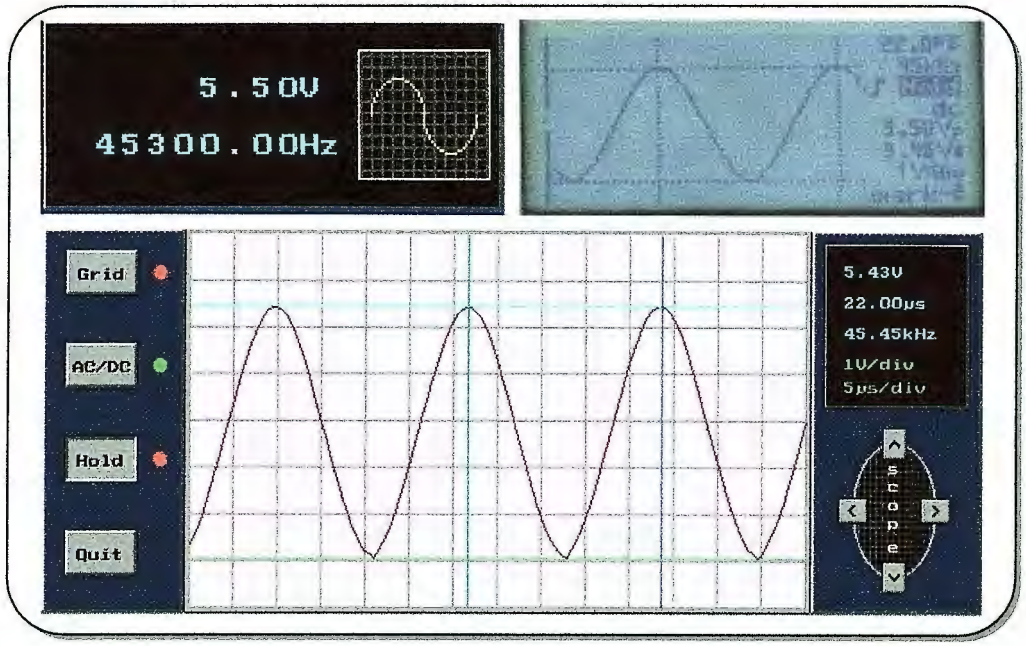
1- En las siguientes pruebas se midieron diferentes señales comprobando el funcionamiento del modulo del osciloscopio y generador de funciones del EAM y se compararon con el osciloscopio personal HPS5 (de 5MHZ), los resultados fueron los siguientes:

- Señal alterna tomada de un transformador de 24VRMS

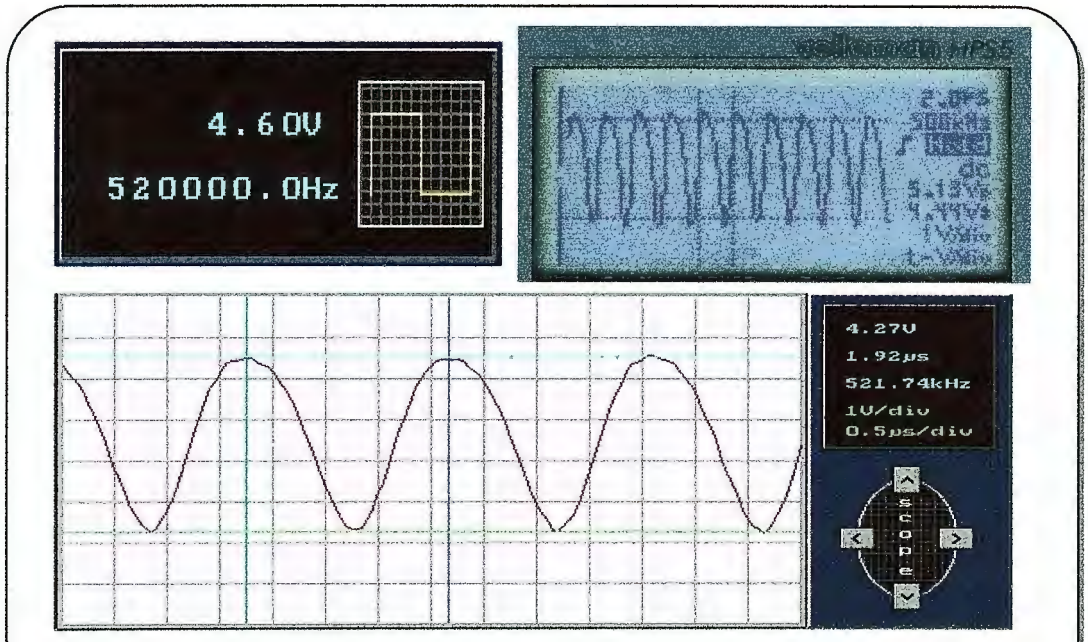


Como puede observarce él HPS5 y el EAM se encuentran en plena escala de VOLT/DIV Sin embargo, el EAM casi está en sus límites en cuanto amplitud.

- Señal senoidal lograda por el GEN – FUN del EAM.

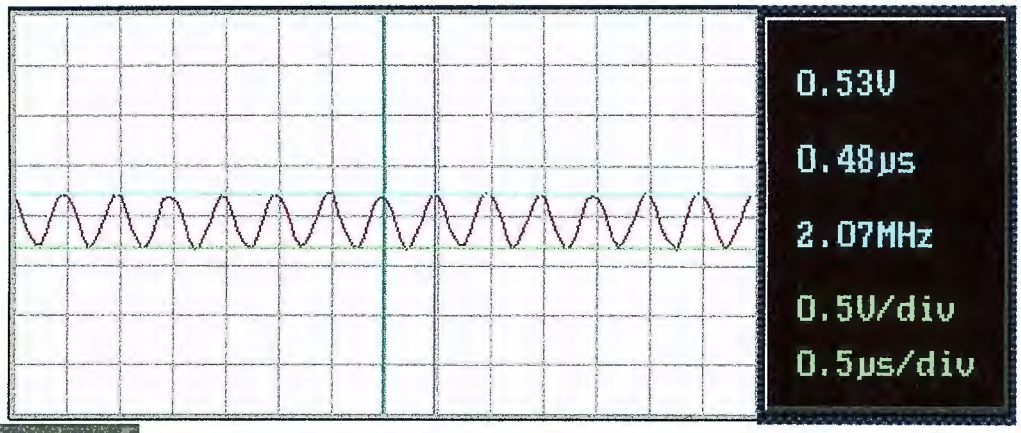


- Señal cuadrada lograda por el GEN – FUN del EAM.

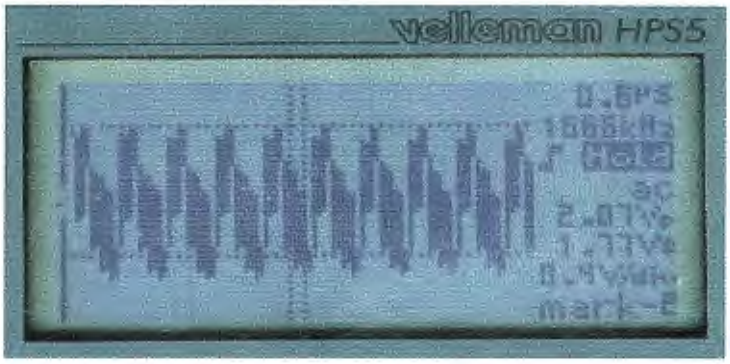


Las imágenes muestran claramente que el GEN-FUN se encuentra en plena escala de frecuencia mientras que ambos osciloscopios están en la máxima escala de TIME/DIV.

- Señal TTL obtenida de un divisor de frecuencia.

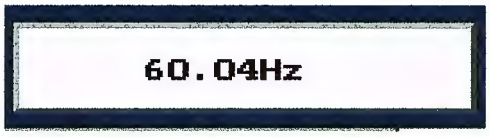


La señal de entrada posee una frecuencia de 2MHZ nuevamente ambos osciloscopios están en la máxima escala de TIME/DIV. Nótese que la señal se ve atenuada pero es mucho más apreciable en el EAM que en el HPS5.



2- A continuación se pone a prueba el frecuencímetro del EAM y se compara con medidor de frecuencia del multímetro digital CHIEF ENGINEER modelo S.M.S.

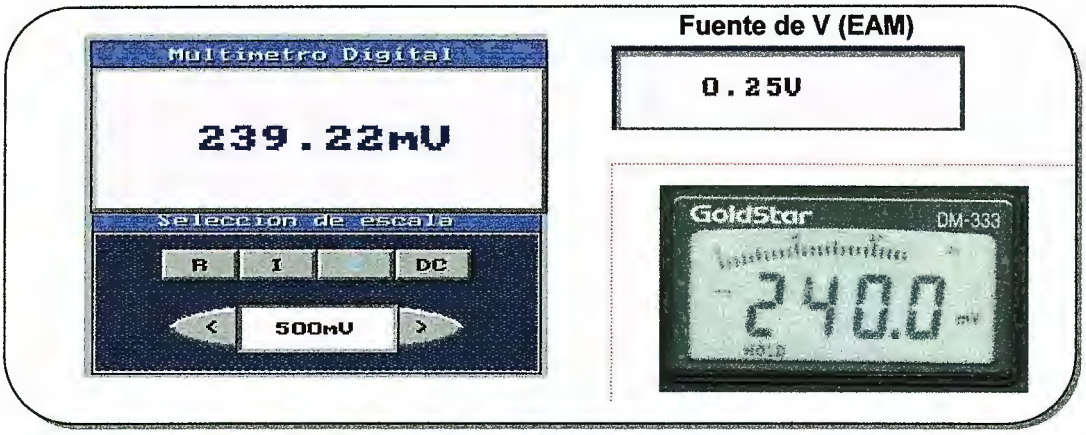
- Señal alterna proporcionada por un transformador de 24V RMS.

S.M.S	EAM frecuencímetro
	

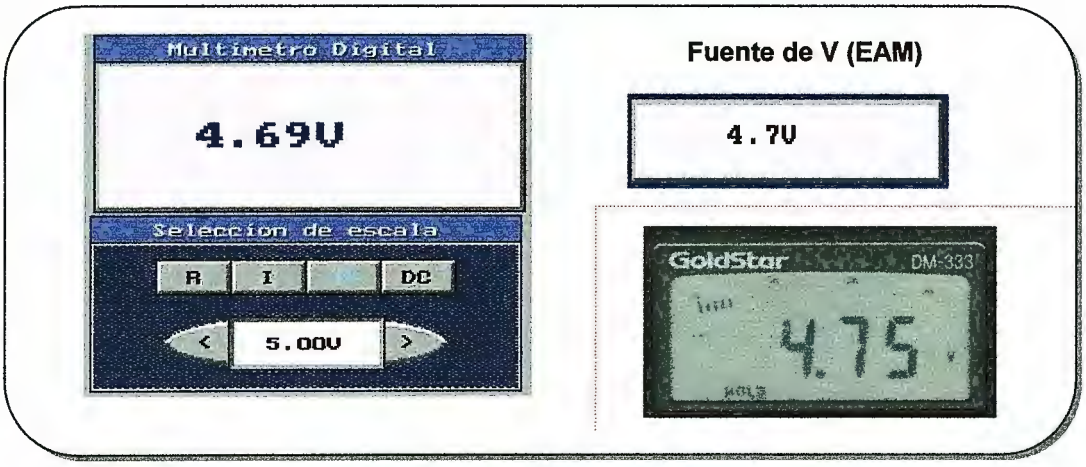
- Señal cuadrada lograda por el GEN-FUN.

3- En las siguientes pruebas se verifico el funcionamiento de la fuente de voltaje y el multímetro del EAM, estos resultados son comparados con un TESTER Golstard modelo DM-333 (autorango) y multímetro digital CHIEF ENGINEER modelo S.M.S :

- Voltaje DC de 0.25V generado por la fuente de voltaje del EAM.



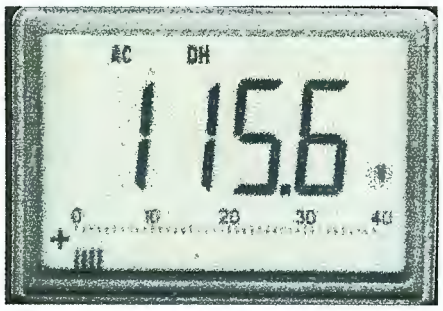
- Voltaje DC de 4.7V generado por la fuente de voltaje del EAM.



- Voltaje DC de 12.00V generado por la fuente de voltaje del EAM.

- Voltaje tomado de la red de suministro AC de 110VRMS / 60HZ.

CHIEF ENGINEER S.M.S

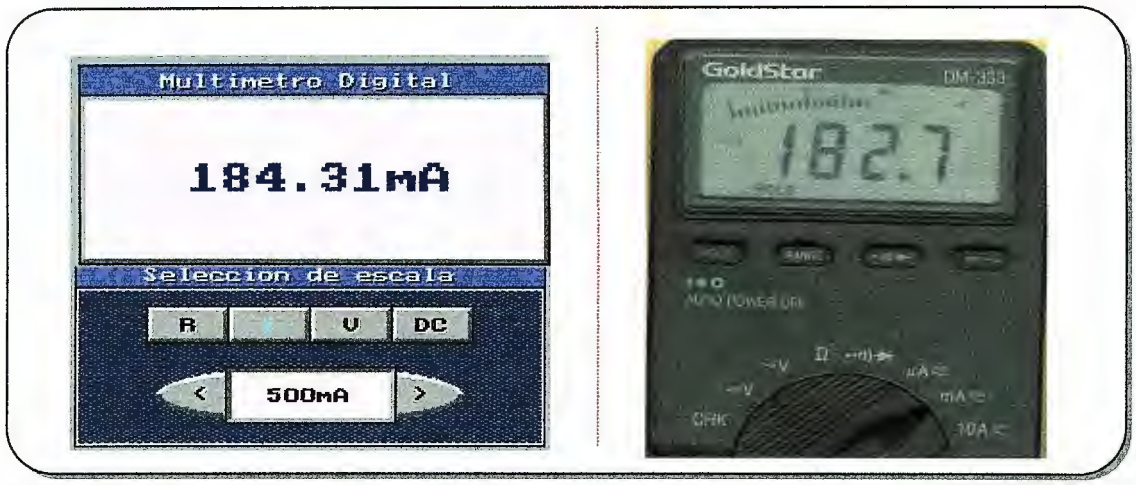


Escala de 400V

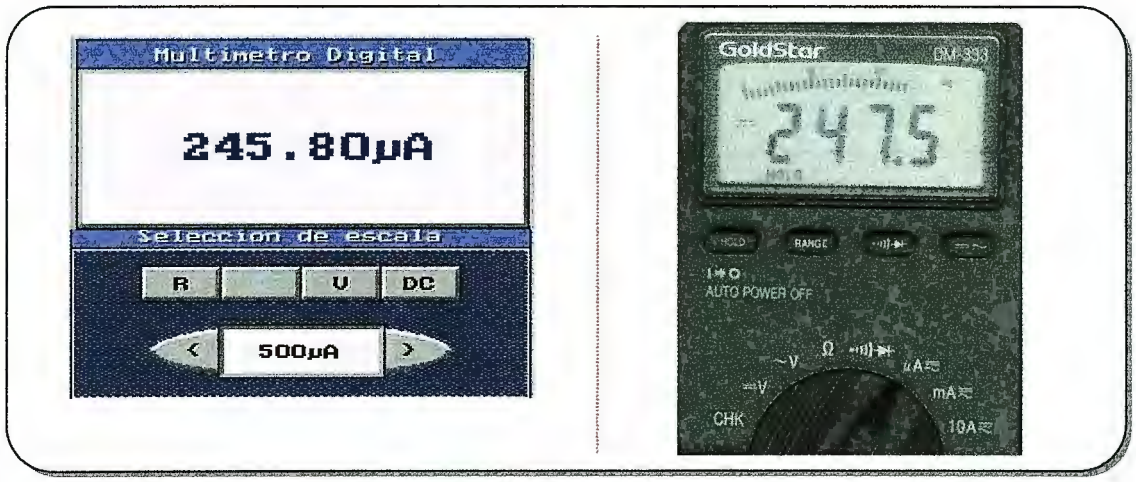
- Corriente AC medida de un transformador.

- Corriente AC medida de un transformador.

- Corriente DC medida de la fuente de voltaje del EAM.



- **Corriente DC medida de la fuente de voltaje del EAM.**



- **Resistencia de 20Ω(5% tolerancia) medida con el óhmetro del EAM.**

- **Resistencia de 2KΩ (2% tolerancia) medida con el óhmetro del EAM.**

- **Resistencia de 2MΩ(2% tolerancia) medida con el óhmetro del EAM.**

- **Corriente medida de la fuente del EAM.**

Escala de 10A

Todas las mediciones y pruebas se efectuaron en tiempo real y se tomaron en cuenta diferentes instrumentos. Como sé a podido apreciar los instrumentos presentan pequeñas diferencias que son completamente normales; sin embargo, estamos totalmente convencidos que se pueden considerar algunas mejoras futuras para el EAM.

Recomendaciones y mejoras.

Frecuencímetro.

A continuación se presentan una serie de recomendaciones para poder mejorar el frecuencímetro:

1) Cambiar el operacional de entrada por un operacional que presente mayor velocidad de respuesta, para que logre medir señales de mayor frecuencia.
2) Agregar una entrada para que pueda medir señales exclusivas de integrados TTL y CMOS a una mayor frecuencia (20MHz).
3) El frecuencímetro puede ser capaz de medir señales con offset si a este se le agrega un capacitor en serie a la entrada.
4) Si se desea aumentar la frecuencia máxima a medir se puede cambiar el cristal de 20MHz por uno mas alto, ya que de éste depende la frecuencia máxima a medir.
5) El frecuencímetro se puede diseñar para que logre medir voltajes muy pequeños, esto se puede hacer con solo aplicar una pequeña ganancia en la etapa de entrada.

Fuente de voltaje.

Las siguientes recomendaciones son para mejorar en un futuro la fuente de voltaje:

1) La fuente de voltaje puede ser rediseñada para que pueda proporcionar voltajes positivos y negativos (fuente bipolar)
2) Que entregue una corriente mayor.
3) La protección de la fuente se puede hacer programable, el usuario puede programar la corriente a la cual se tiene que proteger, esto se puede hacer comparando la corriente de salida con el valor de la corriente de protección.
4) Cambiando el DAC de 8 bits por uno de 16 bits se puede obtener un voltaje a la salida con una mayor resolución.

Recomendaciones para poder calibrar el generador de funciones.

1) Se debe calibrar la salida STO del XR2206 para que de 1Vpp, primero se ajusta la salida de la señal triangular con el potenciómetro R25 y después la señal senoidal con el potenciómetro R27 (por necesitar una resistencia mas baja, para lograr este voltaje que la señal senoidal).
2) Con el osciloscopio conectado a la salida:

   - Se debe ajustar el potenciómetro R29 para reducir la distorsión de la señal senoidal; se debe variar hasta lograr una señal adecuada.

   - Se tiene que ajustar el ciclo de trabajo hasta que se pueda observar que el tiempo en alto es igual al tiempo en bajo, esto se logra variando el potenciómetro R32.

- El potenciómetro R28 se debe ajustar hasta que la señal pase en el centro del osciloscopio. (Con este potenciómetro se agrega o se elimina el offset de la señal.)

NOTA: EL OSCILOSCOPIO TIENE QUE ESTAR EN ACOPLE DC.

3) La corriente que circula a través del pin TR1 se debe calibrar de la siguiente manera:
   - Se debe de enviar el máximo valor de frecuencia de cualquier escala para que el convertidor tenga 0V. a la salida; entonces, se debe variar el potenciómetro R18 hasta que el voltaje del pin 7 del operacional U4B sea igual al del pin GND del XR2206 (-8.3V).

   - Por ultimo se debe enviar el valor mínimo de frecuencia de cualquier escala para que el convertidor tenga su máximo voltaje a la salida y ahora se debe variar el potenciómetro R13 hasta que el voltaje del pin 7 del operacional U4B tenga -5.3V. para lograr la diferencia de 3V.

Sugerencia para mejorar el funcionamiento GEN – FUN.

- Cambiar la fuente de alimentación de +8.3v y -8.3v por un regulador de voltaje para lograr una mejor estabilidad en la señal.

- En un futuro se puede agregar un DC offset programable por software.

- Si se quiere mejorar y simplificar completamente el módulo del generador de funciones se recomienda cambiar el VCO XR2206 por uno mas potente como es el caso del MAX038.

Ajuste de las escalas para el multímetro y posibles mejoras.

El ajuste del voltímetro se logra con cada resistencia de escala y utilizando la IG, se coloca un Vin para la escala que se desea calibrar luego se ajusta la resistencia de escala hasta lograr un valor idéntico al que se tiene a la entrada del voltímetro.
En el caso del amperímetro la calibración es similar a la del voltímetro y siempre se realiza en DC (sí se realiza una buena calibración en DC esta misma calibración se refleja en AC). Sin embargo se debe tener especial cuidado a la hora de calibrar la protección ya que una mala calibración podría oaccionar serias consecuencias y lo más seguro es que el fusible auxiliar se queme y sé tendría que sacar la tarjeta para cambiarlo (en el peor de los casos).
Para el óhmetro es necesario medir el valor óhmico de cada resistencia de escala, así como también el voltaje de VCC (+5V) del divisor del voltaje, la otra variable de la ecuación seria el voltaje del divisor Vx la cual es obtenida a trabes del ADC, finalmente la única incógnita es Rtes, si se tomaron bien las medidas de las variables mencionadas el valor de resistencia que se mide será exacto.
Es necesario tomar en cuenta que el ADC que se utiliza es de 8 bits, si aumentamos la capacidad del ADC se lograría una mayor resolución lo cual implica una mayor exactitud en las mediciones.

Para toda calibración se recomienda buscar variables eléctricas de prueba que sean intermedias a la escala que se desea calibrar, ya que si se hace en los puntos extremos de la escala no se lograra una adecuada calibración y puede que se aumente el margen de error en la medición.

Ajuste de las escalas para el osciloscopio.

La calibración de las escalas de VOLT/DIV se debe realizar en acople DC para permitir que la señal presente todas sus características, de preferencia se puede utilizar una señal constante cuya magnitud dependerá de la escala seleccionada, se recomienda utilizar la ganancia que corresponde a cada escala para la calibración, introduzca un voltaje que sea intermedio a la escala que se desea calibrar y verifique por HARDWARE que se cumple la ecuación general en el punto VD:

$$1VP / VPE\ (max) = \triangle T$$

Es muy importante que se satisfaga la condición de entrada análoga para el LTC1406 en caso de que exista descalibración. Lo cual se puede ajustar por medio de R31 pero después de calibrar las escalas VOLT/DIV, con el objetivo de lograr el rango de voltaje adecuado para el ADC (1.5V - 3.5V).
Sí se realiza una buena calibración en las escalas de VOLT/DIV esta misma calibración se refleja en la IG ya que las escalas de TIME/DIV son constantes predefinidas y están sincronizadas con el programa.

El osciloscopio puede ser mejorado en ciertas etapas como por ejemplo:

A. Implementar un disparo controlado por software en donde la señal de entrada se comience a presentar en pantalla hasta que alcance la magnitud que el usuario seleccione en la IG.

B. Se podría utilizar el mismo osciloscopio para realizar diversos cálculos de la señal de forma automática (Vdc, VRMS, VP, VPP, Vprom y hasta realizar cálculos en db).

C. Además de la opción HOLD, se podría lograr congelar una señal para luego continuar apreciando otra con el fin de comparar condiciones entre ambas señales.

D. En un futuro se podría mejorar la IG para agregar controles que pudieran manipular por SOFTWARE la posición de la señal para que esta se pueda desplazar en la pantalla.

E. El programa se podría mejorar de forma increíble al utilizar un lenguaje de programación más potente (visual Basic), pero los requerimientos de la PC para instalar el programa aumentarían y para propósitos de comercialización no estaría al alcance de ciertos sectores del mercado.

F. La velocidad del osciloscopio puede ser incrementada al eliminar el número de muestras que se realizan por cada división, el problema es que la calidad de la señal se puede ver afectada, ya que el programa contará con

menos puntos para formar la gráfica. Pero si se realiza un algoritmo de interpolación efectivo; esta condición se podría trasformar en una buena alternativa para aumentar la velocidad del sistema.

Expansión del EAM.

El EAM es un prototipo inicial y se puede mejorar en un futuro tanto en la parte de software, así como también en hardware por el momento se a logrado dejar una base pensando en su posible expansión:

A. Se ha logrado conectar un sólo bus en donde viajan líneas de control, de dato y otro donde viajan líneas de alimentación. Estos conectores están disponibles para que puedan ser conectados otros instrumentos auxiliares de medición.

B. El diseñador puede retirar la tarjeta que desea mejorar y utilizar las mismas líneas de control de la tarjeta anterior para la nueva, y en caso de ser necesario, se puede utilizar las líneas sobrantes de la interfaz para completar el instrumento.

C. El sistema de protección se puede vincular con la PC de tal forma que sirva en los laboratorios como método de evaluación y control para los estudiantes, para lograrlo se tendrían que realizar mejoras en el programa y en el EAM.

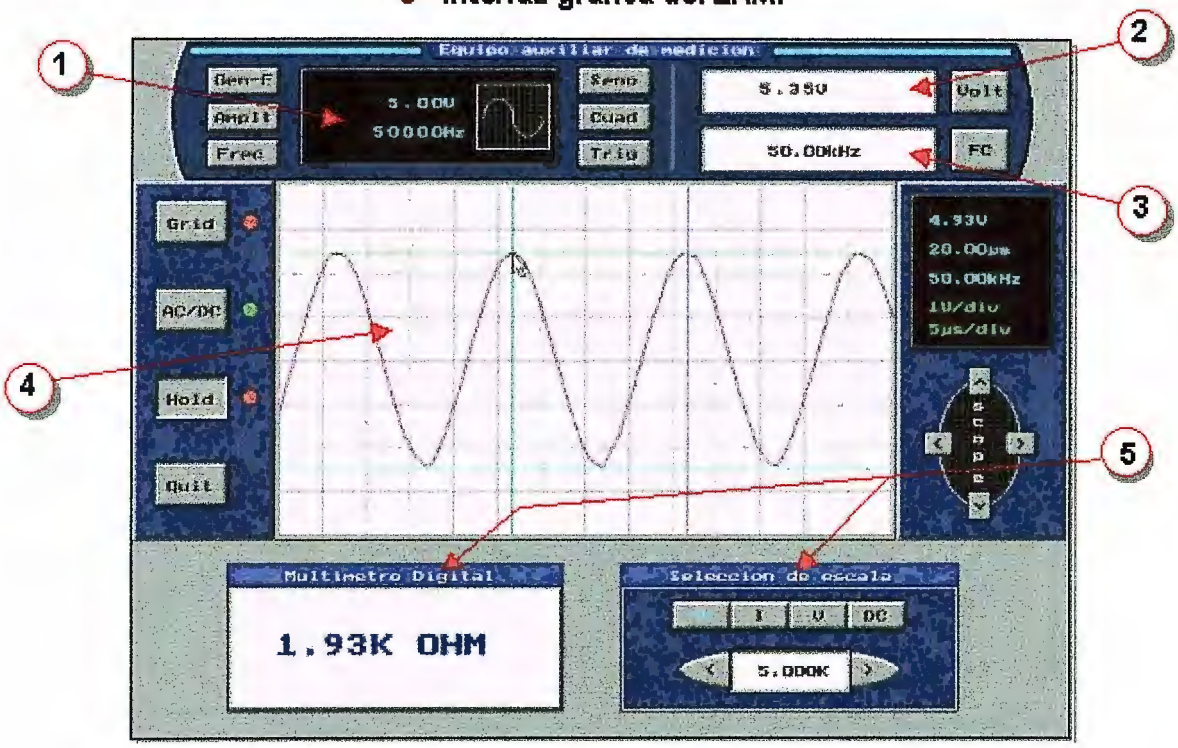
D. Si se quiere mejorar el funcionamiento por individual y en conjunto de las tarjetas se recomienda realizar las pistas con una mayor área de contacto para GND y VCC. Para que exista una mejor conducción y aumentar la eficiencia del equipo.

E. En el EAM se incluyen 2 dispositivos montados sobre superficie (SSOP). Si se quiere reducir el tamaño del prototipo se recomienda utilizar en su mayoría estos dispositivos.

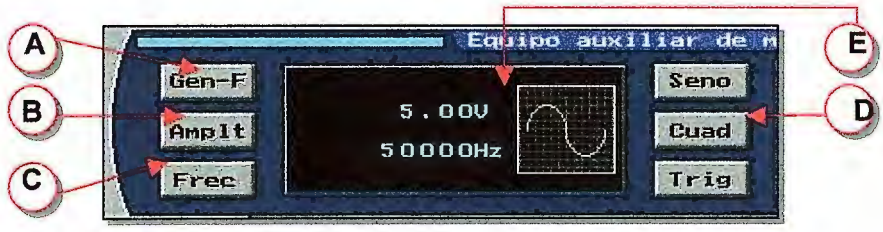
Uso de la interfaz gráfica del EAM

En la PC se encuentra alojado el sistema operativo para manipular y controlar todos los instrumentos del EAM. La manipulación del programa es muy sencilla y práctica, desde un mismo menú se tiene acceso a todos los dispositivos. A continuación se describen las opciones del menú de la IG y sus funciones.

Interfaz gráfica del EAM:



1- Generador de funciones (Gen-Fun):



A: Este botón sirve para quitar la protección cuando se ocasiona un cortocircuito o se sobrepasa un determinado valor de corriente.

B: Con este botón se puede ingresar el valor de amplitud (VPP) de la función a generar, el máximo valor permisible es de 15VPP, en cuyo caso se coloque un valor mayor que 15Vpp el programa lo ignora y presenta automáticamente el máximo valor que puede proporcionar.

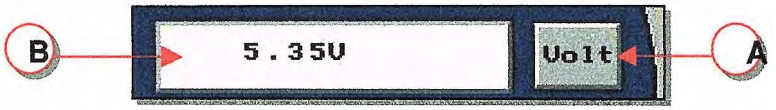
C: Al presionar este botón el programa esperará que el usuario ingrese el valor de frecuencia que desea, este valor debe ser ingresado en HZ.

D: *Selección de señal.*
Utilice estos botones para seleccionar el tipo de señal de la función a generar.

E: P*antalla de presentación del GEN - FUN.*

En esta pantalla se presentan los valores de amplitud, frecuencia y tipo de señal que se desea, estos valores pueden ser modificados en forma independiente y tienen la ventaja de poseer auto escala.

2- Fuente de voltaje:



A: Este Botón posee dos funciones la primera es para quitar la protección de la fuente de voltaje cuando esta sé a protegido, la segunda es para ingresar el valor de voltaje que el usuario desea (0V -12V). Si el usuario digita un valor mayor a 12V el programa coloca automáticamente el máximo valor.

B: *Pantalla de la fuente de voltaje:*

Esta pantalla presenta el valor de voltaje que el usuario desea y también presenta un mensaje de "corto en la fuente" cuando esta se protege.

Nota: Utilice el teclado de la computadora para ingresar números a la IG en cuyo caso se equivoque al digitar un valor determinado presione la tecla SUPR o ESC para borrar él número erróneo y finalmente presione INTRO para darle efectividad al valor.

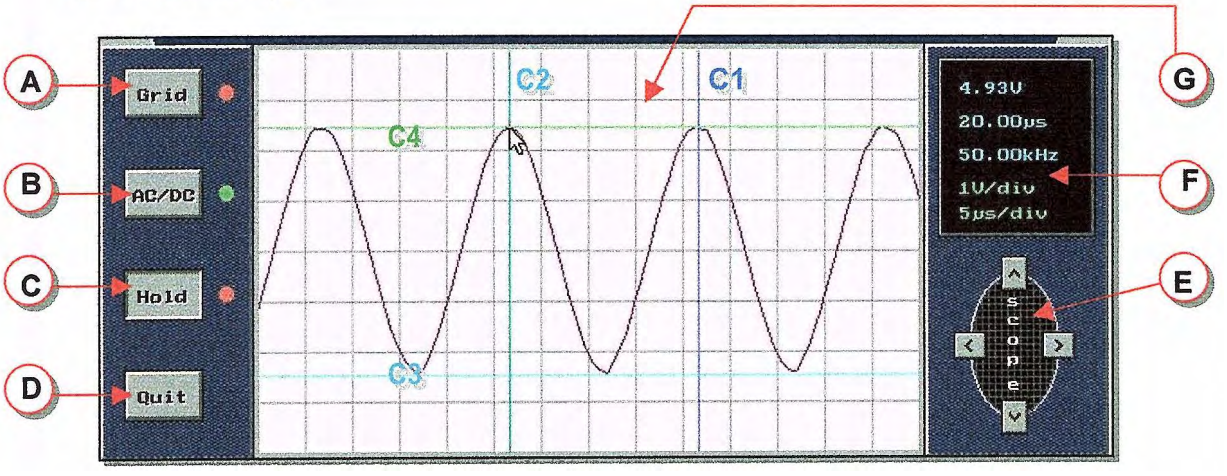
3- frecuencímetro:

A: La función de este botón no es efectiva para este programa, pero tiene aplicaciones futuras en caso que se desee implementar un tipo de protección inteligente.

B: *Pantalla del frecuencímetro.*

El valor de frecuencia que sé este midiendo se presenta de forma permanentemente en esta pantalla.

4- Osciloscopio:

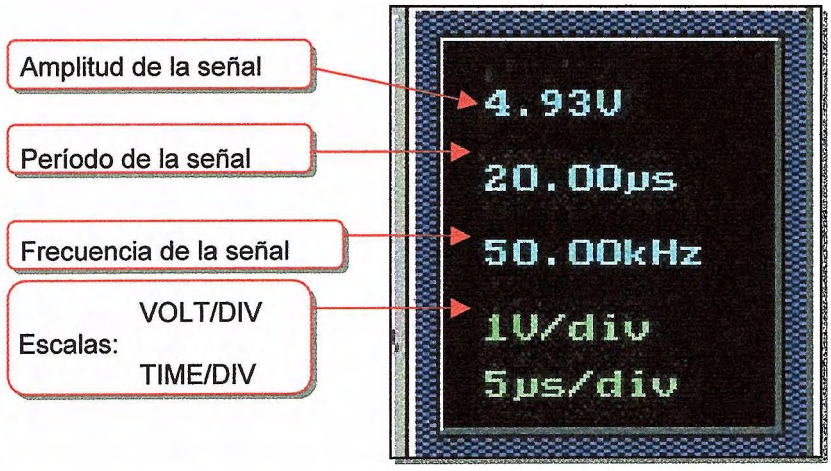


A: *GRID.*
Utilice este botón para colocar la cuadricula del osciloscopio o eliminarla.
B: *Acople de señal.*
En condiciones normales el programa ha seleccionado el acople DC, al presionarlo se cambia por acople AC.
C: *HOLD.*
Este botón se utiliza para congelar la señal que sé esta observando en pantalla y así tener acceso a las cotas y poder determinar características importantes de la señal bajo medición.

Manejo de las cotas en función HOLD.

Después de congelar la señal en pantalla presione el clic derecho del Mouse para llamar a una de las cuatro cotas (C1 - C4), una vez seleccionada la cota a utilizar coloque el ratón en el punto que usted desee que aparezca y luego presione nuevamente el clic izquierdo; en ese momento la cota seleccionada aparecerá en pantalla. Para cambiar de posición la cota se puede arrastrarla con el Mouse o bien colocar el mouse en un punto cualquiera y presionar el clic izquierdo para que la cota aparezca en el punto deseado. Si desea llamar a otra de las cotas repita el procedimiento anterior hasta lograr que las cuatro cotas aparezcan en la pantalla (las cotas pueden ser usadas de forma individual, pero siempre será necesario combinarlas). Las cotas C1 y C2 sirven para determinar la frecuencia de la señal mientras que C3 y C4 para calcular la amplitud.

D: Salir del programa (todas las características almacenadas se pierden).
E: *Cursoras de escalas TIME/DIV y VOLT DIV.*
Utilice las cursoras verticales para controlar las escalas de VOLT/DIV y las cursoras horizontales para seleccionar las escalas de TIME/DIV.
F: *Pantalla de control del osciloscopio.*
Esta pantalla es de esencial importancia para visualizase diversas características como se presentan a continuación:



G: *Pantalla de presentación del osciloscopio.*
En esta pantalla el usuario puede apreciar la señal que se mide, así como también tomar mediciones atraves de las cotas.

- Multímetro:

Ventana de selección de escala e instrumento:

A: *Óhmetro, Amperímetro y voltímetro.*
Utilice los botones **R**, **I** y **V** para seleccionar el dispositivo que se desea utilizar, tanto el programa como el EAM sólo permiten que se pueda seleccionar un instrumento a la vez.

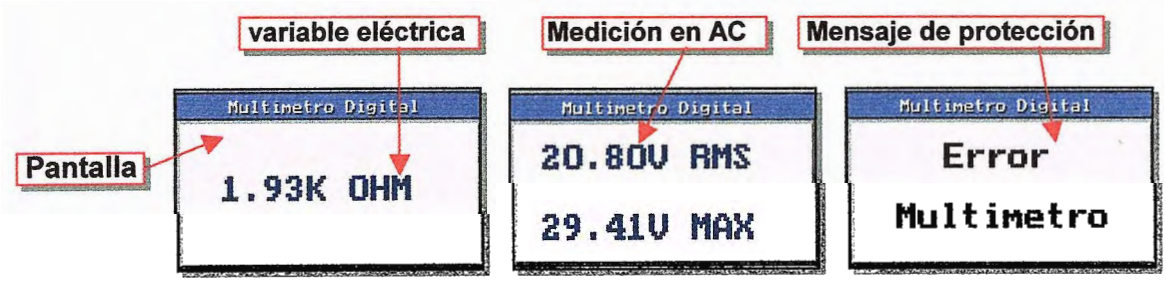
B: *AC.*
Utilice este botón si se desea medir una corriente o voltaje alterno.

C: *Escalas.*
Las cursoras de escalas son empleadas para cambiar las escalas del instrumento que se está utilizando *(Óhmetro, Amperímetro y voltímetro).*

D: *Pantalla de escala.*
Sirve para visualizar la escala seleccionada para cada instrumento con sus respectivas unidades.

Ventana de presentación del multímetro.

En esta pantalla se puede apreciar el valor de la variable eléctrica que se está midiendo y también despliega un mensaje de "Error en el mulíimetro", que indica que el amperímetro se ha protegido. Cuando el programa ha detectado un error en el manejo del amperímetro presenta el mensaje correspondiente y bloquea el programa, hasta que el usuario presione con el clic izquierdo en las coordenadas de la pantalla del multímetro (se asume que el error a sido retirado).
Cuando el amperímetro y voltímetro se encuentran en AC la IG calcula el valor máximo y RMS de la variable eléctrica y lo presenta en pantalla de forma simultanea. En cuyo caso el usuario se equivoque al medir una señal DC, en AC, el programa presentará cero, lo mismo en caso contrario (para lograrlo el programa revisa el signo que proviene del EAM).



BIBLIOGRAFIA

- RONALD J. TOCCI
  SISTEMAS DIGITALES PRINCIPIOS Y APLICACIONES
  6ª EDICION
  EDITORIAL PRENTICE HALL

- JOHN F. WAKERLY.
  DISEÑO DIGITAL, PRINCIPIOS Y PRACTICAS.
  EDITORIAL PRENTICE HALL
  MEXICO 1992

- ROBERT F. COUGHLIN, FREDERICK F. DRISCOLL
  AMPLIFICADORES OPERACIONALES Y CIRCUITOS INTEGRADOS LINEALAES
  CUARTA EDICION
  EDITORIAL PRENTICE HALL

- OPERATING MANUAL
  INSTRUMENTATION UNIT PU-2200
  DEGEM SYSTEMS

- BARRY B. BREY
  LOS MICROPROCESADORES INTEL
  3ª EDICION
  EDITORIAL PRENTICE HALL

- NATIONAL SEMICONDUCTOR.
  NATIONAL DATA ADQUISITION
  DATA BOOK
  USA 1995

- WWW. DIGI- KEY. COM

Presupuesto del proyecto.

A continuación se presenta el presupuesto del anteproyecto para que sirva de parámetro de comparación con el presupuesto final del proyecto. Es importante recalcar que en el presupuesto final no se tomó en cuenta la PC.

Presupuesto del anteproyecto

ITEM	ELEMENTO	CANTIDAD	PRECIO/ U (¢)	PRECIO (¢)
1	Computadora 80486	1	3000	3000
2	Generador de funciones (max038)	1	400	400
3	Convertidor D/A	3	150	450
4	Convertidor A/D	1	200	200
5	Registros	7	15	105
6	Multiplexor/Demultiplexor analógico	4	50	200
7	Buffer de tres estados	4	15	60
8	Compuertas TTL	9	15	135
9	IC operacionales	6	15	90
10	Resistencias	25	1	25
11	Capacitores	10	2	20
12	Memorias RAM	1	60	60
13	IC contadores TTL	3	15	45
14	Terminales	10	1.25	12.5
15	Puntas de prueba	10	5	50
16	Tabletas cobreadas	6	10	60
17	Multiplcador analógico	1	200	200
18	estaño	1	10	10
19	Pasta	1	5	5
20	Bases para IC	6	20	120
21	Alambres	1	30	30
22	Fuente de voltaje	1	175	175
22	Interruptor controlado por V.	3	15	45
			TOTAL	5497.5

Los gastos presentados en esta tabla pueden variar, debido a que se toman en cuenta aquellos materiales y elementos más importantes que creemos que pueden ser utilizados; pero como no se cuenta con un diseño completo estos gastos pueden aumentar o disminuir en pequeñas proporciones por tener en este momento sólo un diagrama a bloques. Los precios han sido consultados con algunos proveedores, ventas de repuestos electrónicos y reflejan un presupuesto hipotético que estimamos conveniente.

Presupuesto del proyecto

ITEM	ELEMENTO	CANTIDAD	PRECIO/ U (¢)	PRECIO (¢)
1	XR2206	1	60	60
2	Convertidor D/A (LTC1597)	1	481.25	481.25
3	Convertidor D/A (DAC0808)	2	30	60
4	Convertidor A/D (LTC1406)	2	61.25	122.5
5	Registros	12	12	144
6	Multiplexor/Demultiplexor analógico	1	70	70
7	Buffer de tres estados	4	15	60
8	Compuertas TTL	16	10	160
9	IC operacionales(LT1356, LT1358)	4	74.4	297.6
10	IC operacionales (LM324, LM074, LM358)	14	10	140
11	Multiplicador análogo(AD633)	1	300	300
12	FLIP-FLOP	6	10	60
13	Relés	24	19	456
14	Potenciometros	32	5	160
15	Transistores	34	3	102
16	Resistencias	179	0.5	89.5
17	Capacitores	135	1.5	202.5
18	Memorias RAM	1	150	150
19	IC contadores TTL	9	8	72
20	Terminales	10	1.25	12.5
21	Puntas de prueba	10	5	50
22	Diodos	51	2	102
23	Cinchas	1	20	20
24	Diodos Led	4	1.75	7
25	Porta Led	3	2	6
26	Cristales	3	60	180
27	DB25	2	18	36
28	Puente de diodos	1	18	18
29	Disipadores	7	10	70
30	Estaño	1	25	25
31	Pasta	1	5	5
32	Bases para IC	102	5	510
33	Chasis	1	150	150
34	Alambres	1	30	30
35	Papelería y acetatos	1	550	550
36	Circuito impresos	8	60	480
37	Fusibles	2	3	6
38	Ventilador	1	100	100
39	Transformador	1	100	100
40	Reguladores de voltaje	7	15	105

TOTAL	5749.85

Cronograma de actividades.

A continuación se presenta el Cronograma del anteproyecto para que sirva de parámetro de comparación con el Cronograma final del proyecto. La demora infortuita más grande que se tuvo es la demora de los integrados que mandamos a traer al extranjero, sin embargo la complejidad del proyecto fue también otro factor que provoco atrasos en el mismo.

Cronograma del anteproyecto

ACTIVIDAD	1er MES				2do MES				3er MES			
	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem
DISEÑO DE INTERFAZ DE COMUNICACION MONTAJE EN BREADBOARD												
DISEÑO DEL OHMETRO MONTAJE EN BREADBOARD												
DISEÑO DE VOLTIMETRO DC MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO DC MONTAJE EN BREADBOARD												
DISEÑO DE VOLTIMETRO AC MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO AC (ma) MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO AC (A) MONTAJE EN BREADBOARD												
DISEÑO DEL FRECUENCIMETRO MONTAJE EN BREADBOARD												
DISEÑO DE LA FUENTE DE VOLTAJE MONTAJE EN BREADBOARD												
DISEÑO DEL GENERADOR DE FUNCIONES MONTAJE EN BREADBOARD												

ACTIVIDAD	4to MES				5to MES				6to MES			
	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem
DISEÑO DEL OSCILOSCOPIO MONTAJE EN BREADBOARD												
DISEÑO DE LA PROTECCION DEL EQUIPO MONTAJE EN BREADBOARD												
ELABORACION DEL PROGRAMA												
ELABORACION DE CTKOS. IMPRESOS												
ENSAMBLE DE ELEMENTOS EN PISTAS												
ELEABORACION DE PRUEBAS Y CORRECCIONES												
DEFENSA DEL PROYECTO												
Elaboración del documento (durante los 6 meses)												

Cronograma del proyecto

ACTIVIDAD	1er MES				2do MES				3er MES			
	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem
DISEÑO DE INTERFAZ DE COMUNICACIÓN MONTAJE EN BREADBOARD												
DISEÑO DE LA FUENTE DE VOLTAJE MONTAJE EN BREADBOARD												
DISEÑO DEL FRECUENCIMETRO MONTAJE EN BREADBOARD												
INVESTIGACIÓN DE INTEGRADOS A UTILIZAR												
ENCARGO DE INTEGRADOS AL EXTRANJERO												
INVESTIGACIÓN DE INTEGRADOS A UTILIZAR												
ENCARGO DE INTEGRADOS AL EXTRANJERO												
ESPERA DE INTEGRADOS												

ACTIVIDAD	4to MES				5to MES				6to MES			
	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem
ESPERA DE INTEGRADOS												
DISEÑO DEL GENERADOR DE FUNCIONES MONTAJE EN BREADBOARD												
DESEÑO DEL OSCILOSCOPIO MONTAJE EN BREADBOARD												
DISEÑO DEL OHMETRO MONTAJE EN BREADBOARD												

ACTIVIDAD	7mo MES				8vo MES				9no MES			
	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem	1ra sem	2da sem	3ra sem	4ta sem
DISEÑO DE VOLTIMETRO DC MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO DC MONTAJE EN BREADBOARD												
DISEÑO DE VOLTIMETRO AC MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO AC (ma) MONTAJE EN BREADBOARD												
DISEÑO DE AMPERIMETRO AC (A) MONTAJE EN BREADBOARD												
ELABORACION DEL PROGRAMA												
ELABORACION DE CTKOS. IMPRESOS												
ENSAMBLE DE ELEMENTOS EN PISTAS												

ACTIVIDAD	10mo MES			
	1ra sem	2da sem	3ra sem	4ta sem
ENSAMBLE DE ELEMENTOS EN PISTAS				
ELEABORACION DE PRUEBAS Y CORRECCIONES				
DEFENSA DEL PROYECTO				

Nota: Elaboración del documento durante la implementación del proyecto.

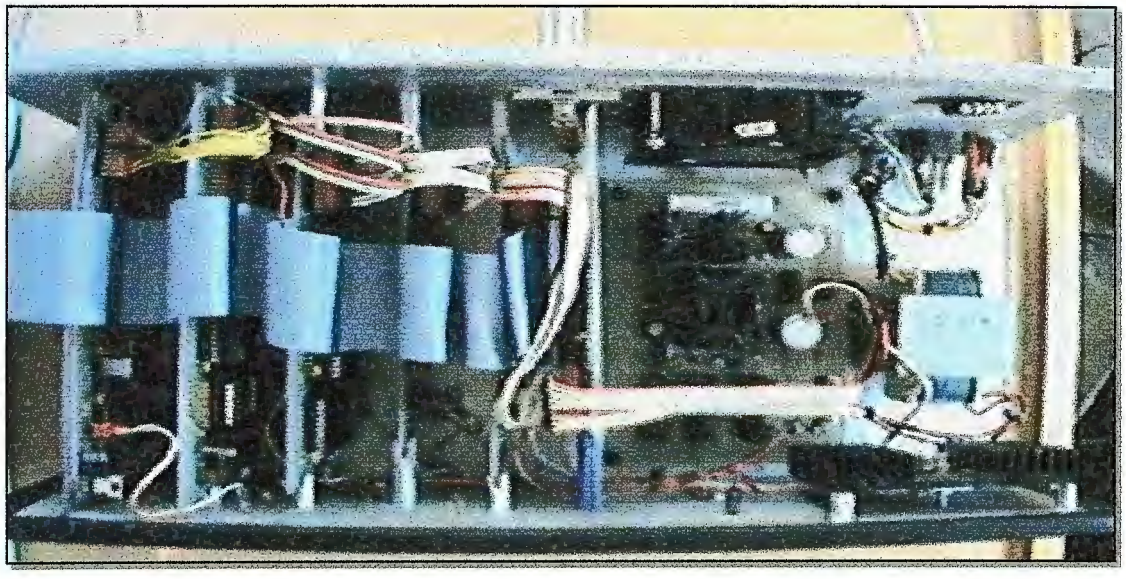
Fotografías tomadas del EAM.

A continuación se presentan diferentes fotografías tomadas al EAM para tener un panorama gráfico del aparato, sus dimensiones y sus diferentes dispositivos que lo conforman.

Fotografía frotal del equipo auxiliar de medición.

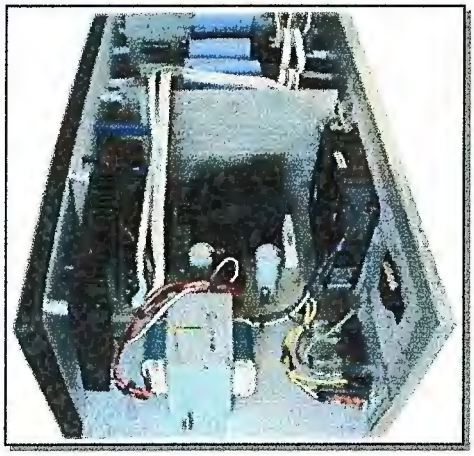
Las dimenciones para la pate frontal del equipo son las siguientes:
49cm largo, 17 cm altura y 22cm profundidad.

Fotografía aérea del equipo auxiliar de medicion.



Fotografía posterior del EAM.

En esta foto se puede apreciar el conector para el puerto de la interfaz de comunicación del EAM, así como también: la conección a la red de alimentación, el interruptor principal de ensendido, el fusible de protección general y el sistema de ventilación para el equipo.

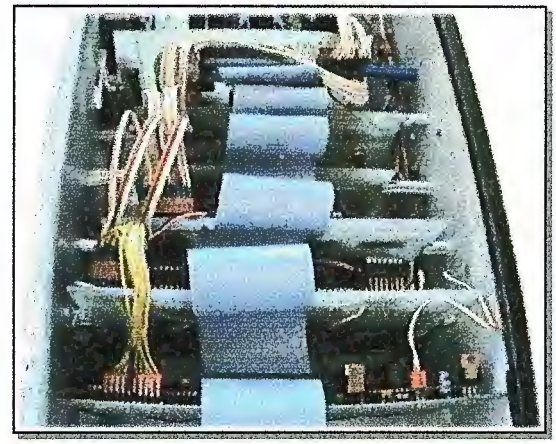


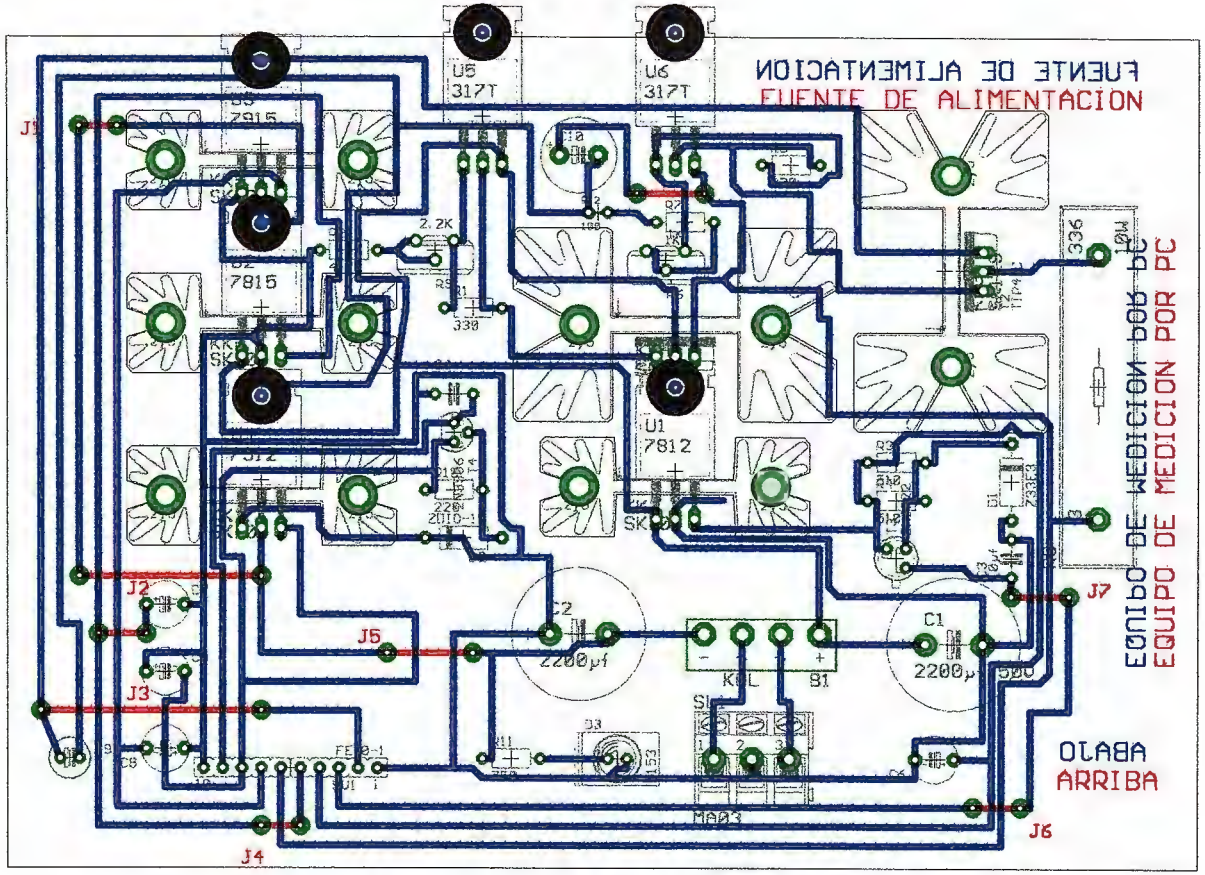
fotografía lateral derecha del EAM.

En esta foto se puede apreciar el sistema de conección eléctrica del EAM y la fuente de alimentación que suministra potencia a todos los instrumentos del EAM.

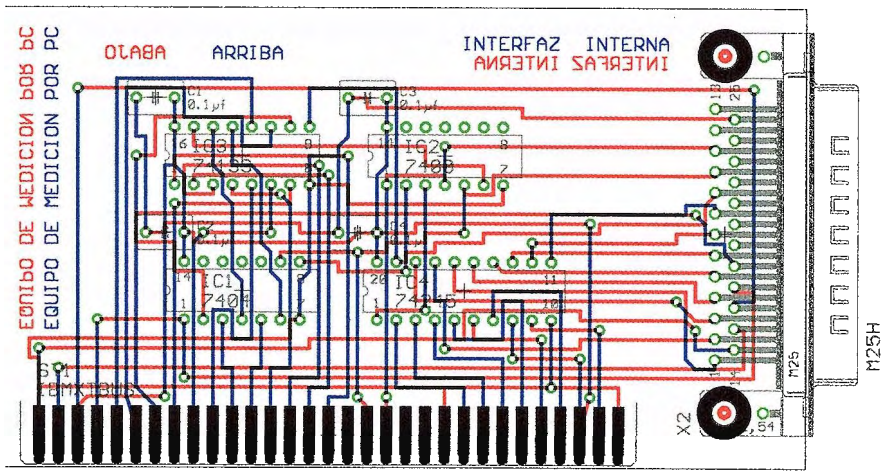
Fotografia lateral izquierda del EAM.

Es aquí donde se encuentran todas las tarjetas del equipo, éstas están colocadas en rieles y pueden ser desmontables para ajustes y reparación, el orden de posición de cada tarjeta corresponde con la posición de cada instrumento en el panel frontal.

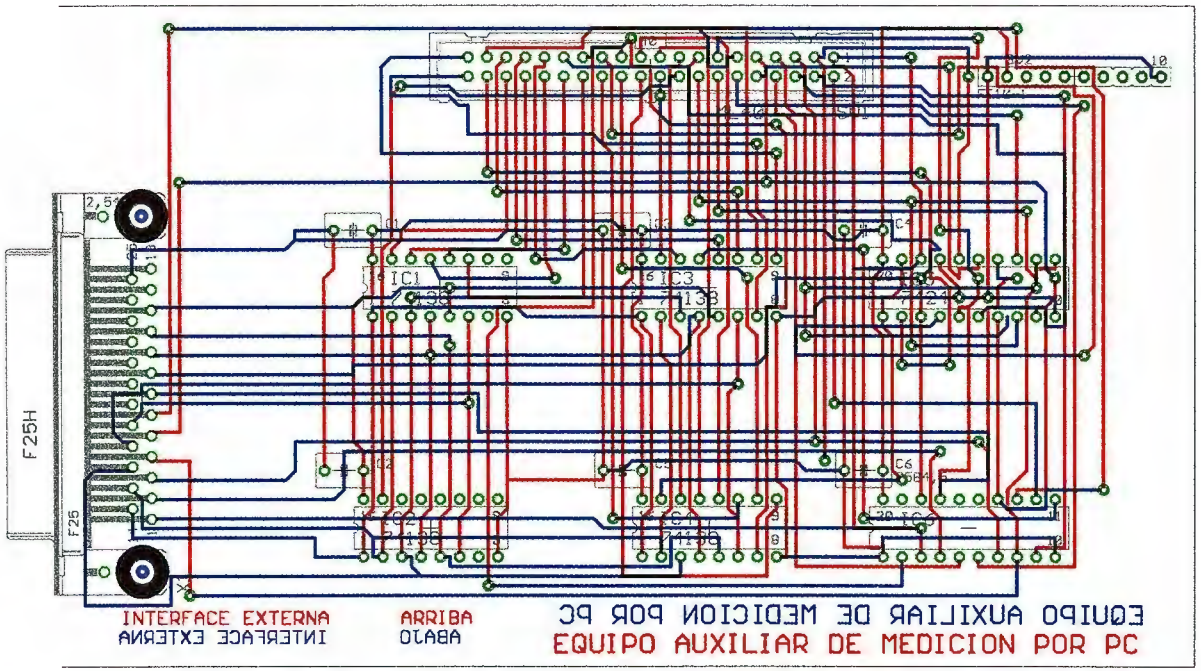




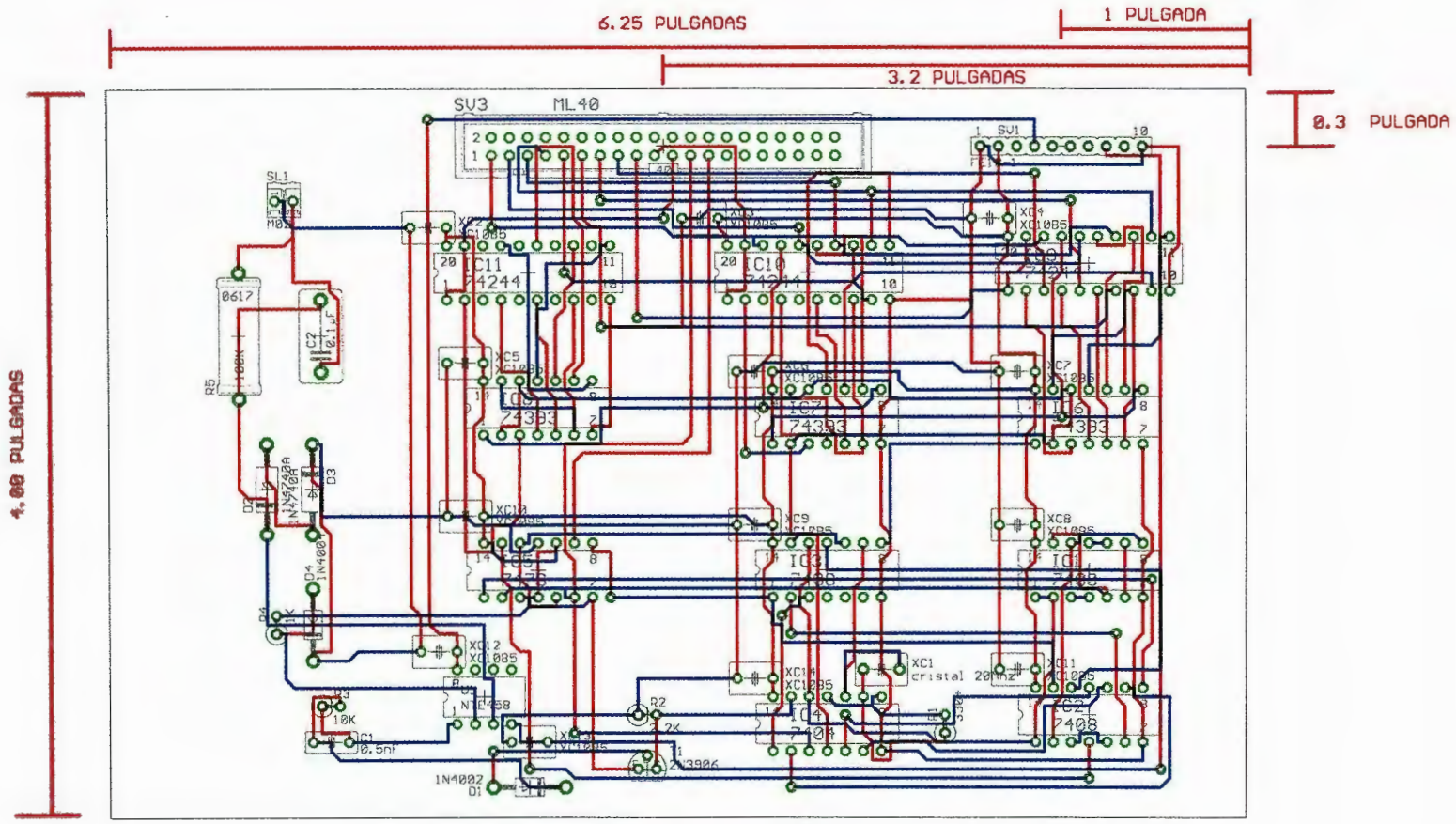
Fuente de alimentación.



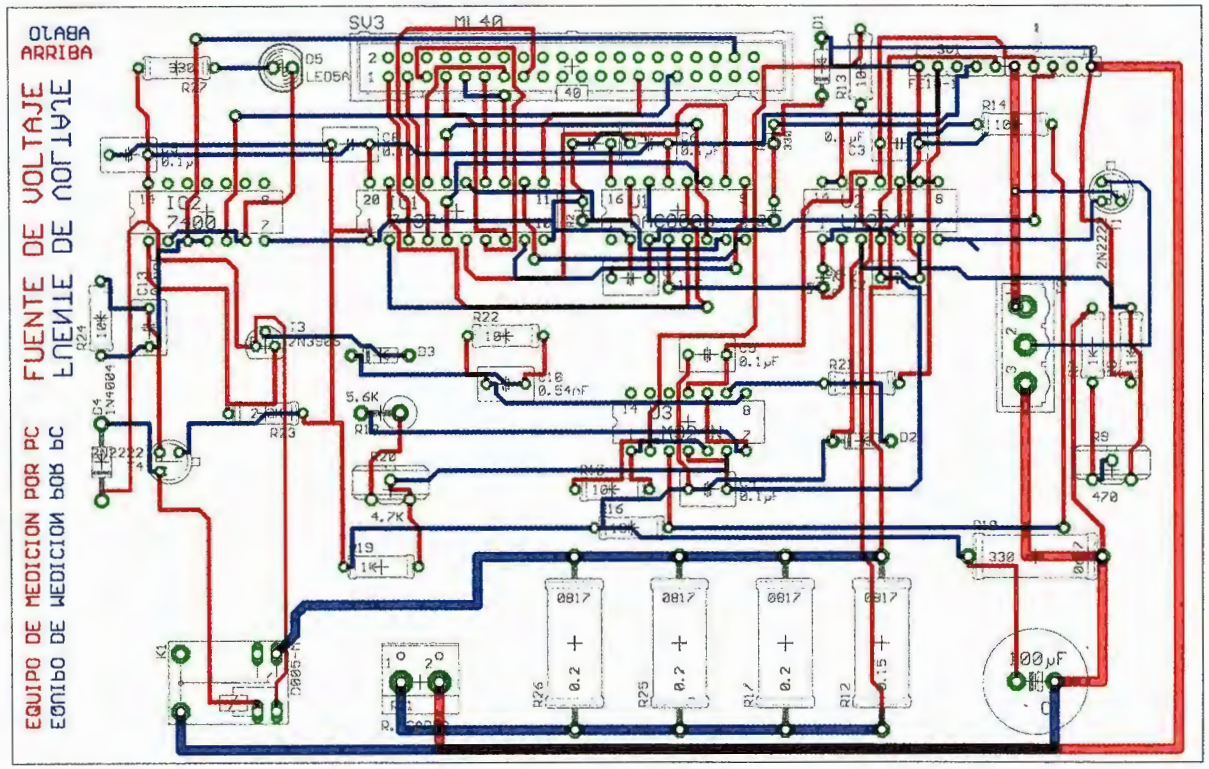
Interfaz interna.



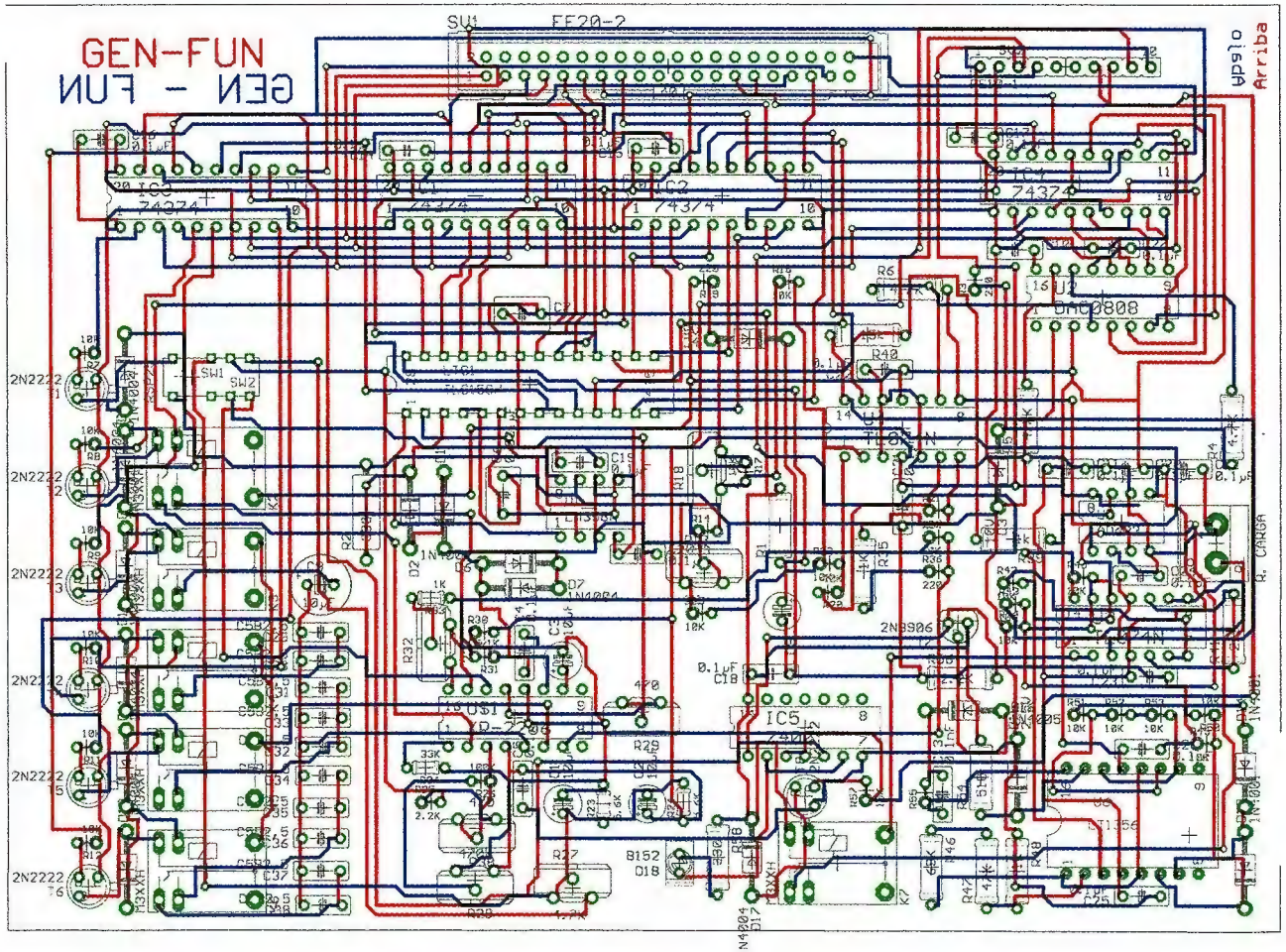
Interfaz externa.

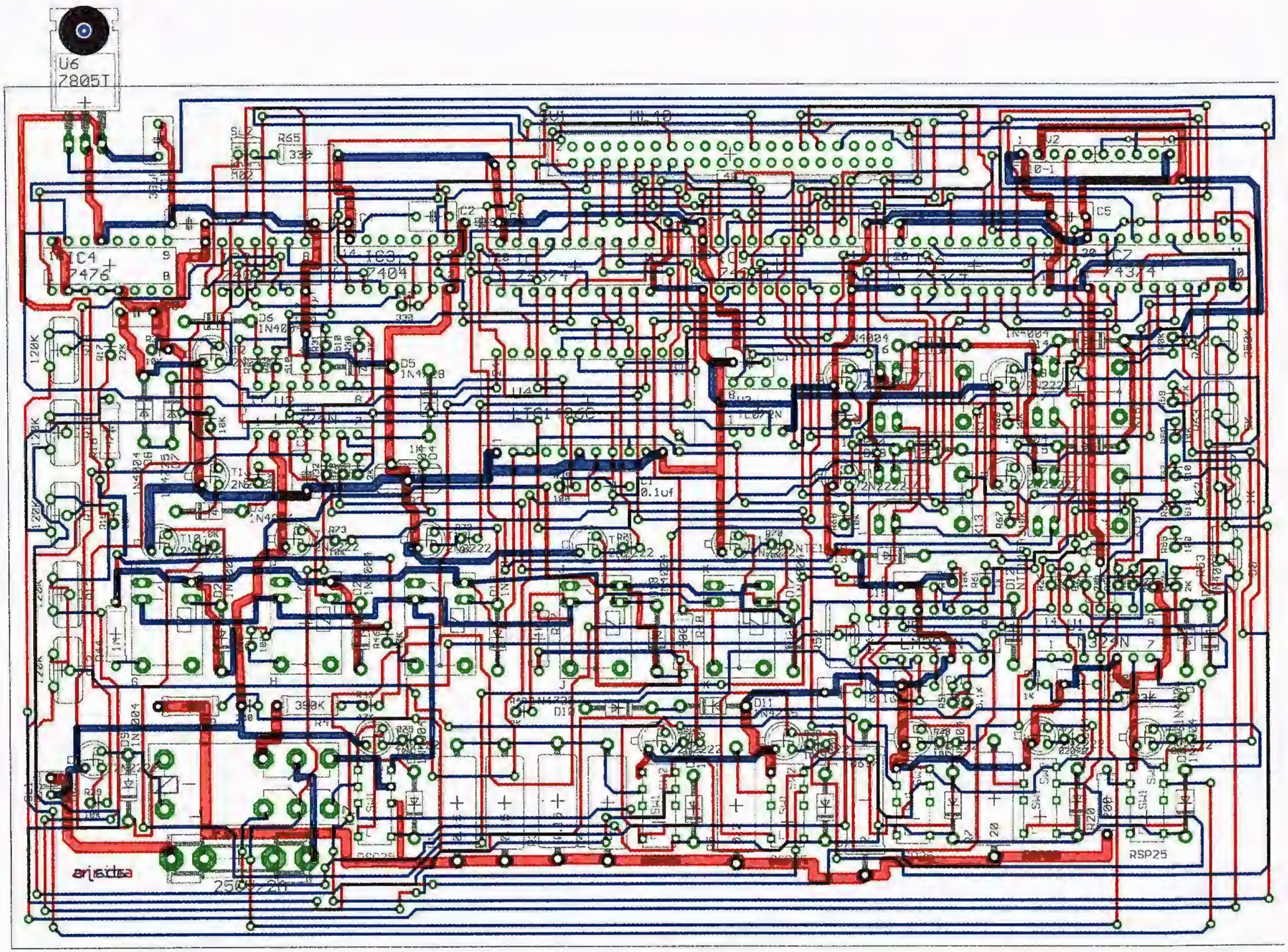


Frecuencimetro.

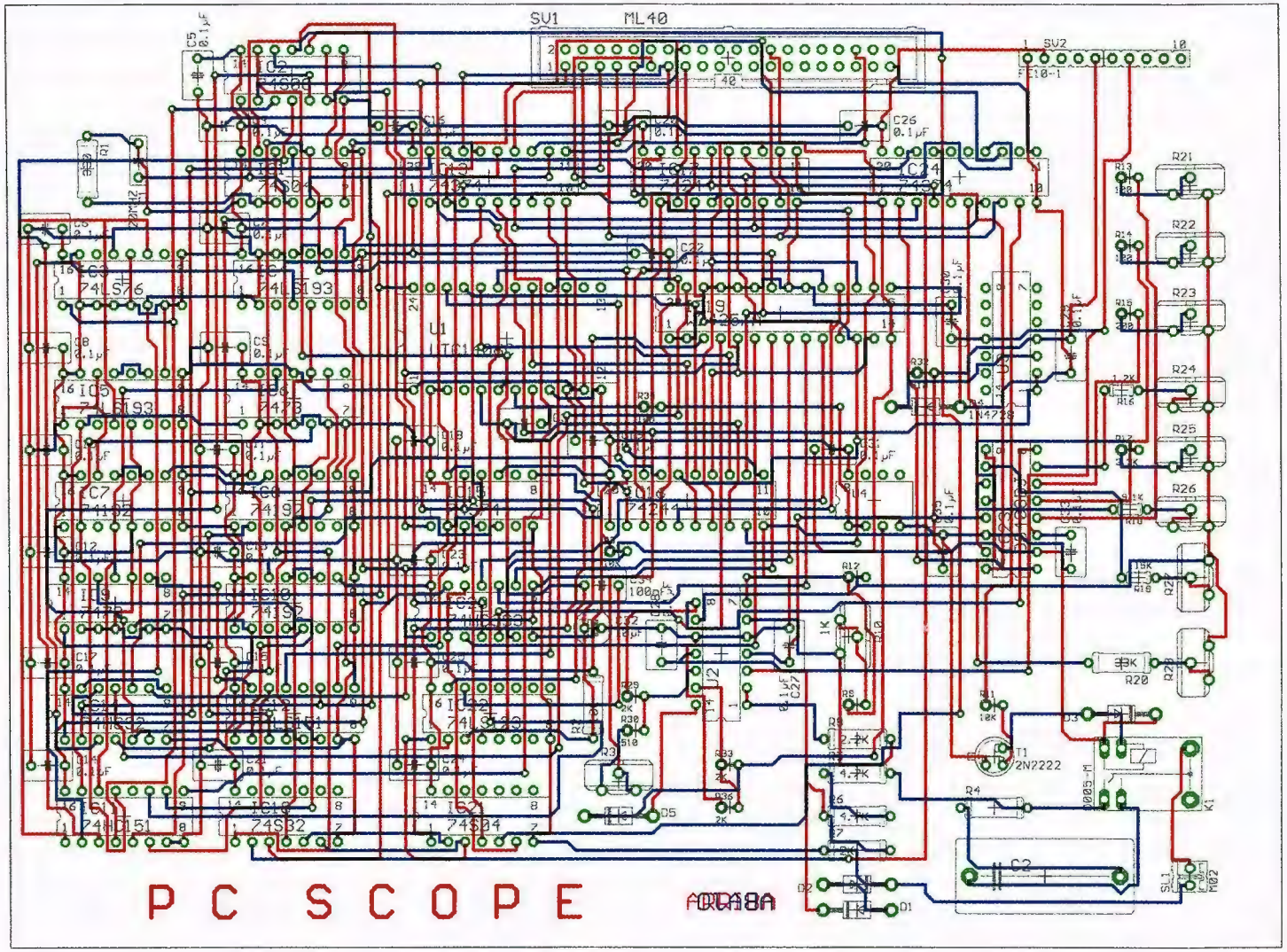


Fuente de voltaje.


GEN-FUN
GEN - FUN
Arriba
Apoyo
SW1
FF20-2
IC1
74374
IC2
74374
IC3
74374
IC4
74374
IC5
U1
ONC0808
U2
TL084
U3
TL084
U4
TL084
U5
TL084
U6
TL084
U7
TL084
U8
TL084
U9
TL084
U10
TL084
U11
TL084
U12
TL084
U13
TL084
U14
TL084
U15
TL084
U16
TL084
U17
TL084
U18
TL084
U19
TL084
U20
TL084
U21
TL084



Multimetro



Osciloscopio.

LTC1406

Low Power, 8-Bit, 20Msps, Sampling A/D Converter

FEATURES

- **Low Power, 8-Bit, 20Msps ADC**
- **250MHz Internal Sample-and-Hold**
- **7 Effective Bits at 70MHz Input Frequency**
- **±1LSB DNL and INL Max**
- Single 5V Supply and 150mW Dissipation
- Power Down to 1μA
- True Differential Inputs Reject Common Mode Noise
- Accepts Single-Ended or Differential Input Signals
- ±1V Differential or 2V Single-Ended Input Span
- Analog Inputs Common Mode to V_{DD} and GND
- 24-Pin Narrow SSOP Package

APPLICATIONS

- Telecommunications
- Wireless Communications
- Digital Cellular Telephones
- CCDs and Image Scanners
- Video Digitizing and Digital Television
- Digital Color Copiers
- High Speed Undersampling
- Personal Computer Video
- High Speed Data Acquisition

DESCRIPTION

The LTC®1406 is a 20Msps, 8-bit, sampling A/D converter which draws only 150mW from a single 5V supply. This easy-to-use device includes a high dynamic range sample-and-hold with a 250MHz bandwidth.

The LTC1406's full-scale input range is ±1V. The inputs can be driven differentially or one input can be tied to a fixed voltage and the other input driven with a ±1V bipolar input. Maximum DC specifications include ±1LSB DNL and INL over temperature. Outstanding AC performance includes 48.5dB S/(N + D) and 62dB THD with a 1MHz input; 47.5dB S/(N + D) and 59dB THD at the Nyquist input frequency of 10MHz.

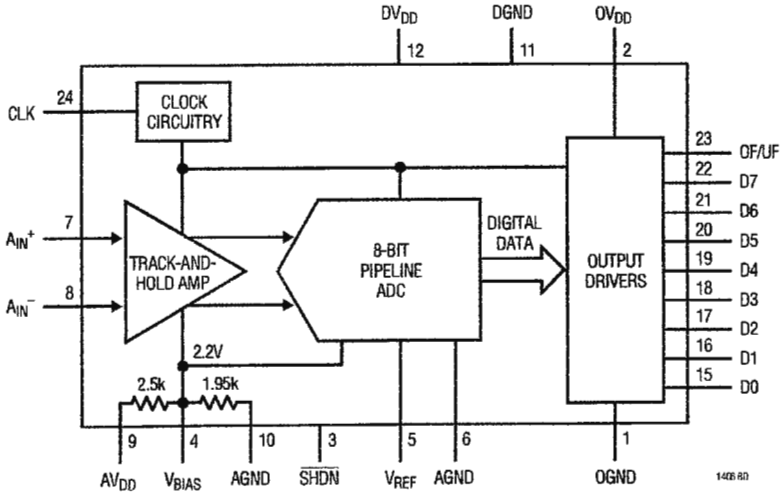
The unique differential input sample-and-hold can acquire single-ended or differential input signals up to its 250MHz bandwidth. The 60dB common mode rejection allows users to eliminate ground loops and common mode noise by measuring signals differentially from the source.

The ADC has an 8-bit parallel output port with separate power supply and ground allowing easy interface to 3V digital systems. The pipelined architecture has five clock cycles of data latency.

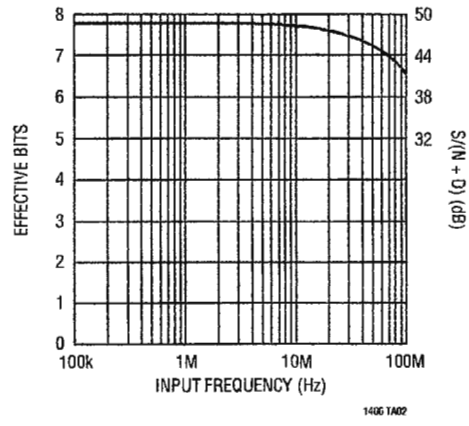
LT, LTC and LT are registered trademarks of Linear Technology Corporation.

TYPICAL APPLICATION

Low Power, 20MHz, 8-Bit Sampling ADC



Effective Bits and Signal-to-Noise + Distortion vs Input Frequency

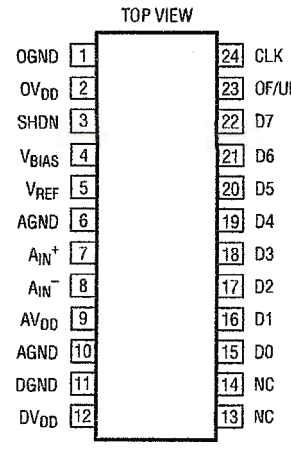


ABSOLUTE MAXIMUM RATINGS

$AV_{DD} = OV_{DD} = DV_{DD} = V_{DD}$ (Notes 1, 2)

Supply Voltage (V_{DD}) 6V
Analog Input Voltage (Note 3) .... – 0.3V to (V_{DD} + 0.3V)
Digital Input Voltage (Note 4) ..... – 0.3V to 10V
Digital Output Voltage ..... – 0.3V to (V_{DD} + 0.3V)
Power Dissipation 500mW
Ambient Operation Temperature Range
LTC1406C ..... 0°C to 70°C
LTC1406I ..... –40°C to 85°C
Storage Temperature Range ..... –65°C to 150°C
Lead Temperature (Soldering, 10 sec) ..... 300°C

PACKAGE/ORDER INFORMATION



GN PACKAGE
24-LEAD PLASTIC SSOP
T_{JMAX} = 110°C, θ_{JA} = 85°C/W

ORDER PART NUMBER
LTC1406CGN LTC1406IGN

Consult factory for Military grade parts.

CONVERTER CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C. (Notes 5, 6)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Resolution (No Missing Codes)		●	8			Bits
Integral Linearity Error	(Note 7)	●		±0.5	±1	LSB
Differential Linearity Error		●		±0.25	±1	LSB
Offset Error	(Note 8)	●		±1	±8	LSB
Gain Error	With External 2.5V Reference			±1	±5	LSB

ANALOG INPUT (Note 5)

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V_{IN}	Analog Input Span [(A_{IN}^+) – (A_{IN}^-)] (Note 9)	4.75V ≤ V_{DD} ≤ 5.25V	●		±1		V
	Input (A_{IN}^+ or A_{IN}^-) Range	Voltage On Either A_{IN}^+ or A_{IN}^-	●	0		V_{DD}	V
I_{IN}	Analog Input Leakage Current	CLK = 0	●			±5	µA
C_{IN}	Analog Input Capacitance	CLK = 1 CLK = 0			4 2		pF pF
	Input Bandwidth				250		MHz
t_{AP}	Sample-and-Hold Aperture Delay Time				3		ns
t_{jitter}	Sample-and-Hold Aperture Delay Time Jitter				5		ps_{RMS}
CMRR	Analog Input Common Mode Rejection Ratio	–2.5V < (A_{IN}^- = A_{IN}^+) < 2.5V			60		dB
V_{BIAS}	Internal Bias Voltage	No Load			2.2		V

DYNAMIC ACCURACY

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C. (Note 5)

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
S/(N + D)	Signal-to-Noise Plus Distortion Ratio	1MHz Input Signal 10MHz Input Signal			48.5 47.5		dB dB
THD	Total Harmonic Distortion	1MHz Input Signal, First 5 Harmonics 10MHz Input Signal, First 5 Harmonics			−62 −59		dB dB
SFDR	Spurious Free Dynamic Range	1MHz Input Signal 10MHz Input Signal			63 60		dB dB
IMD	Intermodulation Distortion	f_{IN1} = 3.500977MHz, f_{IN2} = 3.598633MHz			60		dB

DIGITAL INPUTS AND OUTPUTS (Note 5)

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V_{IH}	High Level Input Voltage	V_{DD} = 5.25V	●	2.4			V
V_{IL}	Low Level Input Voltage	V_{DD} = 4.75V	●			0.8	V
I_{IN}	Digital Input Current	V_{IN} = 0V to V_{DD}	●			±5	µA
C_{IN}	Digital Input Capacitance				5		pF
V_{OH}	High Level Output Voltage	V_{DD} = 4.75V, I_O = −10µA V_{DD} = 4.75V, I_O = −200µA	 ●	 4.0	4.5		V V
V_{OL}	Low Level Output Voltage	V_{DD} = 4.75V, I_O = 160µA V_{DD} = 4.75V, I_O = 1.6mA	 ●		0.05 0.10	 0.4	V V
I_{SOURCE}	Output Source Current	V_{OUT} = 0V			−20		mA
I_{SINK}	Output Sink Current	V_{OUT} = V_{DD}			30		mA

POWER REQUIREMENTS (Note 5)

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
AV_{DD}	Analog Positive Supply Voltage	(Note 10)		4.75		5.25	V
DV_{DD}	Digital Positive Supply Voltage	(Note 10)		4.75		5.25	V
OV_{DD}	Output Positive Supply Voltage	(Note 10)		2.7		5.25	V
V_{BIAS}	Internal Bias Voltage	When Externally Driven (Note 10)		1.9	2.2	2.5	V
V_{REF}	Reference Voltage	(Note 10)		2	2.5	3	V
OGND	Output Ground	(Note 10)		0		2	V
I_{DD}	Positive Supply Current	AV_{DD} = DV_{DD} = OV_{DD} = 5V, f_{SMPL} = 20MHz (Note 13)	●		30	45	mA
P_D	Power Dissipation		●		150	225	mW
	Power Down Positive Supply Current	$\overline{SHDN}$ = 0V, CLK = V_{DD} or 0			1	10	µA
	Power Down Power Dissipation	$\overline{SHDN}$ = 0V, CLK = V_{DD} or 0			5	50	µW

TIMING CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are T_A = 25°C. (Note 5)

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
$f_{SMPL(MAX)}$	Maximum Sampling Frequency		●	20			MHz
t_1	Clock Period	(Notes 11, 12)	●	50			ns
t_2	Pulse Width High	(Notes 11, 12)	●	25			ns
t_3	Pulse Width Low	(Notes 11, 12)	●	25			ns
t_4	Output Delay	C_L = 15pF			15	25	ns
t_5	Pipeline Delay				5		Cycles
t_6	Aperture Delay				3		ns
	Aperture Jitter				5		ps_{RMS}

Note 1: Absolute Maximum Ratings are those values beyond which the life of a device may be impaired.

Note 2: All voltage values are with respect to ground with DGND, OGND and AGND wired together (unless otherwise noted).

Note 3: When these pin voltages are taken below ground or above V_{DD}, they will be clamped by internal diodes. This product can handle input currents greater than 100mA below ground or above V_{DD} without latchup.

Note 4: When these pin voltages are taken below ground they will be clamped by internal diodes. This product can handle input currents up to 100mA below ground without latchup. These pins are not clamped to V_{DD}.

Note 5: V_{DD} = 5V, f_{SMPL} = 20MHz and $t_r = t_f$ = 2ns unless otherwise specified.

Note 6: Linearity, offset and full-scale specifications apply for a single-ended A_{IN}^+ input with A_{IN}^- tied to V_{REF} = 2.5V.

Note 7: Integral nonlinearity is defined as the deviation of a code from a straight line passing through the actual endpoints of the transfer curve. The deviation is measured from the center of the quantization band.

Note 8: Bipolar offset is the offset voltage measured from –0.5LSB when the output code flickers between 0111 1111 and 1000 0000.

Note 9: Guaranteed by design, not subject to test.

Note 10: Recommended operating conditions.

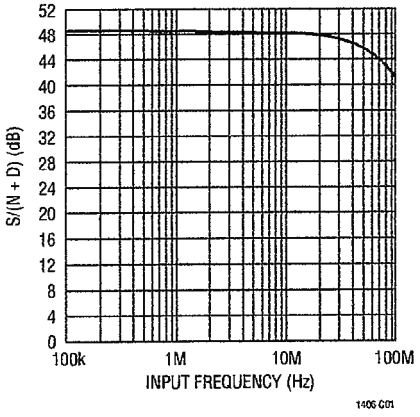
Note 11: The falling CLK edge starts a conversion.

Note 12: At the maximum conversion rate, deviation from a 50% duty cycle results in interstage settling times <25ns and performance may be affected.

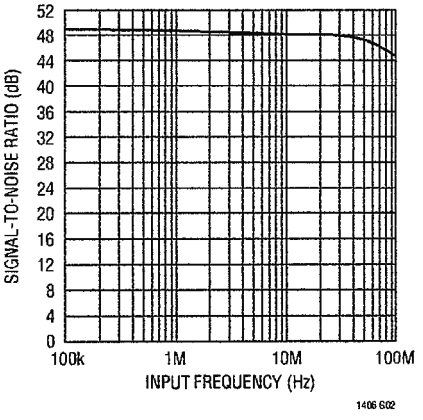
Note 13: V_{IN} = –Full Scale.

TYPICAL PERFORMANCE CHARACTERISTICS

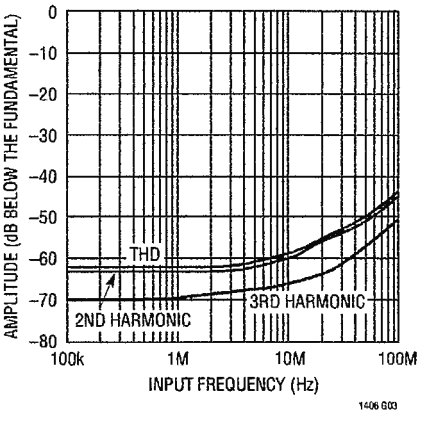
S/(N + D) vs Input Frequency



Signal-to-Noise Ratio vs Input Frequency

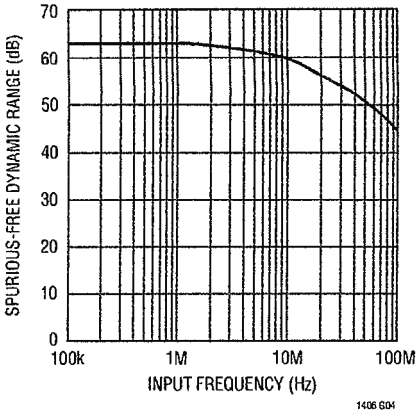


Distortion vs Input Frequency

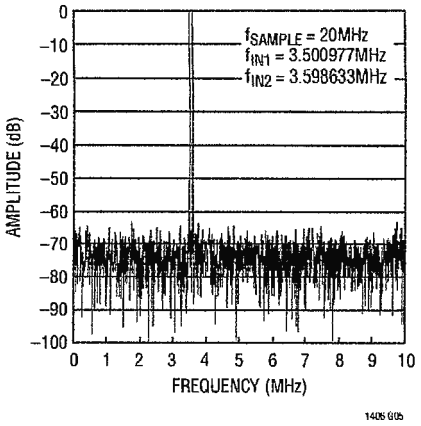


TYPICAL PERFORMANCE CHARACTERISTICS

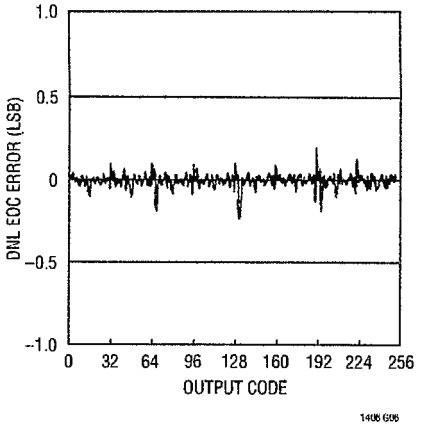
Spurious-Free Dynamic Range vs Input Frequency



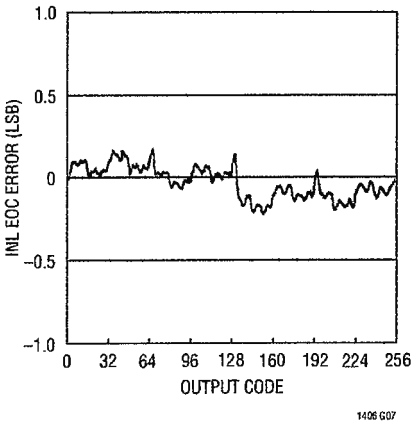
Intermodulation Distortion Plot



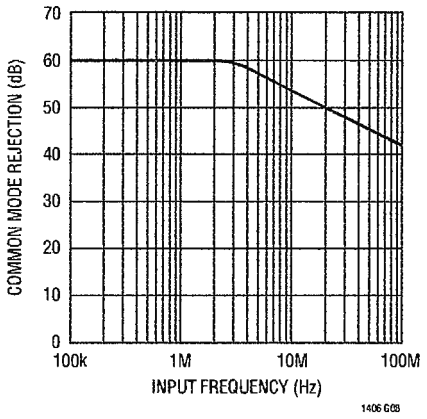
Differential Nonlinearity vs Output Code



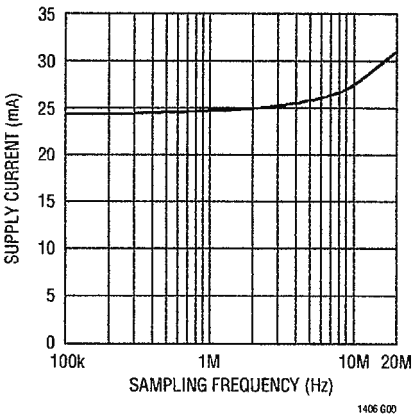
Integral Nonlinearity vs Output Code



Input Common Mode Rejection vs Input Frequency



Supply Current vs Sampling Frequency



PIN FUNCTIONS

OGND (Pin 1): Digital Data Output Ground. Tie to analog ground plane. May be tied to logic ground if desired.

OV_{DD} (Pin 2): Digital Data Output Supply. Normally tied to 5V, can be used to interface with 3V digital logic. Bypass to OGND with 10µF tantalum in parallel with 0.1µF or 10µF ceramic.

$\overline{SHDN}$ (Pin 3): Power Shutdown Input. Logic low selects shutdown.

V_{BIAS} (Pin 4): Internal Bias Voltage. Internally set to 2.2V. Bypass to analog ground plane with 10µF tantalum in parallel with 0.1µF or 10µF ceramic.

V_{REF} (Pin 5): External 2.5V Reference Input. Bypass to analog ground plane with 10µF tantalum in parallel with 0.1µF or 10µF ceramic.

AGND (Pin 6): Analog Ground. Tie to analog ground plane.

A_{IN}^+ (Pin 7): ±1V Input. The maximum output code occurs when $[(A_{IN}^+) - (A_{IN}^-)] = 1V$. The minimum output code occurs when $[(A_{IN}^+) - (A_{IN}^-)] = -1V$.

A_{IN}^- (Pin 8): ±1V Input. The maximum output code occurs when $[(A_{IN}^+) - (A_{IN}^-)] = 1V$. The minimum output code occurs when $[(A_{IN}^+) - (A_{IN}^-)] = -1V$. For single-ended operation, tie A_{IN}^- to a DC voltage (e.g., V_{REF}).

PIN FUNCTIONS

AV_{DD} (Pin 9): Analog 5V Positive Supply. Bypass to analog ground plane with 10µF tantalum in parallel with 0.1µF or 10µF ceramic.

AGND (Pin 10): Analog Ground. Tie to analog ground plane.

DGND (Pin 11): Digital Ground for Internal Logic. Tie to analog ground plane.

DV_{DD} (Pin 12): Digital 5V Positive Supply. Bypass to DGND with 10µF tantalum in parallel with 0.1µF or 10µF ceramic.

NC (Pins 13, 14): No Internal Connection.

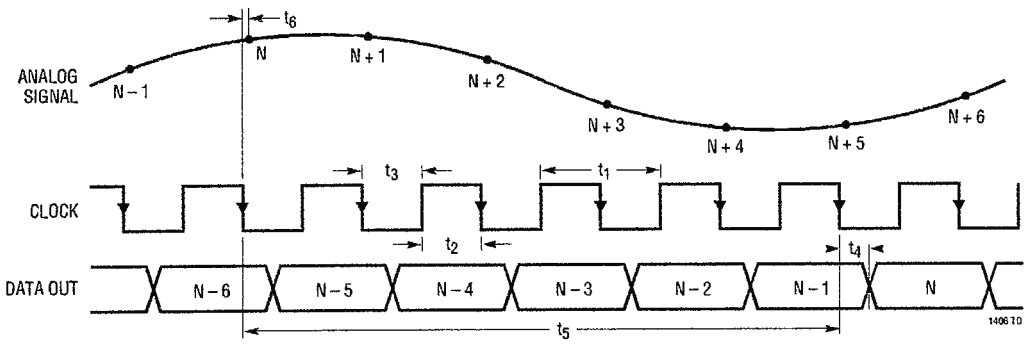
D7 to D0 (Pins 15 to 22): Digital Data Outputs. The outputs swing between OV_{DD} and OGND.

OF/UF (Pin 23): Overflow/Underflow Bit. OF/UF high with D7 to D0 all high indicates an overrange, OF/UF high with D7 to D0 all low indicates an underrange condition. OF/UF low indicates a conversion within the normal input range. The outputs swing between OV_{DD} and OGND.

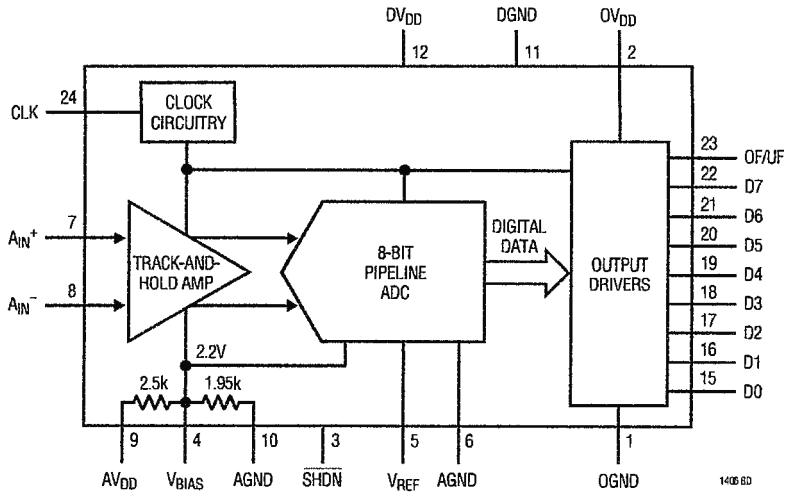
CLK (Pin 24): Clock Input. Internal sample-and-hold tracks the input signal when CLK is high and samples the input signal on the falling edge.

$AV_{DD} = DV_{DD} = V_{DD}$			NOMINAL (V)			ABSOLUTE MAXIMUM (V)	
PIN	NAME	DESCRIPTION	MIN	TYP	MAX	MIN	MAX
1	OGND	Ground for Output Drivers		0		−0.3	V_{DD} + 0.3
2	OV_{DD}	Supply for Output Drivers	2.7	3 or 5	5.25	−0.3	6
3	$\overline{SHDN}$	Shutdown Input, Active Low	0		V_{DD}	−0.3	10
4	V_{BIAS}	Internal Bias Voltage	1.9	2.2	2.5	−0.3	V_{DD} + 0.3
5	V_{REF}	External Reference Input	2	2.5	3	−0.3	V_{DD} + 0.3
6	AGND	Analog Ground, Clean Ground		0		−0.3	V_{DD} + 0.3
7	A_{IN}^+	Positive Analog Input, ±1V Span	0		V_{DD}	−0.3	V_{DD} + 0.3
8	A_{IN}^-	Negative Analog Input	0		V_{DD}	−0.3	V_{DD} + 0.3
9	AV_{DD}	Analog Supply	4.75	5	5.25	−0.3	6
10	AGND	Analog Ground, Substrate Ground		0		−0.3	V_{DD} + 0.3
11	DGND	Digital Ground		0		−0.3	V_{DD} + 0.3
12	DV_{DD}	Digital Supply	4.75	5	5.25	−0.3	6
13 to 14	NC	No Connect, No Internal Connection					
15 to 22	D7 to D0	Data Outputs	OGND		OV_{DD}	−0.3	V_{DD} + 0.3
23	OF/UF	Overflow/Underflow Output	OGND		OV_{DD}	−0.3	V_{DD} + 0.3
24	CLK	Clock Input	0		V_{DD}	−0.3	10

TIMING DIAGRAM



FUNCTIONAL BLOCK DIAGRAM



APPLICATIONS INFORMATION

Conversion Details

The LTC1406 uses an internal sample-and-hold circuit and a pipeline quantizing architecture to convert an analog signal to an 8-bit parallel output. With CLK high the input switches are closed and the analog input will be acquired on the input sampling capacitors C_S (see Figure 1).

On the falling edge of CLK the input switches open, capturing the input signal. The sampling capacitors are then shorted together and the charge is transferred to the hold capacitors C_H resulting in a differential DC voltage on the output of the track-and-hold amplifier that is proportional to the input signal. This differential voltage is fed into a comparator that determines the most significant bit and subtracts the result. The residue is then amplified by two and passed to the next stage via a similar sample-and-hold circuit. This continues down the eight pipeline stages. The comparator outputs are then combined in a digital error correction circuit. The 8-bit word is available at the output, five clock cycles after the sampling edge.

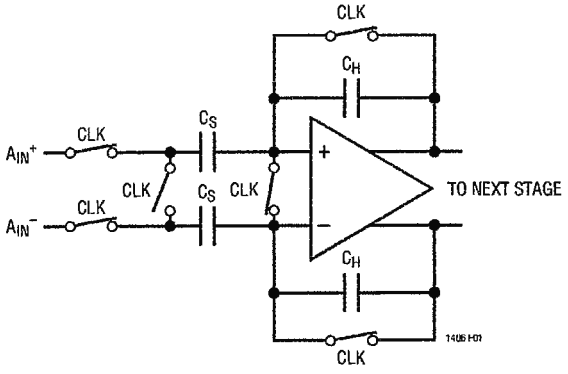


Figure 1. Input Sample-and-Hold Amplifier

Dynamic Performance

The LTC1406 has excellent wideband sampling capability. The sample-and-hold amplifier has a small-signal input bandwidth of 250MHz allowing the ADC to undersample input signals with frequencies well beyond the converter's Nyquist frequency. FFT (Fast Fourier Transform) test techniques are used to test the ADC's frequency response, distortion and noise at the rated throughput. By applying a low distortion sine wave and analyzing the digital output using an FFT algorithm, the ADC's spectral content can be examined for frequencies outside the fundamental. Figure 2 shows a typical LTC1406 FFT plot.

APPLICATIONS INFORMATION

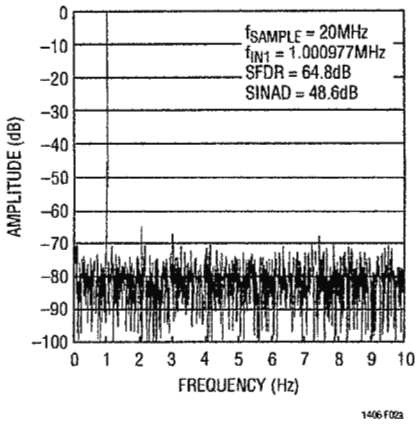


Figure 2a. Nonaveraged, 4096 Point FFT Input Frequency = 1MHz

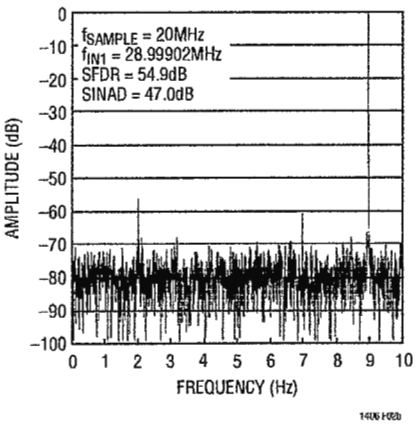


Figure 2b. Nonaveraged, 4096 Point FFT Input Frequency = 30MHz

Signal-to-Noise Ratio

The signal-to-noise plus distortion ratio [S/(N + D)] is the ratio between the RMS amplitude of the fundamental input frequency and the RMS amplitude of all other frequency components at the ADC output. The output is band limited to frequencies above DC to below half the sampling frequency. The effective number of bits (ENOBs) is a measurement of the resolution of an ADC and is directly related to the S/(N + D) by the equation:

$$ENOB = [S/(N + D) - 1.76]/6.02$$

where ENOB is the effective number of bits and S/(N + D) is expressed in dB. At the maximum sampling rate of 20MHz the LTC1406 maintains near ideal ENOBs up to and beyond the Nyquist input frequency of 10MHz (see Figure 3).

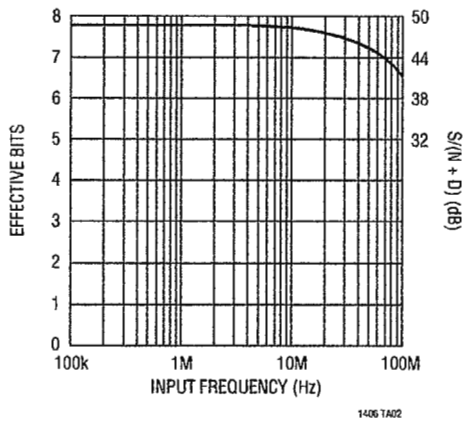


Figure 3. Effective Bits and Signal-to-(Noise + Distortion) vs Input Frequency

Total Harmonic Distortion

Total harmonic distortion is the ratio of the RMS sum of all harmonics of the input signal to the fundamental itself. The out-of-band harmonics alias into the frequency band between DC and half the sampling frequency. THD is expressed as:

$$THD = 20 \log \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \ldots V_n^2}}{V_1}$$

where V_1 is the RMS amplitude of the fundamental frequency and V_2 through V_n are the amplitudes of the second through n^{th} harmonics. THD vs Input Frequency is shown in Figure 4. The LTC1406 has good distortion performance up to the Nyquist frequency and beyond.

Intermodulation Distortion

If the ADC input signal consists of more than one spectral component, the ADC transfer function nonlinearity can produce intermodulation distortion (IMD) in addition to THD. IMD is the change in one sinusoidal input caused by the presence of another sinusoidal input at a different frequency (see Figure 5).

APPLICATIONS INFORMATION

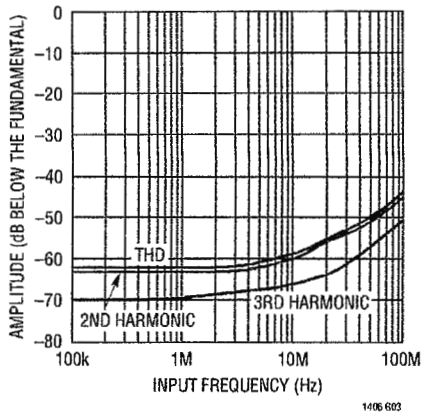


Figure 4. Distortion vs Input Frequency

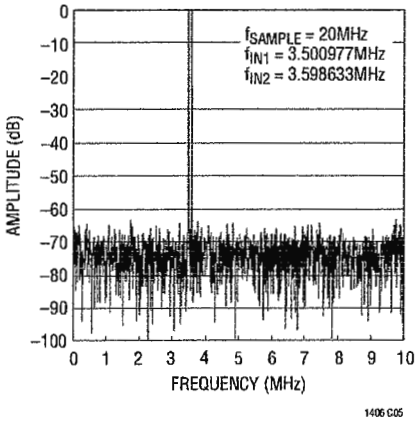


Figure 5. Intermodulation Distortion Plot

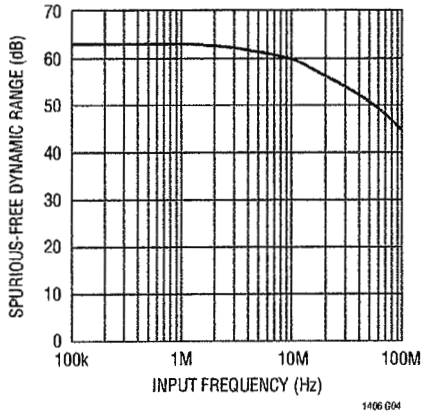


Figure 6. Spurious-Free Dynamic Range vs Input Frequency

If two pure sine waves of frequencies f_a and f_b are applied to the ADC input, nonlinearities in the ADC transfer function can create distortion products at the sum and difference frequencies of $mf_a \pm nf_b$, where m and n = 0, 1, 2, 3, etc. For example, the 2nd order IMD terms include $(f_a \pm f_b)$. If the two input sine waves are equal in magnitude, the value (in decibels) of the 2nd order IMD products can be expressed by the following formula:

$$IMD(f_a \pm f_b) = 20 \log \frac{\text{Amplitude at } (f_a \pm f_b)}{\text{Amplitude at } f_a}$$

Peak Harmonic or Spurious Noise

The peak harmonic or spurious noise is the largest spectral component excluding the input signal and DC. This value is expressed in decibel relative to the RMS value of a full-scale input signal (see Figure 6).

Input Bandwidth

The input bandwidth is that input frequency at which the amplitude of the reconstructed fundamental is reduced by 3dB for a full-scale input signal. The LTC1406 has been designed for wide input bandwidth (250MHz), allowing the ADC to undersample input signals with frequencies above the converter's Nyquist frequency. The noise floor stays very low at high frequencies; S/(N + D) becomes dominated by distortion at frequencies far beyond Nyquist.

Analog Inputs

The LTC1406 has a unique differential sample-and-hold circuit that allows rail-to-rail inputs. The A_{IN}^+ and A_{IN}^- inputs are sampled at the same time and the ADC will always convert the difference of $[(A_{IN}^+) - (A_{IN}^-)]$ independent of the common mode voltage. Any unwanted signal that is common to both inputs will be rejected by the common mode rejection of the sample-and-hold circuit. The common mode rejection holds up to extremely high frequencies (see Figure 7).

The inputs can be driven differentially or single-ended. In differential mode, both inputs are driven ±0.5V out of phase with each other. In single-ended mode, the negative input is tied to a fixed voltage and A_{IN}^+ is used as the

APPLICATIONS INFORMATION

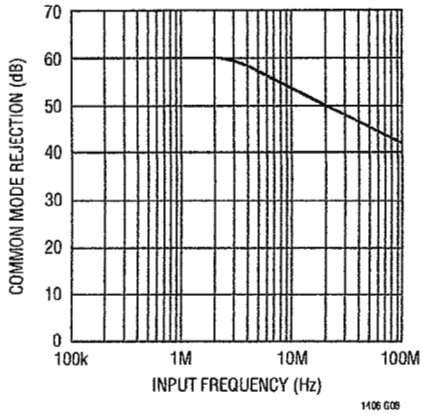


Figure 7. Common Mode Rejection vs Input Frequency

single input providing a ±1V bipolar input range centered around A_{IN}^-. Likewise, A_{IN}^+ can be tied to a fixed voltage and A_{IN}^- used as the single input. In any configuration the maximum output code (1111 1111) occurs when $[(A_{IN}^+) - (A_{IN}^-)] = 1V$ and the minimum output code (0000 0000) occurs when $[(A_{IN}^+) - (A_{IN}^-)] = -1V$.

Each analog input can swing from ground to V_{DD} but not beyond. Therefore, the input common mode voltage can range from 0.5V to 4.5V in differential mode and from 1V to 4V in single-ended mode.

As an example, with A_{IN}^- connected to the V_{REF} pin (2.5V) the input range will be 1.5V to 3.5V (see Figure 8a). To achieve other ranges the input may be capacitively coupled to achieve a 2V span with virtually any common mode voltage (see Figure 8b).

The 2V input span requires a 2.5V external reference be connected to the V_{REF} pin. The LT1460-2.5 micropower precision series reference is recommended. To achieve other input spans, the reference voltage (V_{REF}) can vary between 2V to 3V. The V_{REF} pin can also be driven with a DAC or other means. This is useful in applications where the peak input signal amplitude may vary. The input span of the ADC can then be adjusted to match the peak input signal, maximizing the signal-to-noise ratio.

The analog inputs of the LTC1406 are easy to drive. The inputs draw only one small current spike while charging the sample-and-hold capacitors following a rising CLK edge.

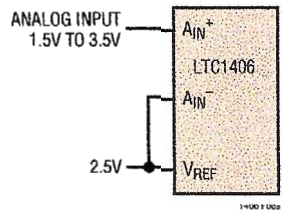


Figure 8a. DC Coupled

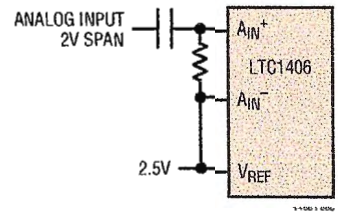


Figure 8b. AC Coupled

While CLK is low the analog inputs draw only a small leakage current. If the source impedance of the driving circuit is low, then the LTC1406 inputs can be driven directly. As source impedance increases, so will acquisition time. For minimum acquisition time with high source impedance, a buffer amplifier should be used. The only requirement is that the amplifier driving the analog input(s) must settle after the small current spike before the next conversion starts (settling time must be 25ns for full throughput rate).

Choosing an Input Amplifier

Choosing an input amplifier is easy if a few requirements are taken into consideration. First, to limit the magnitude of the voltage spike seen by the amplifier from charging the sampling capacitor, choose an amplifier that has a low output impedance (<50Ω) at the closed-loop bandwidth frequency. For example, if an amplifier is used in a gain of 1 and has a unity-gain bandwidth of 50MHz, then the output impedance at 50MHz must be less than 50Ω. The second requirement is that the closed-loop bandwidth must be greater than 70MHz to ensure adequate small-signal settling for full throughput rate.

The following list is a summary of the op amps that are suitable for driving the LTC1406. More detailed information is available in the Linear Technology Databooks and on the LinearView™ CD-ROM.

LT®1223: 100MHz Video Current Feedback Amplifier. 6mA supply current. ±5V to ±15V supplies. Low noise.

LT1227: 140MHz Video Current Feedback Amplifier. 10mA supply current. ±5V to ±15V supplies. Low distortion. Low noise.

LinearView is a trademark of Linear Technology Corporation.

APPLICATIONS INFORMATION

LT1229/LT1230: Dual and Quad 100MHz Current Feedback Amplifiers. ±2V to ±15V supplies. Low noise. 6mA supply current each amplifier.

LT1259/LT1260: Dual and Triple 130MHz Current Feedback Amplifiers. ±2V to ±14V supplies. 5mA supply current. Low distortion. Low noise.

LT1363: 70MHz Voltage Feedback Amplifier. ±2.5V to ±15V supplies. 7.5mA supply current. Low distortion.

LT1364/LT1365: Dual and Quad 70MHz Voltage Feedback Amplifiers. ±2.5V to ±15V supplies. 7.5mA supply current per amplifier. Low distortion.

Input Filtering

The noise and the distortion of the input amplifier and other circuitry must be considered since they will add to the LTC1406 noise and distortion. The small-signal bandwidth of the sample-and-hold circuit is 250MHz. Any noise or distortion products that are present at the analog inputs will be summed over this entire bandwidth. Noisy input circuitry should be filtered prior to the analog inputs to minimize noise. A simple 1-pole RC filter is sufficient for many applications. For example, Figure 9 shows a 220pF capacitor from A_{IN}^+ to A_{IN}^- and a 75Ω source resistor to limit the input bandwidth to 9.6MHz. The 220pF capacitor also acts as a charge reservoir for the input sample-and-hold and isolates the ADC input from sampling glitch sensitive circuitry. Larger value capacitors may be substituted to further limit the input bandwidth. High quality capacitors and resistors should be used since these components can add distortion. NPO and silver mica type dielectric capacitors have excellent linearity. Carbon surface mount resistors can also generate distortion from self-heating and from damage that may occur during soldering. Metal film surface mount resistors are much less susceptible to both problems.

Input/Output Characteristics

Figure 10 shows the ideal input/output characteristics for the LTC1406. The code transitions occur midway between successive integer LSB values (i.e., –FS + 0.5LSB, –FS + 1.5LSB, –FS + 2.5LSB...FS – 1.5LSB, FS – 0.5LSB). The output is straight binary with 1LSB = FS – (–FS)/256 = 2V/256 = 7.8125mV. The OF/UF bit indicates that the input has exceeded full scale and can be used to detect an overrange or underrange condition. A logic high output on the OF/UF pin with an output code of 0000 0000 indicates the input is less than the negative full scale. A logic high output on the OF/UF pin with an output code of 1111 1111 indicates that the input is greater than the positive full scale. A logic low output on the OF/UF pin indicates the input is within the full-scale range of the converter.

In applications where absolute accuracy is important, offset and full-scale errors can be adjusted to zero. Offset error must be adjusted before full-scale error. Zero offset is achieved by adjusting the offset applied to the A_{IN}^- input. For zero offset error, apply a voltage equal to the input

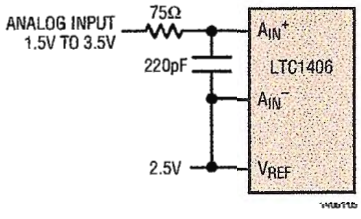


Figure 9. RC Input Filter

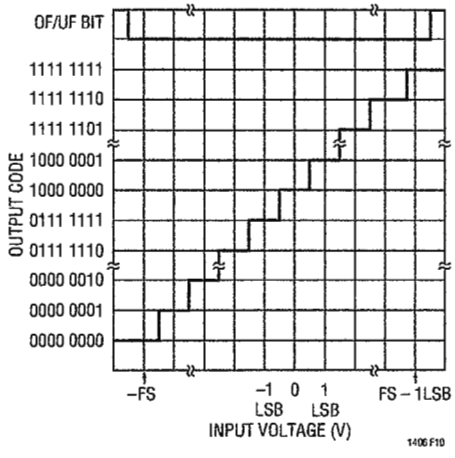


Figure 10. Transfer Characteristics

APPLICATIONS INFORMATION

common mode voltage minus 3.90625mV (i.e., –0.5LSB) and adjust the offset at the A_{IN}^- input until the output code flickers between 0111 1111 and 1000 0000. For full-scale adjustment, an input voltage equal to the input common mode voltage plus 988.28125mV (i.e., FS – 1.5LSBs) is applied to A_{IN}^+ and the V_{REF} input is adjusted until the output code flickers between 1111 1110 and 1111 1111.

Digital Inputs and Outputs

The LTC1406 is designed to easily interface with either 3V or 5V logic. The digital input pins, $\overline{SHDN}$ and CLK, have thresholds of nominally 1.9V and will accept a 3V or 5V logic input. The data output pins, including OF/UF, are connected to a separate supply and ground (OV_{DD} and OGND respectively). OV_{DD} is normally connected to DV_{DD} but can be connected to an external supply as low as 2.7V. OGND is normally connected to DGND but can be connected to an external ground or an external voltage source as high as 2V.

Clock

The LTC1406 requires a 50% duty cycle clock. The duty cycle should be timed from the nominal threshold of the CLK input which is 1.9V. At conversion speeds below the maximum conversion rate of 20MHz, the duty cycle can deviate from 50% with no degradation in performance as long as each clock phase is at least 25ns long. At the maximum conversion rate, deviation from a 50% duty cycle clock results in interstage settling times of <25ns and performance may be affected.

With the CLK pin high, the ADC will track the difference of the two analog inputs. On the falling edge of CLK the input is sampled and the conversion begins. At the end of five clock cycles (on the fifth falling CLK edge following the start of conversion) the data from the conversion will be available at the digital outputs until the next falling CLK edge. Each falling edge of CLK starts a new conversion so successive conversion results are available on successive falling CLK edges.

While the falling edge starts the conversion, both rising and falling edges are used internally during the conversion. It is therefore important to provide a clock signal that has low jitter and fast rise and fall times (<2ns). Much of the internal circuitry operates dynamically limiting the minimum conversion rate to 10kHz. To ensure proper operation after power is first applied, or the clock stops for more than 100µs, typically 20 clock cycles must be performed at a sample rate above 10kHz before the output data will be valid.

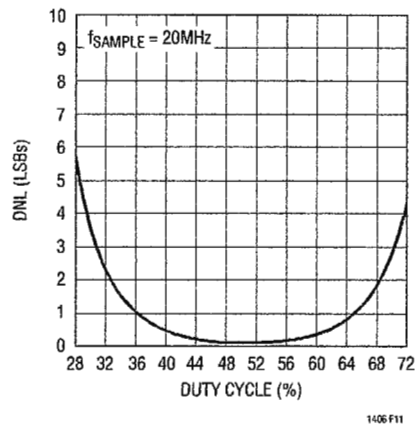


Figure 11. Typical DNL vs Duty Cycle

Power Shutdown

The quiescent power of the LTC1406 can be further reduced between conversions by taking the $\overline{SHDN}$ pin low. This powers down all of the internal amplifiers and bias circuitry and the part draws only a small quiescent current of 1µA from the 5V supply. There is a nominally 4k internal resistor between V_{REF} and AGND that will continue to draw current during shutdown as long as V_{REF} is driven. It should also be noted that the data output drivers are not three-state devices and do not go into a high impedance state during shutdown. If the data output pins will remain connected to a load during shutdown, current may be drawn through the OV_{DD} supply pin. This can be prevented by including a FET switch in series with OV_{DD} or OGND controlled by $\overline{SHDN}$. If the data bus will remain active during

APPLICATIONS INFORMATION

shutdown. It may also be desirable to isolate the data output pins from the bus to reduce the load capacitance. To resume normal operation the $\overline{SHDN}$ pin must be brought high and typically 20 clock cycles must be performed at a sample rate above 10kHz before the output data will be valid.

Board Layout and Bypassing

Wire wrap boards are not recommended for high resolution or high speed A/D converters. To obtain the best performance from the LTC1406, a printed circuit board with ground plane is required. Layout for the printed circuit board should ensure that digital and analog signal lines are separated as much as possible. In particular, care should be taken not to run any digital track alongside an analog signal track or underneath the ADC.

An analog ground plane separate from the logic system ground should be established under and around the ADC. Pin 1 (OGND), Pin 6 (AGND), Pin 10 (AGND) and Pin 11 (DGND) and all other analog grounds should be connected to this single analog ground point. The V_{CM}, V_{REF}, DV_{DD} and OV_{DD} bypass capacitors should also be connected to this analog ground plane. No other digital grounds should be connected to this analog ground plane. In some applications it may be desirable to connect the OV_{DD} to the logic system supply and OGND to the logic system ground. In these cases OV_{DD} should be bypassed to OGND instead of the analog ground plane.

Low impedance analog and digital power supply common returns are essential to low noise operation of the ADC and the foil width for these tracks should be as wide as possible. In applications where the ADC data outputs and control signals are connected to a continuously active microprocessor bus, it is possible to get errors in the conversion results. These errors are due to feedthrough from the microprocessor to the comparators. The problem can be eliminated by forcing the microprocessor into a wait state during conversion or by using three-state buffers to isolate the ADC data bus.

The LTC1406 has differential inputs to minimize noise coupling. Common mode noise on the A_{IN}^+ and A_{IN}^- leads will be rejected by the input CMRR. The LTC1406 will hold and convert the difference voltage between A_{IN}^+ and A_{IN}^-. The leads to A_{IN}^+ (Pin 7) and A_{IN}^- (Pin 8) should be kept as short as possible. In applications where this is not possible, the A_{IN}^+ and A_{IN}^- traces should be run side by side to equalize coupling.

Supply Bypassing

High quality, low series resistance ceramic, 10µF bypass capacitors should be used at the V_{DD}, V_{CM} and V_{REF} pins as shown in the Typical Application on the first page of this data sheet. Surface mount ceramic capacitors such as Murata GRM235Y5V106Z016 provide excellent bypassing in a small board space. Alternatively, 10µF tantalum capacitors in parallel with 0.1µF ceramic capacitors can be used. Bypass capacitors must be located as close to the pins as possible. The traces connecting the pins and the bypass capacitors must be kept short and should be made as wide as possible.

Example Layout

Figures 12a, 12b, 12c and 12d show the schematic and layout of an evaluation board. The layout demonstrates the proper use of decoupling capacitors and ground plane with a 2-layer printed circuit board.

APPLICATIONS INFORMATION

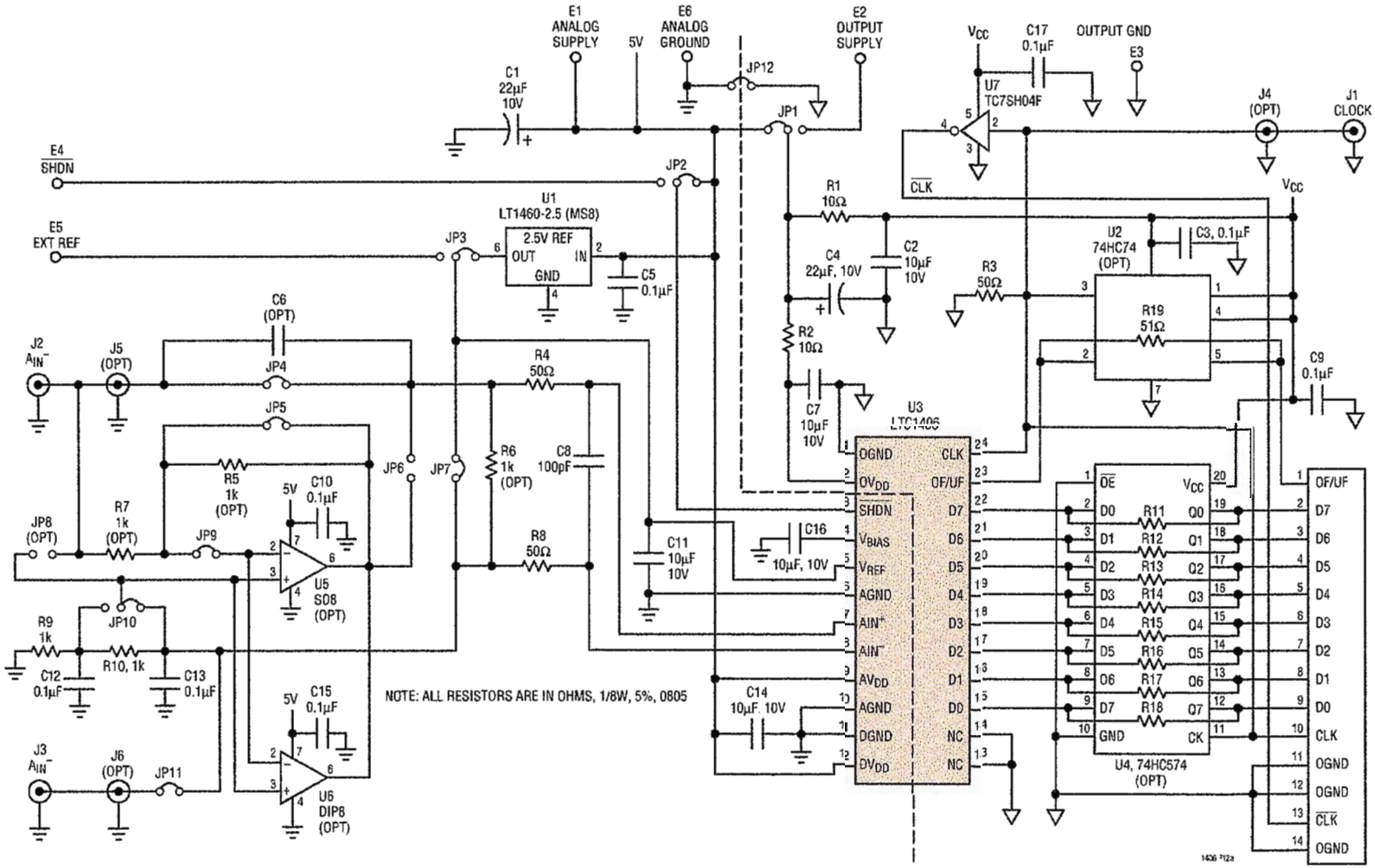


Figure 12a. Suggested Evaluation Circuit Schematic

APPLICATIONS INFORMATION

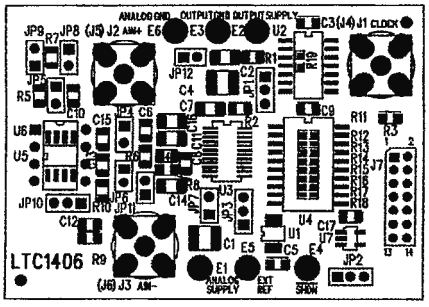


Figure 12b. Suggested Evaluation Circuit Board—Component Side Silkscreen

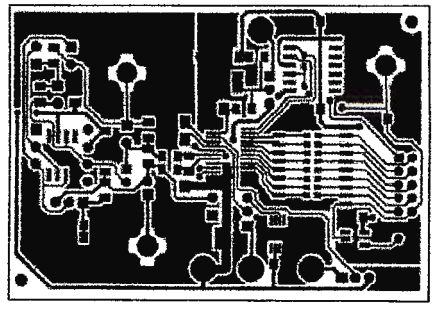


Figure 12c. Suggested Evaluation Circuit Board—Component Side Layout

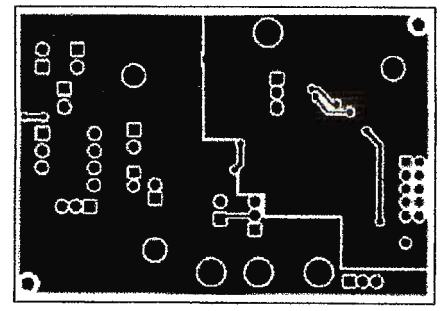
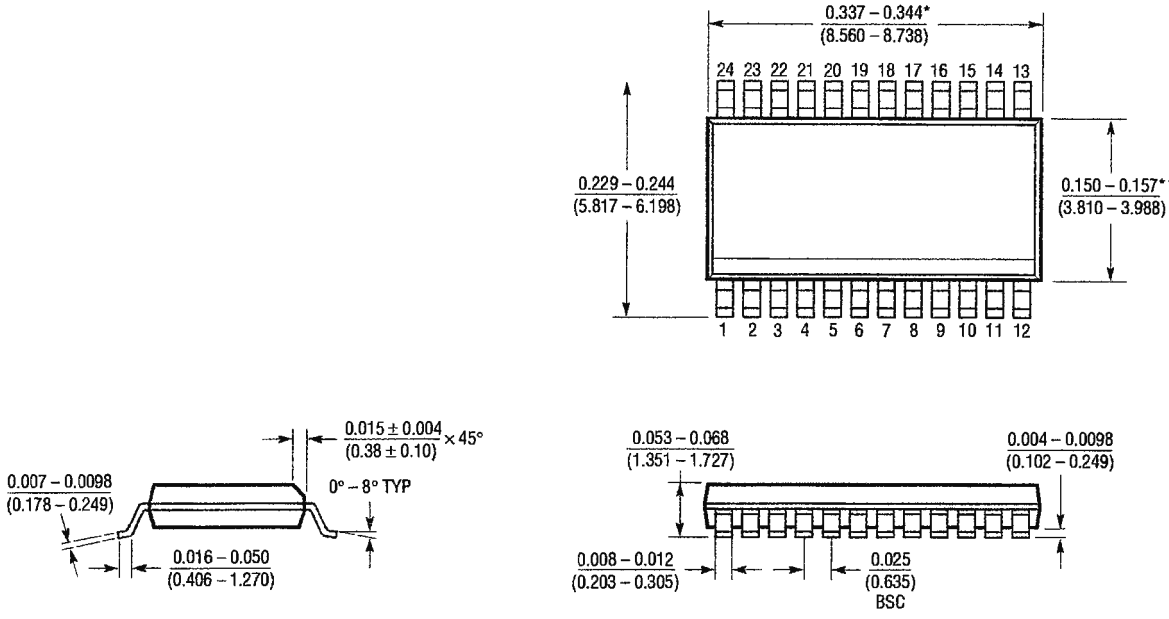


Figure 12d. Suggested Evaluation Circuit Board—Solder Side Layout

PACKAGE DESCRIPTION

Dimensions in inches (millimeters) unless otherwise noted.

GN Package
24-Lead Plastic SSOP (Narrow 0.150)
(LTC DWG # 05-08-1641)



* DIMENSION DOES NOT INCLUDE MOLD FLASH. MOLD FLASH SHALL NOT EXCEED 0.006" (0.152mm) PER SIDE

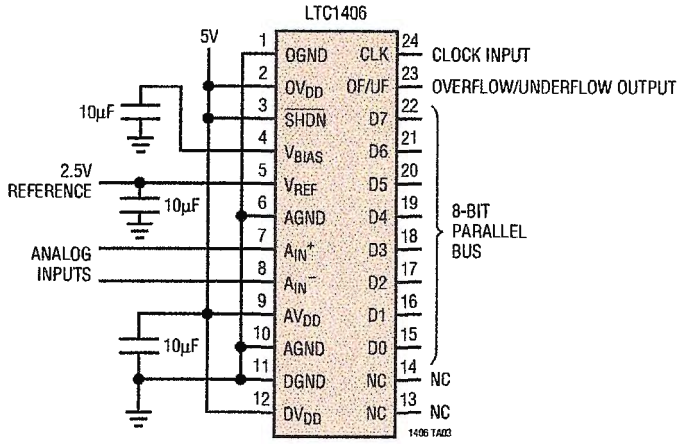
** DIMENSION DOES NOT INCLUDE INTERLEAD FLASH. INTERLEAD FLASH SHALL NOT EXCEED 0.010" (0.254mm) PER SIDE

GN24 (SSOP) 1197

Information furnished by Linear Technology Corporation is believed to be accurate and reliable. However, no responsibility is assumed for its use. Linear Technology Corporation makes no representation that the interconnection of its circuits as described herein will not infringe on existing patent rights.

TYPICAL APPLICATION

Low Power, 20MHz, 8-Bit Sampling ADC



RELATED PARTS

PART NUMBER	DESCRIPTION	COMMENTS
ADCs		
LTC1196/LTC1198	Single Supply, 8-Bit, 1Msps/750ksps ADCs	Single 3V or 5V Supply, Low Power, Serial Interface, SO-8 Package
LTC1197/LTC1199	Single Supply, 10-Bit, 500ksps/450ksps ADCs	Single 3V or 5V Supply, Low Power, Serial Interface, SO-8 Package
LTC1410	12-Bit, 1.25Msps Sampling ADC with Shutdown	Best Dynamic Performance, THD = 84dB and SINAD = 71dB at Nyquist
LTC1415	Single 5V, 12-Bit, 1.25Msps ADC	Single Supply 55mW Dissipation
LTC1419	14-Bit, 800ksps Sampling ADC with Shutdown	81.5dB SINAD, 150mW from ±5V Supplies
LTC1604	16-Bit, 333ksps ADC	90dB SINAD, 100dB THD, 250mW Dissipation
LTC1605	Single 5V, 16-Bit, 100ksps ADC	Low Power, ±10V Inputs
DACs		
LTC1446/LTC1446L	Dual 12-Bit V_{OUT} DACs in SO-8 Package	LTC1446: V_{CC} = 4.5V to 5.5V, V_{OUT} = 0V to 4.095V LTC1446L: V_{CC} = 2.7V to 5.5V, V_{OUT} = 0V to 2.5V
LTC1448	Dual 12-Bit Rail-to-Rail Output DAC in SO-8 Package	V_{CC} = 2.7V to 5.5V, Output Swings from GND to REF, REF Input Can Be Tied to V_{CC}
LTC1458/LTC1458L	Quad 12-Bit Rail-to-Rail Output DACs	LTC1458: V_{CC} = 4.5V to 5.5V, V_{OUT} 0V to 4.095V LTC1458L: V_{CC} = 2.7V to 5.5V, V_{OUT} = 0V to 2.5V

Linear Technology Corporation
1630 McCarthy Blvd., Milpitas, CA 95035-7417
(408)432-1900 • FAX: (408) 434-0507 • www.linear-tech.com
1406f LT/TP 0299 4K • PRINTED IN USA
© LINEAR TECHNOLOGY CORPORATION 1998

Winbond Electronics Corp.

32K × 8 HIGH SPEED CMOS STATIC RAM

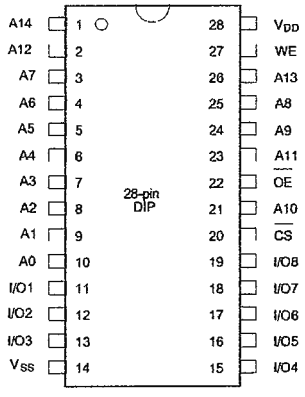
GENERAL DESCRIPTION

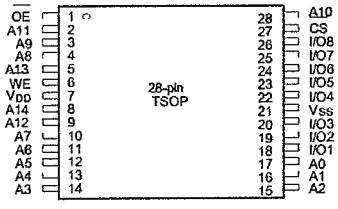
The W24257A is a high speed, low power CMOS static RAM organized as 32768 × 8 bits that operates on a single 5-volt power supply. This device is manufactured using Winbond's high performance CMOS technology.

FEATURES

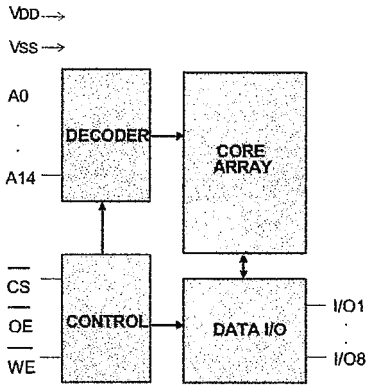
- High speed access time: 10/12/15/20 nS (max.)
- Low power consumption:
  - Active: 400 mW (typ.)
- Single +5V power supply
- Fully static operation
- All inputs and outputs directly TTL compatible
- Three-state outputs
- Available packages: 28-pin 300 mil SOJ, 330 mil SOP, skinny DIP and standard type one TSOP (8 mm × 13.4 mm)

PIN CONFIGURATIONS





BLOCK DIAGRAM



PIN DESCRIPTION

SYMBOL	DESCRIPTION
A0–A14	Address Inputs
I/O1–I/O8	Data Inputs/Outputs
$\overline{CS}$	Chip Select Input
$\overline{WE}$	Write Enable Input
$\overline{OE}$	Output Enable Input
VDD	Power Supply
VSS	Ground

Publication Release Date: May 1997
Revision A14

Winbond Electronics Corp.

DC CHARACTERISTICS

Absolute Maximum Ratings

PARAMETER	RATING	UNIT
Supply Voltage to V_{SS} Potential	-0.5 to +7.0	V
Input/Output to V_{SS} Potential	-0.5 to V_{DD} +0.5	V
Allowable Power Dissipation	1.0	W
Storage Temperature	-65 to +150	°C
Operating Temperature	0 to +70	°C

Note: Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

TRUTH TABLE

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	MODE	I/O1–I/O8	V_{DD} CURRENT
H	X	X	Not Selected	High Z	I_{SB}, I_{SB1}
L	H	H	Output Disable	High Z	I_{DD}
L	L	H	Read	Data Out	I_{DD}
L	X	L	Write	Data In	I_{DD}

OPERATING CHARACTERISTICS

(V_{DD} = 5V ±10%, V_{SS} = 0V, T_A = 0 to 70° C)

PARAMETER	SYM.	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
Input Low Voltage	V_{IL}	-		-0.5	-	+0.8	V
Input High Voltage	V_{IH}	-		+2.2	-	V_{DD} +0.5	V
Input Leakage Current	I_{LI}	V_{IN} = V_{SS} to V_{DD}		-10	-	+10	μA
Output Leakage Current	I_{LO}	$V_{I/O}$ = V_{SS} to V_{DD}, $\overline{CS}$ = V_{IH} or $\overline{OE}$ = V_{IH} or $\overline{WE}$ = V_{IL}		-10	-	+10	μA
Output Low Voltage	V_{OL}	I_{OL} = +8.0 mA		-	-	0.4	V
Output High Voltage	V_{OH}	I_{OH} = -4.0 mA		2.4	-	-	V
Operating Power Supply Current	I_{DD}	$\overline{CS}$ = V_{IL}, I/O = 0 mA Cycle = MIN Duty = 100%	10	-	-	170	mA
			12	-	-	160	mA
			15			150	mA
			20	-	-	140	mA
Standby Power Supply Current	I_{SB}	$\overline{CS}$ = V_{IH} Cycle = MIN, Duty = 100%		-	-	30	mA
	I_{SB1}	$\overline{CS}$ ≥ V_{DD} -0.2V		-	-	10	mA

Note: Typical characteristics are at V_{DD} = 5V, T_A = 25° C.

AC CHARACTERISTICS

(V_{DD} = 5V ±10%, V_{SS} = 0V, T_A = 0 to 70° C)

Read Cycle

PARAMETER	SYM.	W24257A-10		W24257A-12		W24257A-15		W24257A-20		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Read Cycle Time	T_{RC}	10	-	12	-	15	-	20	-	nS
Address Access Time	T_{AA}	-	10	-	12	-	15	-	20	nS
Chip Select Access Time	T_{ACS}	-	10	-	12	-	15	-	20	nS
Output Enable to Output Valid	T_{AOE}	-	5	-	6	-	7	-	10	nS
Chip Selection to Output in Low Z	T_{CLZ}*	3	-	3	-	3	-	3	-	nS
Output Enable to Output in Low Z	T_{OLZ}*	0	-	0	-	0	-	0	-	nS
Chip Deselection to Output in High Z	T_{CHZ}*	-	5	-	6	-	7	-	10	nS
Output Disable to Output in High Z	T_{OHZ}*	-	5	-	6	-	7	-	10	nS
Output Hold from Address Change	T_{OH}	3	-	3	-	3	-	3	-	nS

* These parameters are sampled but not 100% tested.

Write Cycle

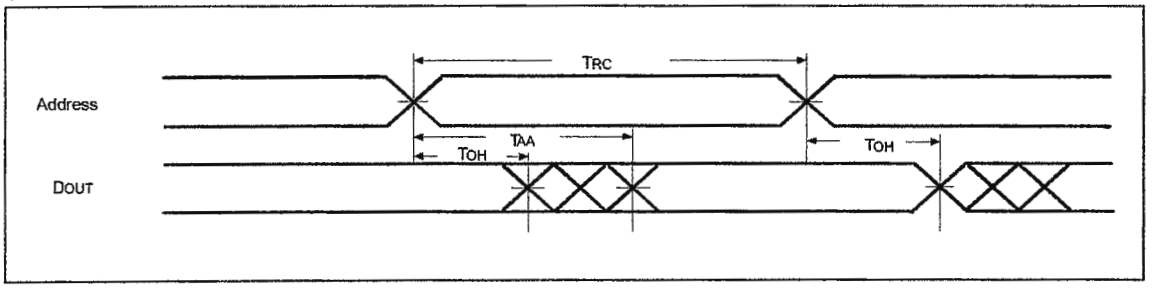
PARAMETER	SYM.	W24257A-10		W24257A-12		W24257A-15		W24257A-20		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Write Cycle Time	T_{WC}	10	-	12	-	15	-	20	-	nS
Chip Selection to End of Write	T_{CW}	9	-	10	-	13	-	17	-	nS
Address Valid to End of Write	T_{AW}	9	-	10	-	13	-	17	-	nS
Address Setup Time	T_{AS}	0	-	0	-	0	-	0	-	nS
Write Pulse Width	T_{WP}	9	-	10	-	10	-	12	-	nS
Write Recovery Time $\overline{CS}$, $\overline{WE}$	T_{WR}	0	-	0	-	0	-	0	-	nS
Data Valid to End of Write	T_{DW}	6	-	7	-	9	-	10	-	nS
Data Hold from End of Write	T_{DH}	0	-	0	-	0	-	0	-	nS
Write to Output in High Z	T_{WHZ}*	-	6	-	7	-	8	-	10	nS
Output Disable to Output in High Z	T_{OHZ}*	-	6	-	7	-	8	-	10	nS
Output Active from End of Write	T_{OW}	0	-	0	-	0	-	0	-	nS

* These parameters are sampled but not 100% tested.

TIMING WAVEFORMS

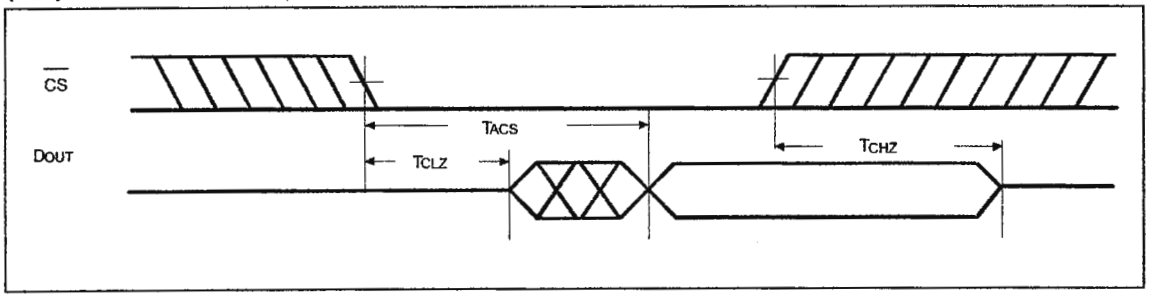
Read Cycle 1

(Address Controlled)



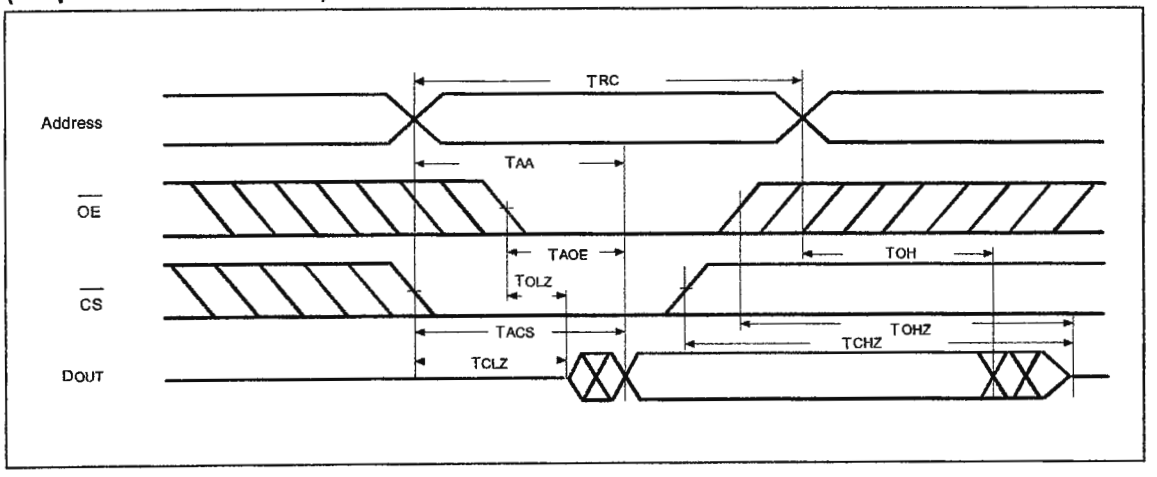
Read Cycle 2

(Chip Select Controlled)



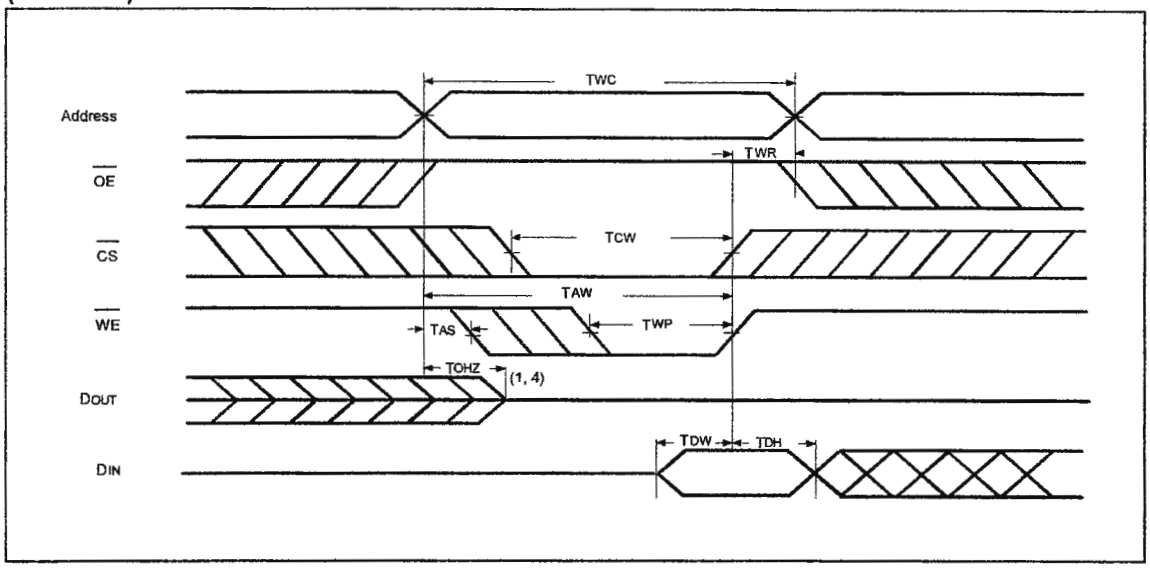
Read Cycle 3

(Output Enable Controlled)

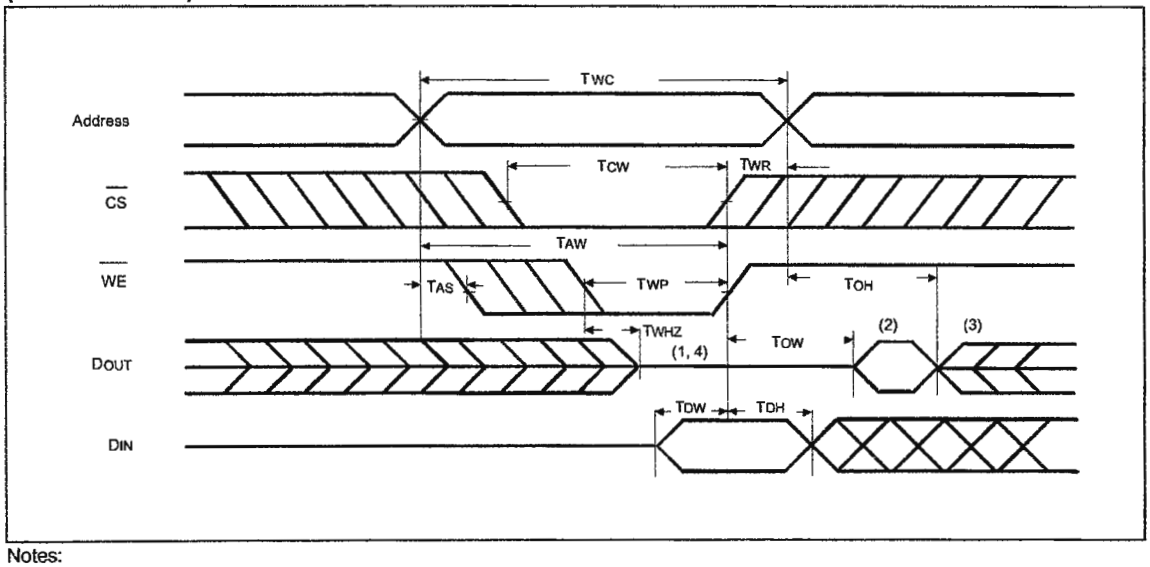


Timing Waveforms, continued

Write Cycle 1
(OE Clock)



Write Cycle 2
(OE = VIL Fixed)



Notes:
1. During this period, I/O pins are in the output state, so input signals of opposite phase to the outputs should not be applied.
2. The data output from DOUT are the same as the data written to DIN during the write cycle.
3. DOUT provides the read data for the next address.
4. Transition is measured ±500 mV from steady state with CL = 5 pF. This parameter is guaranteed but not 100% tested.

Winbond Electronics Corp.

ORDERING INFORMATION

PART NO.	ACCESS TIME (nS)	OPERATING CURRENT MAX. (mA)	STANDBY CURRENT MAX. (mA)	PACKAGE
W24257AK-10	10	170	10	300 mil skinny DIP
W24257AK-12	12	160	10	300 mil skinny DIP
W24257AK-15	15	150	10	300 mil skinny DIP
W24257AK-20	20	140	10	300 mil skinny DIP
W24257AJ-10	10	170	10	300 mil SOJ
W24257AJ-12	12	160	10	300 mil SOJ
W24257AJ-15	15	150	10	300 mil SOJ
W24257AJ-20	20	140	10	300 mil SOJ
W24257AS-10	10	170	10	330 mil SOP
W24257AS-12	12	160	10	330 mil SOP
W24257AS-15	15	150	10	330 mil SOP
W24257AS-20	20	140	10	330 mil SOP
W24257AQ-10	10	170	10	Standard type one TSOP
W24257AQ-12	12	160	10	Standard type one TSOP
W24257AQ-15	15	150	10	Standard type one TSOP
W24257AQ-20	20	140	10	Standard type one TSOP

Notes:

1. Winbond reserves the right to make changes to its products without prior notice.
2. Purchasers are responsible for performing appropriate quality assurance testing on products intended for use in applications where personal injury might occur as a consequence of product failure.

19-0271; Rev 1; 1/00

MAXIM

Precision, 8-Channel/Dual 4-Channel, High-Performance, CMOS Analog Multiplexers

MAX308/MAX309

General Description

The MAX308/MAX309 precision, monolithic, CMOS analog multiplexers (muxes) offer low on-resistance (less than 100Ω), which is matched to within 5Ω between channels and remains flat over the specified analog signal range (7Ω max). They also offer low leakage over temperature (NO-off leakage current less than 5nA at +85°C) and fast switching speeds (transition time less than 250ns). The MAX308 is a single-ended 1-of-8 device, and the MAX309 is a differential 2-of-4 device.

The MAX308/MAX309 are fabricated with Maxim's improved 44V silicon-gate process. Design improvements yield extremely low charge injection (less than 10pC) and guarantee electrostatic discharge protection greater than 2000V.

These muxes operate with a single +4.5V to +30V supply or bipolar ±4.5V to ±20V supplies, while retaining TTL/CMOS-logic input compatibility and fast switching. CMOS inputs provide reduced input loading. These improved parts are plug-in upgrades for the industry-standard DG408, DG409, DG508A, and DG509A.

Applications

Sample-and-Hold Circuits
Automatic Test Equipment
Heads-Up Displays
Guidance and Control Systems
Military Radios
Communications Systems
Battery-Operated Systems
PBX, PABX
Audio Signal Routing

Features

- **Guaranteed On-Resistance Match Between Channels, <5Ω Max**
- **Low On-Resistance, <100Ω Max**
- **Guaranteed Flat On-Resistance over Specified Signal Range, 7Ω Max**
- **Guarateed Low Charge Injection, <10pC**
- **NO-Off Leakage Current <5nA at +85°C**
- **COM-Off Leakage Current <20nA at +85°C**
- **ESD Protection >2000V**
- **Plug-In Upgrade for Industry-Standard DG408/DG409/DG508A/DG509A**
- **Single-Supply Operation (+4.5V to +30V) Bipolar-Supply Operation (±4.5V to ±20V)**
- **Low Power Consumption, <300µW**
- **Rail-to-Rail Signal Handling**
- **TTL/CMOS-Logic Compatible**

Ordering Information

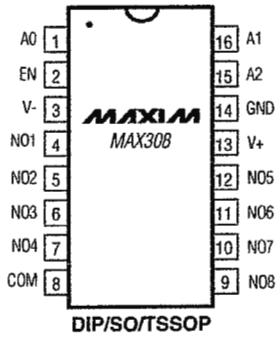
PART	TEMP. RANGE	PIN-PACKAGE
MAX308CUE	0°C to +70°C	16 TSSOP
MAX308CSE	0°C to +70°C	16 SO
MAX308CPE	0°C to +70°C	16 Plastic DIP
MAX308C/D	0°C to +70°C	Dice*
MAX308EUE	-40°C to +85°C	16 TSSOP
MAX308ESE	-40°C to +85°C	16 SO
MAX308EPE	-40°C to +85°C	16 Plastic DIP
MAX308EJE	-40°C to +85°C	16 CERDIP
MAX308MJE	-55°C to +125°C	16 CERDIP**

Ordering Information continued at end of data sheet.

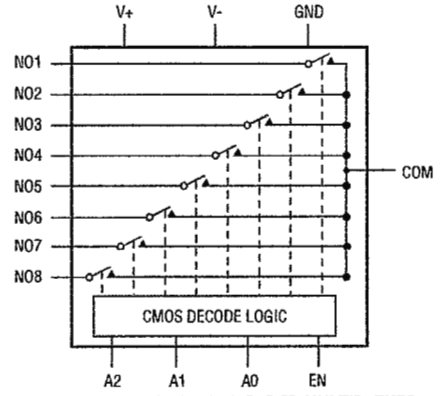
** Contact factory for dice specifications.*
*** Contact factory for availability.*

Pin Configurations/Functional Diagrams/Truth Tables

TOP VIEW



DIP/SO/TSSOP



MAX308 8-CHANNEL SINGLE-ENDED MULTIPLEXER

Continued at end of data sheet.

For price, delivery, and to place orders, please contact Maxim Distribution at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

Precision, 8-Channel/Dual 4-Channel, High-Performance, CMOS Analog Multiplexers

ABSOLUTE MAXIMUM RATINGS

Voltage Referenced to V-
- V+ ..... -0.3V, 44V
- GND ..... -0.3V, 25V

Digital Inputs, NO, COM (Note 1) ..... (V- - 2V) to (V+ + 2V) or 30mA, (whichever occurs first)

Continuous Current (any terminal) 30mA

Peak Current, NO or COM (pulsed at 1ms, 10% duty cycle max) 100mA

Continuous Power Dissipation (T_A = +70°C)
- TSSOP (derate 6.7mW/°C above +70°C) 457m
- Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
- Narrow SO (derate 8.70mW/°C above +70°C) 696mW
- CERDIP (derate 10.00mW/°C above +70°C) 800mW

Operating Temperature Ranges
- MAX30_C_ _ ..... 0°C to +70°C
- MAX30_E_ _ ..... -40°C to +85°C
- MAX30_MJE ..... -55°C to +125°C

Storage Temperature Range ..... -65°C to +150°C

Lead Temperature (soldering, 10s) ..... +300°C

Note 1: Signals on NO, COM, EN, A0, A1, or A2 exceeding V+ or V- are clamped by internal diodes. Limit forward current to maximum current ratings.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—Dual Supplies

(V+ = +15V, V- = -15V, GND = 0V, V_{AH} = +2.4V, V_{AL} = +0.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS				MIN	TYP (Note 2)	MAX	UNITS
SWITCH									
Analog Signal Range	V_{NO}, V_{COM}	(Note 3)				-15		15	V
On-Resistance	R_{ON}	I_{NO} = -1.0mA, V_{COM} = ±10V		T_A = +25°C			60	100	Ω
				T_A = T_{MIN} to T_{MAX}				125	
On-Resistance Matching Between Channels	ΔR_{ON}	I_{NO} = -1.0mA, V_{COM} = ±10V (Note 4)		T_A = +25°C			1.5	5	Ω
				T_A = T_{MIN} to T_{MAX}				8	
On-Resistance Flatness	R_{FLAT}	I_{NO} = -1.0mA, V_{COM} = ±5V or 0V		T_A = +25°C			1.8	7	Ω
				T_A = T_{MIN} to T_{MAX}				10	
NO-Off Leakage Current (Note 5)	$I_{NO(OFF)}$	V_{COM} = +10V, V_{NO} = ±10V, V_{EN} = 0V		T_A = +25°C		-0.5	0.01	0.5	nA
				T_A = T_{MIN} to T_{MAX}	C, E	-2.5		2.5	
					M	-5.0		5.0	
COM-Off Leakage Current (Note 5)	$I_{COM(OFF)}$	V_{NO} = ±10V, V_{COM} = +10V, V_{EN} = 0V	MAX308	T_A = +25°C		-0.75	0.02	0.75	nA
				T_A = T_{MIN} to T_{MAX}	C, E	-10		10	
					M	-20		20	
		V_{NO} = +10V, V_{COM} = ±10V, V_{EN} = 0V	MAX309	T_A = +25°C		-0.75	0.02	0.75	
				T_A = T_{MIN} to T_{MAX}	C, E	-5		5	
					M	-10		10	
COM-On Leakage Current (Note 5)	$I_{COM(ON)}$	V_{COM} = ±10V, V_{NO} = ±10V, sequence each switch on	MAX308	T_A = +25°C		-0.75	0.02	0.75	nA
				T_A = T_{MIN} to T_{MAX}	C, E	-10		10	
					M	-20		20	
			MAX309	T_A = +25°C		-0.75	0.02	0.75	
				T_A = T_{MIN} to T_{MAX}	C, E	-5		5	
					M	-10		10	

ELECTRICAL CHARACTERISTICS—Dual Supplies (continued)

($V+$ = +15V, $V-$ = -15V, GND = 0V, V_{AH} = +2.4V, V_{AL} = +0.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS			MIN	TYP (Note 2)	MAX	UNITS
INPUT								
Input Current with Input Voltage High	I_{AH}	V_A = 2.4V or 15V			-1.0		1.0	µA
Input Current with Input Voltage Low	I_{AL}	V_{EN} = 0V or 2.4V, V_A = 0V			-1.0		1.0	µA
SUPPLY								
Power-Supply Range					±4.5		±20	V
Positive Supply Current	I+	V_{EN} = V_A = 0V or 4.5V		T_A = +25°C		16	30	µA
				T_A = T_{MIN} to T_{MAX}			75	
		V_{EN} = 2.4V, $V_{A(ALL)}$ = 0V or 2.4V		T_A = +25°C		0.075	0.5	mA
				T_A = T_{MIN} to T_{MAX}				
Negative Supply Current	I-	V_{EN} = 2.4V, $V_{A(ALL)}$ = 0V or 2.4V		T_A = +25°C	-1		1	µA
				T_A = T_{MIN} to T_{MAX}	-10		10	
DYNAMIC								
Transistion Time	t_{TRANS}	Figure 2		T_A = +25°C		85	175	ns
				T_A = T_{MIN} to T_{MAX}			250	
Break-Before-Make Interval	t_{OPEN}	Figure 4		T_A = +25°C	10	40		ns
Enable Turn-On Time	$t_{ON(EN)}$	Figure 3		T_A = +25°C		85	150	ns
				T_A = T_{MIN} to T_{MAX}			225	
Enable Turn-Off Time	$t_{OFF(EN)}$	Figure 3		T_A = +25°C			150	ns
				T_A = T_{MIN} to T_{MAX}			300	
Charge Injection (Note 3)	Q	C_L = 1.0nF, V_{NO} = 0V, R_S = 0Ω, Figure 5		T_A = +25°C		2	10	pC
Off Isolation (Note 6)	V_{ISO}	V_{EN} = 0V, R_L = 1kΩ, f = 100kHz, Figure 6		T_A = +25°C		-75		dB
Crosstalk Between Channels	V_{CT}	V_{EN} = 2.4V, f = 100kHz, V_{GEN} = $1V_{P-P}$, R_L = 1kΩ, Figure 7		T_A = +25°C		-92		dB
Logic Input Capacitance	C_{IN}	f = 1MHz		T_A = +25°C		8		pF
NO-Off Capacitance	$C_{NO(OFF)}$	f = 1MHz, V_{EN} = V_{NO} = 0V, Figure 8		T_A = +25°C		3		pF
COM-Off Capacitance	$C_{COM(OFF)}$	f = 1MHz, V_{EN} = 0.8V V_{COM} = 0V, Figure 8	MAX308	T_A = +25°C		26		pF
			MAX309			14		
COM-On Capacitance	$C_{COM(ON)}$	f = 1MHz, V_{EN} = 2.4V V_{COM} = 0V, Figure 8	MAX308	T_A = +25°C		37		pF
			MAX309			25		

Precision, 8-Channel/Dual 4-Channel, High-Performance, CMOS Analog Multiplexers

ELECTRICAL CHARACTERISTICS—Single Supply

($V+ = +12V$, $V- = 0V$, GND = 0V, $V_{AH} = +2.4V$, $V_{AL} = +0.8V$, $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP (Note 2)	MAX	UNITS
SWITCH							
Analog Signal Range	V_{NO}, V_{COM}	(Note 3)		0		12	V
On-Resistance	R_{ON}	$I_{NO} = -1.0mA$ $V_{COM} = 3V$ or 10V	$T_A = +25°C$		120	175	Ω
DYNAMIC							
Transition Time (Note 3)	t_{TRANS}	$V_{NO1} = 8V$, $V_{NO8} = 0V$, $V_{IN} = 2.4V$, Figure 2	$T_A = +25°C$		115	450	ns
Enable Turn-On Time (Note 3)	$t_{ON(EN)}$	$V_{INH} = 2.4V$, $V_{INL} = 0V$, $V_{NO1} = 5V$, Figure 3	$T_A = +25°C$		100	600	ns
Enable Turn-Off Time (Note 3)	$t_{OFF(EN)}$	$V_{INH} = 2.4V$, $V_{INL} = 0V$, $V_{NO1} = 5V$, Figure 3	$T_A = +25°C$		75	300	ns
Charge Injection (Note 3)	Q	$C_L = 1.0nF$, $V_{NO} = 0V$, $R_S = 0Ω$	$T_A = +25°C$		2	10	pC

Note 2: The algebraic convention where the most negative value is a minimum and the most positive value a maximum is used in this data sheet.

Note 3: Guaranteed by design.

Note 4: $\Delta R_{ON} = R_{ON(MAX)} - R_{ON(MIN)}$. On-resistance match between channels and flatness are guaranteed only with specified voltages. Flatness is defined as the difference between the maximum and minimum value of on-resistance as measured at the extremes of the specified analog signal range.

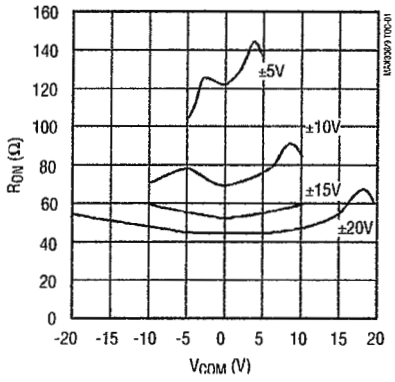
Note 5: Leakage parameters are 100% tested at the maximum rated hot temperature and guaranteed by correlation at +25°C.

Note 6: Off isolation = 20log V_{COM}/V_{NO}, where V_{COM} = output and V_{NO} = input to off switch.

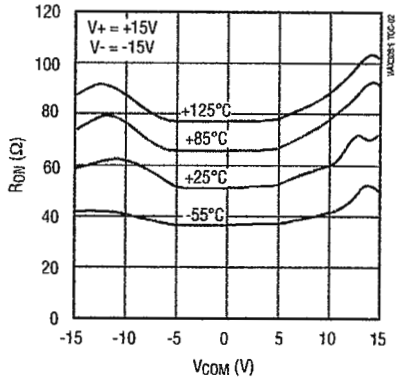
Typical Operating Characteristics

(T_A = +25°C, unless otherwise noted.)

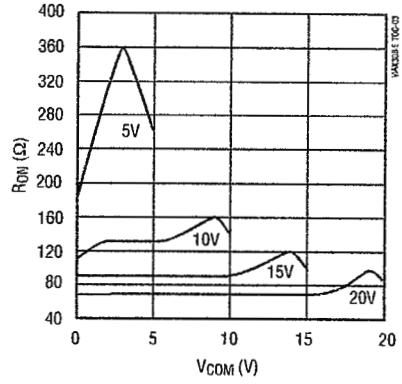
ON-RESISTANCE vs. V_{COM} (DUAL SUPPLIES)



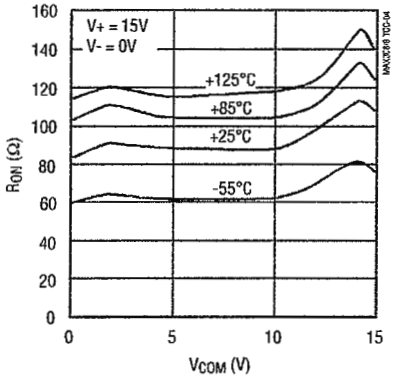
ON-RESISTANCE vs. V_{COM} AND TEMPERATURE (DUAL SUPPLIES)



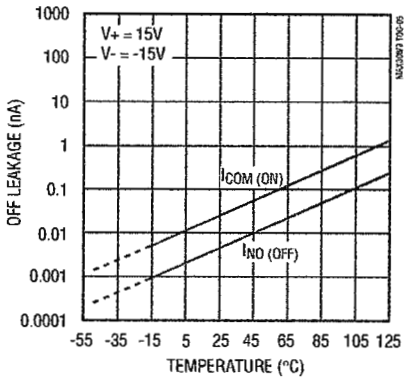
ON-RESISTANCE vs. V_{COM} (SINGLE SUPPLY)



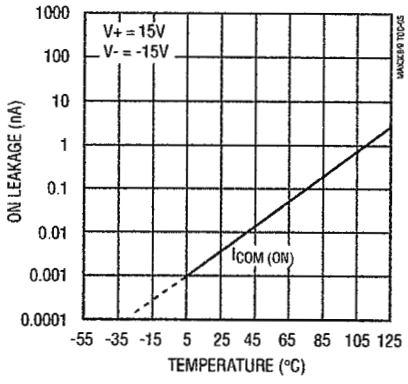
ON-RESISTANCE vs. V_{COM} AND TEMPERATURE (SINGLE SUPPLY)



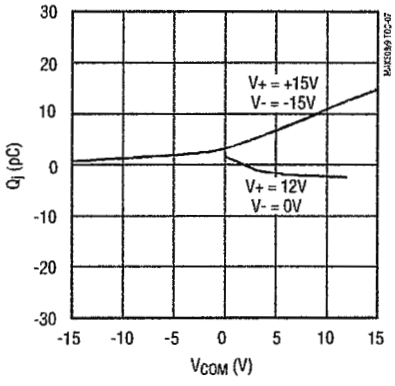
OFF LEAKAGE vs. TEMPERATURE



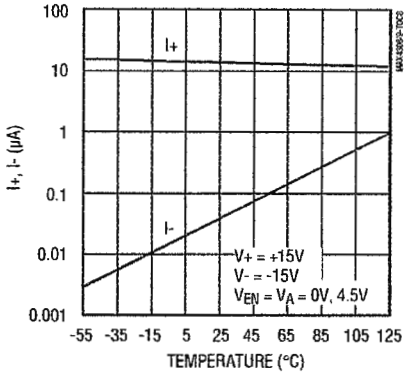
ON LEAKAGE vs. TEMPERATURE



CHARGE INJECTION vs. V_{COM}



SUPPLY CURRENT vs. TEMPERATURE



Precision, 8-Channel/Dual 4-Channel, High-Performance, CMOS Analog Multiplexers

Pin Description

PIN		NAME	FUNCTION
MAX308	MAX309		
1, 15, 16	—	A0, A2, A1	Address Inputs
—	1, 16	A0, A1	Address Inputs
2	2	EN	Enable Input
3	3	V-	Negative Supply Voltage Input
4–7	—	NO1–NO4	Analog Inputs—bidirectional
—	4–7	NO1A–NO4A	Analog Inputs—bidirectional
8	—	COM	Analog Output—bidirectional
—	8, 9	COMA, COMB	Analog Outputs—bidirectional
9–12	—	NO8–NO5	Analog Inputs—bidirectional
—	10–13	NO4B–NO1B	Analog Inputs—bidirectional
13	14	V+	Positive Supply Voltage Input
14	15	GND	Ground

Applications Information

Operation with Supply Voltages Other than 15V

Using supply voltages less than ±15V will reduce the analog signal range. The MAX308/MAX309 switches operate with ±4.5V to ±20V bipolar supplies or with a +4.5V to +30V single supply. Connect V- to GND when operating with a single supply. Both device types can also operate with unbalanced supplies, such as +24V and -5V. The *Typical Operating Characteristics* graphs show typical on-resistance with 20V, 15V, 10V, and 5V supplies. (Switching times increase by a factor of two or more for operation at 5V.)

Overvoltage Protection

Proper power-supply sequencing is recommended for all CMOS devices. Do not exceed the absolute maximum ratings, because stresses beyond the listed ratings may cause permanent damage to the devices. Always sequence V+ on first, then V-, followed by the logic inputs, NO, or COM. If power-supply sequencing is not possible, add two small signal diodes in series with supply pins for overvoltage protection (Figure 1). Adding diodes reduces the analog signal range to 1V below V+ and 1V above V-, but does not affect the devices' low switch resistance and low leakage characteristics. Device operation is unchanged, and the difference between V+ and V- should not exceed +44V.

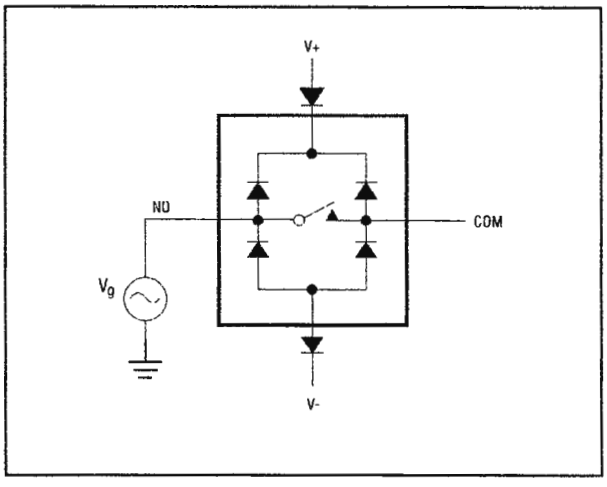
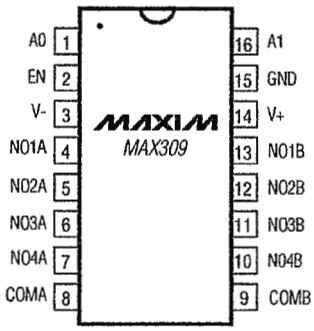


Figure 1. Overvoltage Protection Using External Blocking Diodes

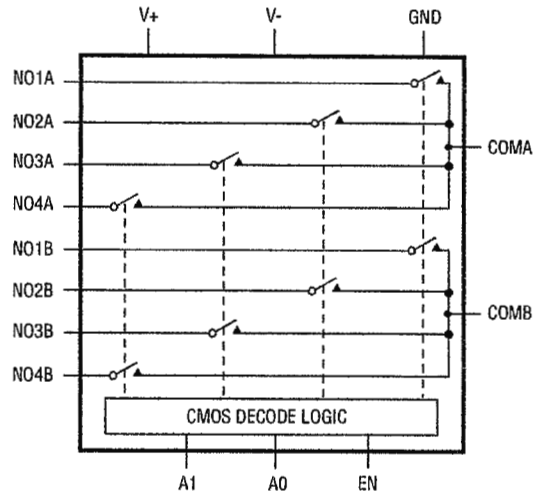
Precision, 8-Channel/Dual 4-Channel, High-Performance, CMOS Analog Multiplexers

Pin Configurations/Functional Diagrams/Truth Tables (continued)

TOP VIEW



DIP/SO/TSSOP



MAX309 4-CHANNEL DIFFERENTIAL MULTIPLEXER

A2	A1	A0	EN	ON SWITCH
X	X	X	0	None
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	1	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	1	8

MAX308

LOGIC "0" $V_{AL} \leq 0.8V$, LOGIC "1" $V_{AH} \geq 2.4V$

A1	A0	EN	ON SWITCH
X	X	0	None
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

MAX309

LOGIC "0" $V_{AL} \leq 0.8V$, LOGIC "1" $V_{AH} \geq 2.4V$

14-Bit and 16-Bit Parallel Low Glitch Multiplying DACs with 4-Quadrant Resistors

FEATURES

- **True 16-Bit Performance Over Industrial Temperature Range**
- **DNL and INL: 1LSB Max**
- **On-Chip 4-Quadrant Resistors Allow Precise 0V to 10V, 0V to –10V or ±10V Outputs**
- **Pin Compatible 14- and 16-Bit Parts**
- Asynchronous Clear Pin
  - LTC1591/LTC1597: Reset to Zero Scale
  - LTC1591-1/LTC1597-1: Reset to Midscale
- **Glitch Impulse < 2nV-s**
- 28-Lead SSOP Package
- Low Power Consumption: 10µW Typ
- Power-On Reset

APPLICATIONS

- Process Control and Industrial Automation
- Direct Digital Waveform Generation
- Software-Controlled Gain Adjustment
- Automatic Test Equipment

DESCRIPTION

The LTC®1591/LTC1597 are pin compatible, parallel input 14-bit and 16-bit multiplying current output DACs that operate from a single 5V supply. INL and DNL are accurate to 1LSB over the industrial temperature range in both 2- and 4-quadrant multiplying modes. True 16-bit 4-quadrant multiplication is achieved with on-chip 4-quadrant multiplication resistors.

These DACs include an internal deglitcher circuit that reduces the glitch impulse to less than 2nV-s (typ). The asynchronous $\overline{CLR}$ pin resets the LTC1591/LTC1597 to zero scale and LTC1591-1/LTC1597-1 to midscale.

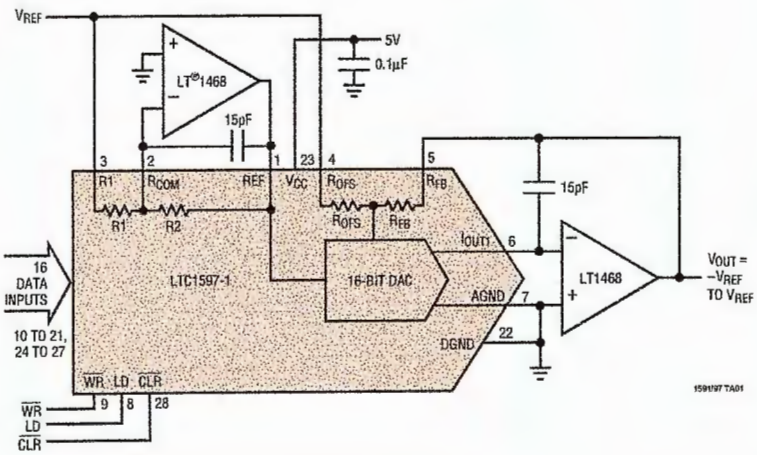
The LTC1591/LTC1597 are available in 28-pin SSOP and PDIP packages and are specified over the industrial temperature range.

For serial interface 16-bit current output DACs refer to the LTC1595/LTC1596 data sheet.

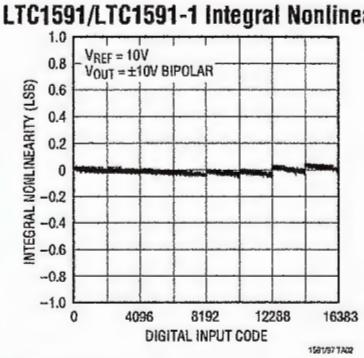
LT, LTC and LT are registered trademarks of Linear Technology Corporation.

TYPICAL APPLICATION

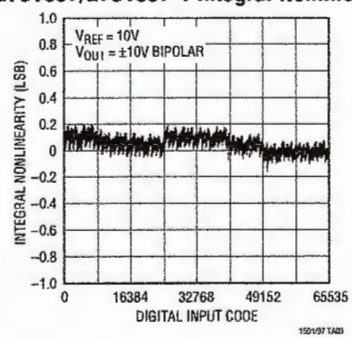
16-Bit, 4-Quadrant Multiplying DAC with a Minimum of External Components



LTC1591/LTC1591-1 Integral Nonlinearity



LTC1597/LTC1597-1 Integral Nonlinearity

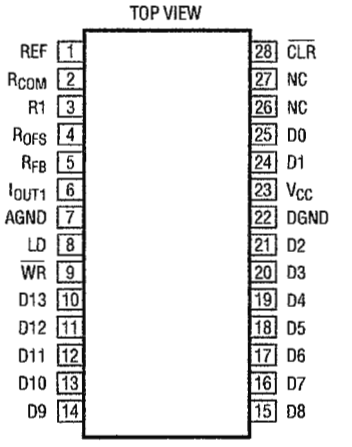


ABSOLUTE MAXIMUM RATINGS (Note 1)

V_{CC} to AGND ..... −0.5V to 7V
V_{CC} to DGND ..... −0.5V to 7V
AGND to DGND ..... V_{CC} + 0.5V
DGND to AGND ..... V_{CC} + 0.5V
REF, R_{OFS}, R_{FB}, R1, R_{COM} to AGND, DGND ..... ±25V
Digital Inputs to DGND ..... −0.5V to (V_{CC} + 0.5V)
I_{OUT1} to AGND ..... −0.5V to(V_{CC} + 0.5V)
Maximum Junction Temperature ..... 125°C

Operating Temperature Range
LTC1591C/LTC1591-1C
LTC1597C/LTC1597-1C ..... 0°C to 70°C
LTC1591I/LTC1591-1I
LTC1597I/LTC1597-1I ..... −40°C to 85°C
Storage Temperature Range ..... −65°C to 150°C
Lead Temperature (Soldering, 10 sec) ..... 300°C

PACKAGE/ORDER INFORMATION

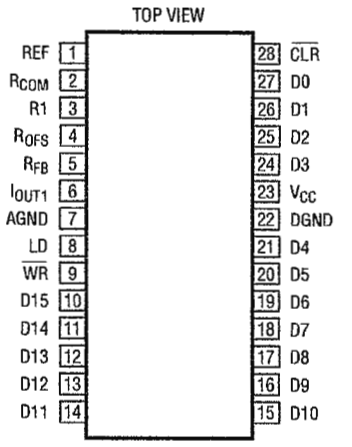


G PACKAGE 28-LEAD PLASTIC SSOP
N PACKAGE 28-LEAD NARROW PDIP

T_{JMAX} = 125°C, θ_{JA} = 95°C/W (G)
T_{JMAX} = 125°C, θ_{JA} = 70°C/W (N)

ORDER PART NUMBER

LTC1591CG
LTC1591CN
LTC1591IG
LTC1591IN
LTC1591-1CG
LTC1591-1CN
LTC1591-1IG
LTC1591-1IN



G PACKAGE 28-LEAD PLASTIC SSOP
N PACKAGE 28-LEAD NARROW PDIP

T_{JMAX} = 125°C, θ_{JA} = 95°C/W (G)
T_{JMAX} = 125°C, θ_{JA} = 70°C/W (N)

ORDER PART NUMBER

LTC1597ACG
LTC1597ACN
LTC1597BCG
LTC1597BCN
LTC1597-1ACG
LTC1597-1ACN
LTC1597-1BCG
LTC1597-1BCN
LTC1597AIG
LTC1597AIN
LTC1597BIG
LTC1597BIN
LTC1597-1AIG
LTC1597-1AIN
LTC1597-1BIG
LTC1597-1BIN

Consult factory for Military grade parts.

ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ±10%, V_{REF} = 10V, I_{OUT1} = AGND = DGND = 0V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		LTC1591/-1 MIN	LTC1591/-1 TYP	LTC1591/-1 MAX	LTC1597B/-1B MIN	LTC1597B/-1B TYP	LTC1597B/-1B MAX	LTC1597A/-1A MIN	LTC1597A/-1A TYP	LTC1597A/-1A MAX	UNITS
Accuracy													
	Resolution		●	14			16			16			Bits
	Monotonicity		●	14			16			16			Bits
INL	Integral Nonlinearity	(Note 2) T_A = 25°C				±1			±2		±0.25	±1	LSB
		T_{MIN} to T_{MAX}	●			±1			±2		±0.35	±1	LSB
DNL	Differential Nonlinearity	T_A = 25°C				±1			±1		±0.2	±1	LSB
		T_{MIN} to T_{MAX}	●			±1			±1		±0.2	±1	LSB
GE	Gain Error	Unipolar Mode											
		(Note 3) T_A = 25°C				±4			±16		2	±16	LSB
		T_{MIN} to T_{MAX}	●			±6			±24		3	±16	LSB
		Bipolar Mode											
		(Note 3) T_A = 25°C				±4			±16		2	±16	LSB
		T_{MIN} to T_{MAX}	●			±6			±24		3	±16	LSB
	Gain Temperature Coefficient	(Note 4) ΔGain/ΔTemperature	●		1	2		1	2		1	2	ppm/°C
	Bipolar Zero-Scale Error	T_A = 25°C				±3			±10			±5	LSB
		T_{MIN} to T_{MAX}	●			±5			±16			±8	LSB
I_{LKG}	OUT1 Leakage Current	(Note 5) T_A = 25°C				±5			±5			±5	nA
		T_{MIN} to T_{MAX}	●			±15			±15			±15	nA
PSRR	Power Supply Rejection Ratio	V_{CC} = 5V ±10	●		±0.1	±1		±0.4	±2		±0.4	±2	LSB/V

V_{CC} = 5V ±10%, V_{REF} = 10V, I_{OUT1} = AGND = DGND = 0V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Reference Input							
R_{REF}	DAC Input Resistance (Unipolar)	(Note 6)	●	4.5	6	10	kΩ
R1/R2	R1/R2 Resistance (Bipolar)	(Notes 6, 13)	●	9	12	20	kΩ
R_{OFS}, R_{FB}	Feedback and Offset Resistances	(Note 6)	●	9	12	20	kΩ
AC Performance (Note 4)							
	Output Current Settling Time	(Notes 7, 8)			1		µs
	Midscale Glitch Impulse	(Note 12)			2		nV-s
	Digital-to-Analog Glitch Impulse	(Note 9)			1		nV-s
	Multiplying Feedthrough Error	V_{REF} = ±10V, 10kHz Sine Wave			1		mV_{P-P}
THD	Total Harmonic Distortion	(Note 10)			108		dB
	Output Noise Voltage Density	(Note 11)			10		nV/√Hz
	Harmonic Distortion (Digital Waveform Generation)	Unipolar Mode (Note 14)					
		2nd Harmonic			94		dB
		3rd Harmonic			101		dB
		SFDR			94		dB
		Bipolar Mode (Note 14)					
		2nd Harmonic			94		dB
		3rd Harmonic			101		dB
		SFDR			94		dB

ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ±10%, V_{REF} = 10V, I_{OUT1} = AGND = DGND = 0V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Analog Outputs (Note 4)							
C_{OUT}	Output Capacitance (Note 4)	DAC Register Loaded to All 1s: C_{OUT1} DAC Register Loaded to All 0s: C_{OUT1}	● ●		115 70	130 80	pF pF
Digital Inputs							
V_{IH}	Digital Input High Voltage		●	2.4			V
V_{IL}	Digital Input Low Voltage		●			0.8	V
I_{IN}	Digital Input Current		●		0.001	±1	µA
C_{IN}	Digital Input Capacitance	(Note 4) V_{IN} = 0V	●			8	pF
Timing Characteristics							
t_{DS}	Data to $\overline{WR}$ Setup Time		●	60	20		ns
t_{DH}	Data to $\overline{WR}$ Hold Time		●	0	–12		ns
t_{WR}	$\overline{WR}$ Pulse Width		●	60	25		ns
t_{LD}	LD Pulse Width		●	110	55		ns
t_{CLR}	Clear Pulse Width		●	60	40		ns
t_{LWD}	$\overline{WR}$ to LD Delay Time		●	0			ns
Power Supply							
V_{DD}	Supply Voltage		●	4.5	5	5.5	V
I_{DD}	Supply Current	Digital Inputs = 0V or V_{CC}	●			10	µA

The ● denotes specifications that apply over the full operating temperature range.

Note 1: Absolute Maximum Values are those beyond which the life of a device may be impaired.

Note 2: ±1LSB = ±0.006% of full scale = ±61ppm of full scale for the LTC1591/LTC1591-1. ±1LSB = ±0.0015% of full scale = ±15.3ppm of full scale for the LTC1597/LTC1597-1.

Note 3: Using internal feedback resistor.

Note 4: Guaranteed by design, not subject to test.

Note 5: $I_{(OUT1)}$ with DAC register loaded to all 0s.

Note 6: Typical temperature coefficient is 100ppm/°C.

Note 7: I_{OUT1} load = 100Ω in parallel with 13pF.

Note 8: To 0.006% for a full-scale change, measured from the rising edge of LD for the LTC1591/LTC1591-1. To 0.0015% for a full-scale change, measured from the rising edge of LD for the LTC1597/LTC1597-1.

Note 9: V_{REF} = 0V. DAC register contents changed from all 0s to all 1s or all 1s to all 0s.

Note 10: V_{REF} = $6V_{RMS}$ at 1kHz. DAC register loaded with all 1s.

Note 11: Calculation from $e_n = \sqrt{4kTRB}$ where: k = Boltzmann constant (J/°K), R = resistance (Ω), T = temperature (°K), B = bandwidth (Hz).

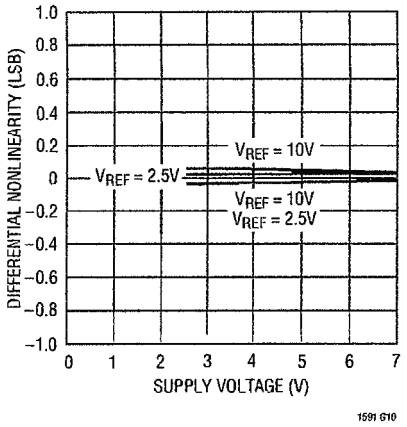
Note 12: Midscale transition code: 01 1111 1111 1111 to 10 0000 0000 0000 for the LTC1591/LTC1591-1 and 0111 1111 1111 1111 to 1000 0000 0000 0000 for the LTC1597/LTC1597-1.

Note 13: R1 and R2 are measured between R1 and R_{COM}, REF and R_{COM}.

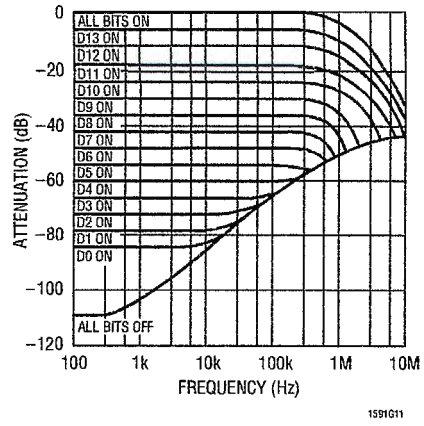
Note 14: Measured using the LT1468 op amp in unipolar mode for I/V converter and LT1468 I/V and LT1001 reference inverter in bipolar mode. Sample Rate = 50kHz, Signal Frequency = 1kHz, V_{REF} = 5V, T_A = 25°C.

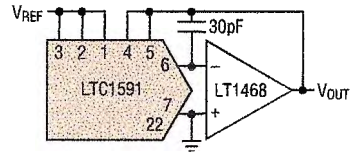
TYPICAL PERFORMANCE CHARACTERISTICS (LTC1591)

Differential Nonlinearity vs Supply Voltage in Bipolar Mode

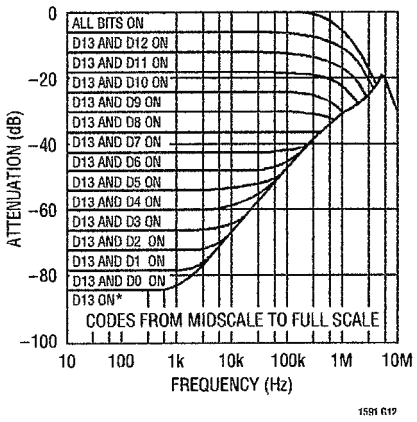


Unipolar Multiplying Mode Frequency Response vs Digital Code

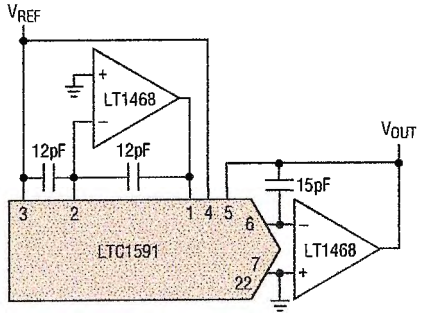




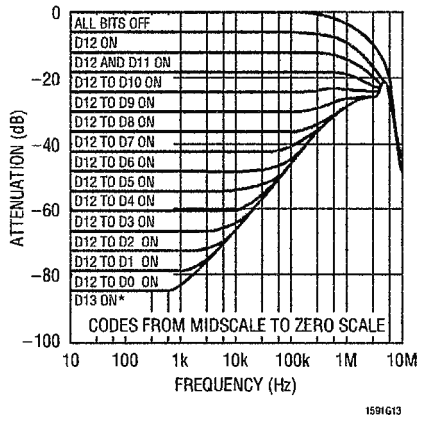
Bipolar Multiplying Mode Frequency Response vs Digital Code



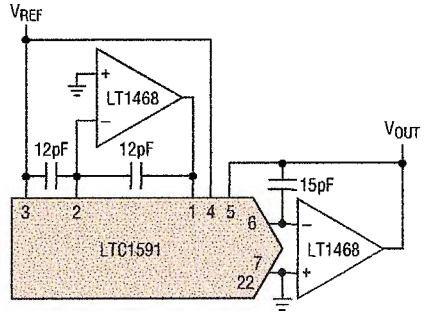
*DAC ZERO VOLTAGE OUTPUT LIMITED BY BIPOLAR ZERO ERROR TO –84dB TYPICAL (–70dB MAX)



Bipolar Multiplying Mode Frequency Response vs Digital Code



*DAC ZERO VOLTAGE OUTPUT LIMITED BY BIPOLAR ZERO ERROR TO –84dB TYPICAL (–70dB MAX)



PIN FUNCTIONS

LTC1591

REF (Pin 1): Reference Input and 4-Quadrant Resistor R2. Typically ±10V, accepts up to ±25V. In 2-Quadrant mode this is the reference input. In 4-quadrant mode, this pin is driven by external inverting reference amplifier.

R_{COM} (Pin 2): Center Tap Point of the Two 4-Quadrant Resistors R1 and R2. Normally tied to the inverting input of an external amplifier in 4-quadrant operation, otherwise shorted to the REF pin. See Figures 1a and 2a.

R1 (Pin 3): 4-Quadrant Resistor R1. In 2-quadrant operation short to the REF pin. In 4-quadrant mode tie to R_{OFS} (Pin 4).

R_{OFS} (Pin 4): Bipolar Offset Resistor. Typically swings ±10V, accepts up to ±25V. In 2-quadrant operation tie to R_{FB}. In 4-quadrant operation tie to R1.

R_{FB} (Pin 5): Feedback Resistor. Normally tied to the output of the current to voltage converter op amp. Swings to $\pm V_{REF}$. V_{REF} is typically ±10V.

I_{OUT1} (Pin 6): DAC Current Output. Tie to the inverting input of the current to voltage converter op amp.

AGND (Pin 7): Analog Ground. Tie to ground.

LD (Pin 8): DAC Digital Input Load Control Input. When LD is taken to a logic high, data is loaded from the input register into the DAC register, updating the DAC output.

$\overline{WR}$ (Pin 9): DAC Digital Write Control Input. When $\overline{WR}$ is taken to a logic low, data is loaded from the digital input pins into the 14-bit wide input register.

DB13 to D2 (Pins 10 to 21): Digital Input Data Bits.

DGND (Pin 22): Digital Ground. Tie to ground.

V_{CC} (Pin 23): The Positive Supply Input. $4.5V \leq V_{CC} \geq 5.5V$. Requires a bypass capacitor to ground.

DB1, DB0 (Pins 24, 25): Digital Input Data Bits.

NC (Pins 26, 27): No Connect.

$\overline{CLR}$ (Pin 28): Digital Clear Control Function for the DAC. When $\overline{CLR}$ is taken to a logic low, it sets the DAC output and all internal registers to zero code for the LTC1591 and midscale code for the LTC1591-1.

LTC1597

REF (Pin 1): Reference Input and 4-Quadrant Resistor R2. Typically ±10V, accepts up to ±25V. In 2-Quadrant mode this is the reference input. In 4-quadrant mode, this pin is driven by external inverting reference amplifier.

R_{COM} (Pin 2): Center Tap Point of the Two 4-Quadrant Resistors R1 and R2. Normally tied to the inverting input of an external amplifier in 4-quadrant operation, otherwise shorted to the REF pin. See Figures 1b and 2b.

R1 (Pin 3): 4-Quadrant Resistor R1. In 2-quadrant operation short to the REF pin. In 4-quadrant mode tie to R_{OFS} (Pin 4).

R_{OFS} (Pin 4): Bipolar Offset Resistor. Typically swings ±10V, accepts up to ±25V. In 2-quadrant operation tie to R_{FB}. In 4-quadrant operation tie to R1.

R_{FB} (Pin 5): Feedback Resistor. Normally tied to the output of the current to voltage converter op amp. Swings to $\pm V_{REF}$. V_{REF} is typically ±10V.

I_{OUT1} (Pin 6): DAC Current Output. Tie to the inverting input of the current to voltage converter op amp.

AGND (Pin 7): Analog Ground. Tie to ground.

LD (Pin 8): DAC Digital Input Load Control Input. When LD is taken to a logic high, data is loaded from the input register into the DAC register, updating the DAC output.

$\overline{WR}$ (Pin 9): DAC Digital Write Control Input. When $\overline{WR}$ is taken to a logic low, data is loaded from the digital input pins into the 16-bit wide input register.

DB15 to D4 (Pins 10 to 21): Digital Input Data Bits.

DGND (Pin 22): Digital Ground. Tie to ground.

V_{CC} (Pin 23): The Positive Supply Input. $4.5V \leq V_{CC} \geq 5.5V$. Requires a bypass capacitor to ground.

DB3 to DB0 (Pins 24 to 27): Digital Input Data Bits.

$\overline{CLR}$ (Pin 28): Digital Clear Control Function for the DAC. When $\overline{CLR}$ is taken to a logic low, it sets the DAC output and all internal registers to zero code for the LTC1597 and midscale code for the LTC1597-1.

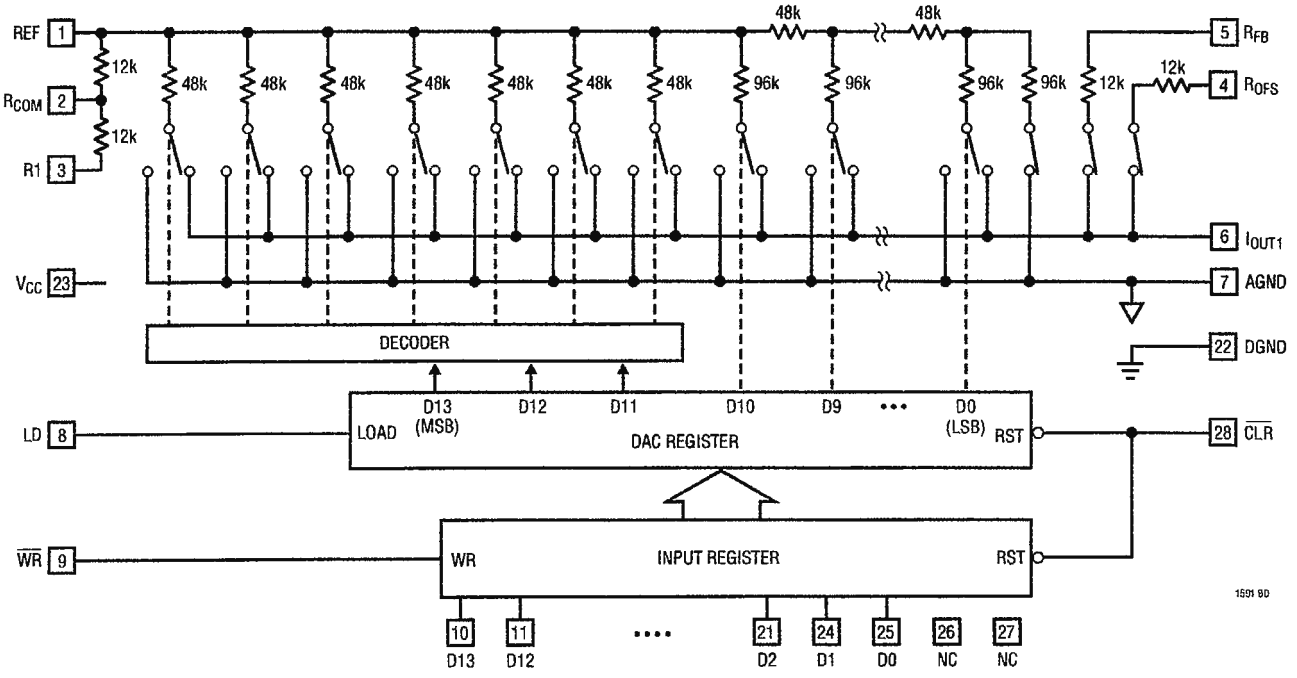
TRUTH TABLE

Table 1

CONTROL INPUTS $\overline{CLR}$	$\overline{WR}$	LD	REGISTER OPERATION
0	X	X	Reset Input and DAC Register to All 0s for LTC1591/LTC1597 and Midscale for LTC1591-1/LTC1597-1 (Asynchronous Operation)
1	0	0	Load Input Register with All 14/16 Data Bits
1	1	1	Load DAC Register with the Contents of the Input Register
1	0	1	Input and DAC Register Are Transparent
1	⎍	⎍	CLK = LD and $\overline{WR}$ Tied Together. The 14/16 Data Bits Are Loaded into the Input Register on the Falling Edge of the CLK and Then Loaded into the DAC Register on the Rising Edge of the CLK
1	1	0	No Register Operation

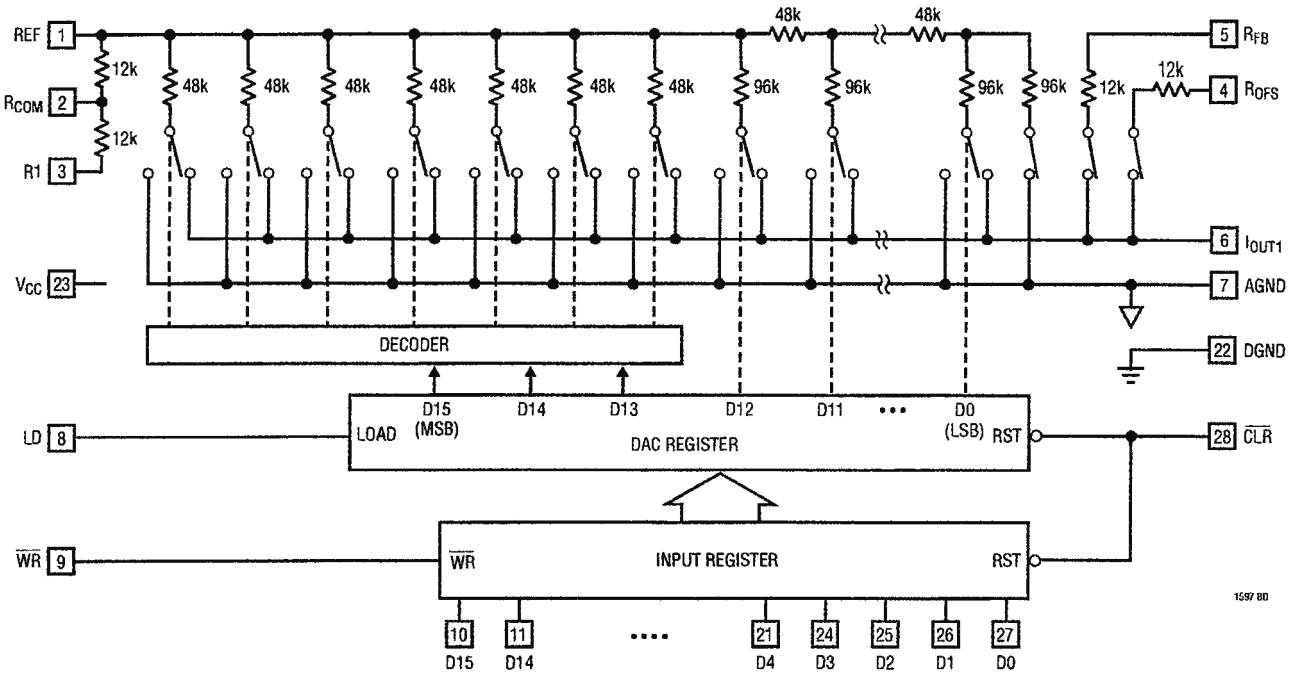
BLOCK DIAGRAMS

LTC1591

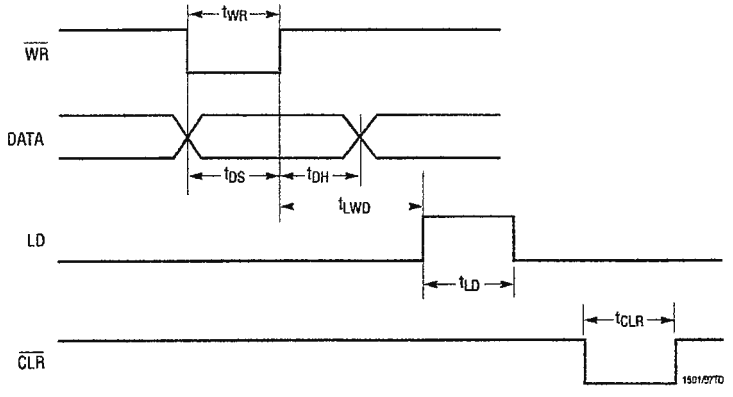


BLOCK DIAGRAMS

LTC1597

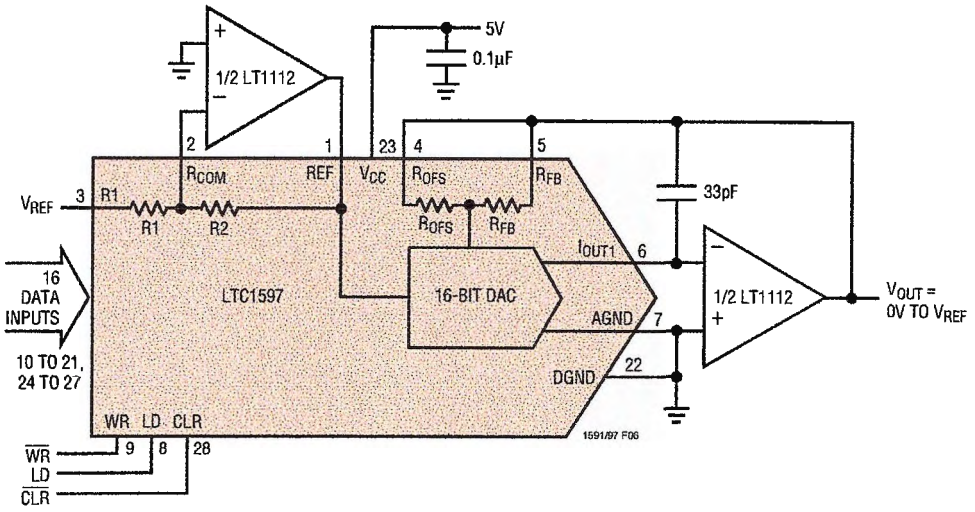


TIMING DIAGRAM



TYPICAL APPLICATIONS

Noninverting Unipolar Operation (2-Quadrant Multiplication) V_{OUT} = 0V to V_{REF}



LT1355/LT1356

Dual and Quad 12MHz, 400V/µs Op Amps

FEATURES

- **12MHz Gain Bandwidth**
- **400V/µs Slew Rate**
- **1.25mA Maximum Supply Current per Amplifier**
- Unity-Gain Stable
- C-Load™ Op Amp Drives All Capacitive Loads
- 10nV/√Hz Input Noise Voltage
- 800µV Maximum Input Offset Voltage
- 300nA Maximum Input Bias Current
- 70nA Maximum Input Offset Current
- 12V/mV Minimum DC Gain, R_L=1k
- 230ns Settling Time to 0.1%, 10V Step
- 280ns Settling Time to 0.01%, 10V Step
- ±12.5V Minimum Output Swing into 500Ω
- ±3V Minimum Output Swing into 150Ω
- Specified at ±2.5V, ±5V, and ±15V

APPLICATIONS

- Wideband Amplifiers
- Buffers
- Active Filters
- Data Acquisition Systems
- Photodiode Amplifiers

DESCRIPTION

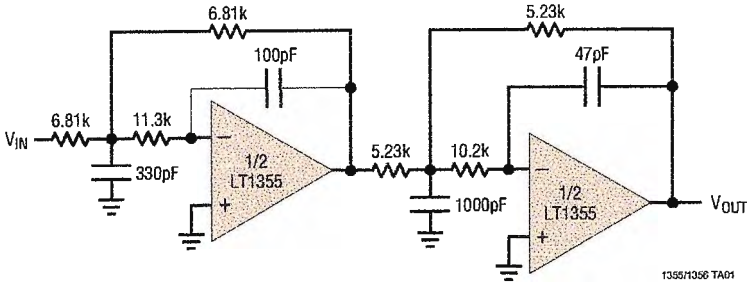
The LT1355/LT1356 are dual and quad low power high speed operational amplifiers with outstanding AC and DC performance. The amplifiers feature much lower supply current and higher slew rate than devices with comparable bandwidth. The circuit topology is a voltage feedback amplifier with matched high impedance inputs and the slewing performance of a current feedback amplifier. The high slew rate and single stage design provide excellent settling characteristics which make the circuit an ideal choice for data acquisition systems. Each output drives a 500Ω load to ±12.5V with ±15V supplies and a 150Ω load to ±3V on ±5V supplies. The amplifiers are stable with any capacitive load making them useful in buffer applications.

The LT1355/LT1356 are members of a family of fast, high performance amplifiers using this unique topology and employing Linear Technology Corporation's advanced bipolar complementary processing. For a single amplifier version of the LT1355/LT1356 see the LT1354 data sheet. For higher bandwidth devices with higher supply currents see the LT1357 through LT1365 data sheets. Bandwidths of 25MHz, 50MHz, and 70MHz are available with 2mA, 4mA, and 6mA of supply current per amplifier. Singles, duals, and quads of each amplifier are available.

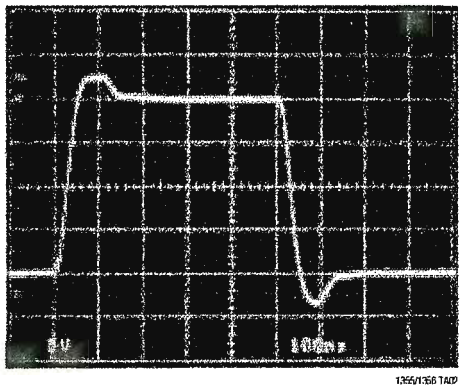
LT, LTC and LT are registered trademarks of Linear Technology Corporation.
C-Load is a trademark of Linear Technology Corporation.

TYPICAL APPLICATION

100kHz, 4th Order Butterworth Filter



A_V = –1 Large-Signal Response



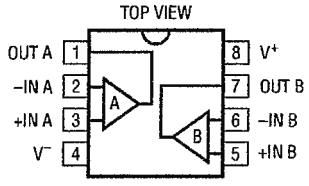
1355/1356 TA02

ABSOLUTE MAXIMUM RATINGS (Note 1)

Total Supply Voltage (V^+ to V^-) 36V
Differential Input Voltage
(Transient Only) (Note 2) ..... ±10V
Input Voltage ..... $\pm V_S$
Output Short-Circuit Duration (Note 3) ..... Indefinite

Operating Temperature Range (Note 7) .. –40°C to 85°C
Specified Temperature Range (Note 8) ... –40°C to 85°C
Maximum Junction Temperature (See Below)
Plastic Package ..... 150°C
Storage Temperature Range ..... –65°C to 150°C
Lead Temperature (Soldering, 10 sec) ..... 300°C

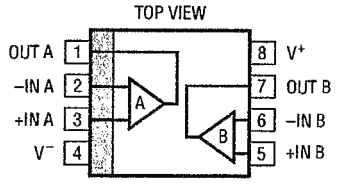
PACKAGE/ORDER INFORMATION



N8 PACKAGE
8-LEAD PDIP

T_{JMAX} = 150°C, θ_{JA} = 130°C/W

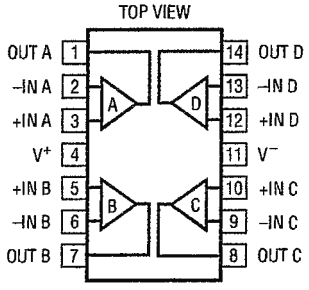
ORDER PART NUMBER
LT1355CN8



S8 PACKAGE
8-LEAD PLASTIC SO

T_{JMAX} = 150°C, θ_{JA} = 190°C/W

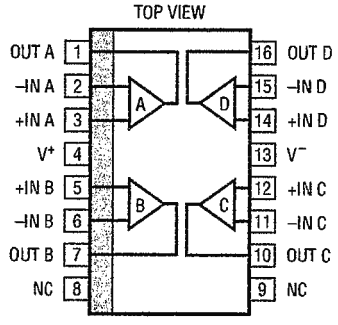
ORDER PART NUMBER
LT1355CS8
S8 PART MARKING
1355



N PACKAGE
14-LEAD PDIP

T_{JMAX} = 150°C, θ_{JA} = 110°C/W

ORDER PART NUMBER
LT1356CN



S PACKAGE
16-LEAD PLASTIC SO

T_{JMAX} = 150°C, θ_{JA} = 150°C/W

ORDER PART NUMBER
LT1356CS

Consult factory for Industrial and Military grade parts.

ELECTRICAL CHARACTERISTICS T_A = 25°C, V_{CM} = 0V unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	V_{SUPPLY}	MIN	TYP	MAX	UNITS
V_{OS}	Input Offset Voltage		±15V ±5V ±2.5V		0.3 0.3 0.4	0.8 0.8 1.0	mV mV mV
I_{OS}	Input Offset Current		±2.5V to ±15V		20	70	nA
I_B	Input Bias Current		±2.5V to ±15V		80	300	nA
e_n	Input Noise Voltage	f = 10kHz	±2.5V to ±15V		10		$nV/\sqrt{Hz}$
i_n	Input Noise Current	f = 10kHz	±2.5V to ±15V		0.6		$pA/\sqrt{Hz}$
R_{IN}	Input Resistance	V_{CM} = ±12V	±15V	70	160		MΩ
	Input Resistance	Differential	±15V		11		MΩ
C_{IN}	Input Capacitance		±15V		3		pF

ELECTRICAL CHARACTERISTICS $T_A = 25°C$, $V_{CM} = 0V$ unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	V_{SUPPLY}	MIN	TYP	MAX	UNITS
	Input Voltage Range +		±15V	12.0	13.4		V
			±5V	2.5	3.5		V
			±2.5V	0.5	1.1		V
	Input Voltage Range −		±15V		−13.2	−12.0	V
			±5V		−3.4	−2.5	V
			±2.5V		−0.9	−0.5	V
CMRR	Common Mode Rejection Ratio	$V_{CM} = ±12V$	±15V	83	97		dB
		$V_{CM} = ±2.5V$	±5V	78	84		dB
		$V_{CM} = ±0.5V$	±2.5V	68	75		dB
PSRR	Power Supply Rejection Ratio	$V_S = ±2.5V$ to ±15V		92	106		dB
A_{VOL}	Large-Signal Voltage Gain	$V_{OUT} = ±12V$, $R_L = 1k$	±15V	12	36		V/mV
		$V_{OUT} = ±10V$, $R_L = 500Ω$	±15V	5	15		V/mV
		$V_{OUT} = ±2.5V$, $R_L = 1k$	±5V	12	36		V/mV
		$V_{OUT} = ±2.5V$, $R_L = 500Ω$	±5V	5	15		V/mV
		$V_{OUT} = ±2.5V$, $R_L = 150Ω$	±5V	1	4		V/mV
		$V_{OUT} = ±1V$, $R_L = 500Ω$	±2.5V	5	20		V/mV
V_{OUT}	Output Swing	$R_L = 1k$, $V_{IN} = ±40mV$	±15V	13.3	13.8		±V
		$R_L = 500Ω$, $V_{IN} = ±40mV$	±15V	12.5	13.0		±V
		$R_L = 500Ω$, $V_{IN} = ±40mV$	±5V	3.5	4.0		±V
		$R_L = 150Ω$, $V_{IN} = ±40mV$	±5V	3.0	3.3		±V
		$R_L = 500Ω$, $V_{IN} = ±40mV$	±2.5V	1.3	1.7		±V
I_{OUT}	Output Current	$V_{OUT} = ±12.5V$	±15V	25	30		mA
		$V_{OUT} = ±3V$	±5V	20	25		mA
I_{SC}	Short-Circuit Current	$V_{OUT} = 0V$, $V_{IN} = ±3V$	±15V	30	42		mA
SR	Slew Rate	$A_V = -2$, (Note 4)	±15V	200	400		V/µs
			±5V	70	120		V/µs
	Full Power Bandwidth	10V Peak, (Note 5)	±15V		6.4		MHz
		3V Peak, (Note 5)	±5V		6.4		MHz
GBW	Gain Bandwidth	$f = 200kHz$, $R_L = 2k$	±15V	9.0	12.0		MHz
			±5V	7.5	10.5		MHz
			±2.5V		9.0		MHz
t_r, t_f	Rise Time, Fall Time	$A_V = 1$, 10%-90%, 0.1V	±15V		14		ns
			±5V		17		ns
	Overshoot	$A_V = 1$, 0.1V	±15V		20		%
			±5V		18		%
	Propagation Delay	50% V_{IN} to 50% V_{OUT}, 0.1V	±15V		16		ns
			±5V		19		ns
t_s	Settling Time	10V Step, 0.1%, $A_V = -1$	±15V		230		ns
		10V Step, 0.01%, $A_V = -1$	±15V		280		ns
		5V Step, 0.1%, $A_V = -1$	±5V		240		ns
		5V Step, 0.01%, $A_V = -1$	±5V		380		ns
	Differential Gain	$f = 3.58MHz$, $A_V = 2$, $R_L = 1k$	±15V		2.2		%
			±5V		2.1		%
	Differential Phase	$f = 3.58MHz$, $A_V = 2$, $R_L = 1k$	±15V		3.1		Deg
			±5V		3.1		Deg
R_O	Output Resistance	$A_V = 1$, $f = 100kHz$	±15V		0.7		Ω
	Channel Separation	$V_{OUT} = ±10V$, $R_L = 500Ω$	±15V	100	113		dB
I_S	Supply Current	Each Amplifier	±15V		1.0	1.25	mA
		Each Amplifier	±5V		0.9	1.20	mA

ELECTRICAL CHARACTERISTICS

The ● denotes the specifications which apply over the temperature range $0°C \le T_A \le 70°C$, V_{CM} = 0V unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	V_{SUPPLY}		MIN	TYP	MAX	UNITS
V_{OS}	Input Offset Voltage		±15V	●			1.0	mV
			±5V	●			1.0	mV
			±2.5V	●			1.2	mV
	Input V_{OS} Drift	(Note 6)	±2.5V to ±15V	●		5	8	µV/°C
I_{OS}	Input Offset Current		±2.5V to ±15V	●			100	nA
I_B	Input Bias Current		±2.5V to ±15V	●			450	nA
CMRR	Common Mode Rejection Ratio	V_{CM} = ±12V	±15V	●	81			dB
		V_{CM} = ±2.5V	±5V	●	77			dB
		V_{CM} = ±0.5V	±2.5V	●	67			dB
PSRR	Power Supply Rejection Ratio	V_S = ±2.5V to ±15V		●	90			dB
A_{VOL}	Large-Signal Voltage Gain	V_{OUT} = ±12V, R_L = 1k	±15V	●	10.0			V/mV
		V_{OUT} = ±10V, R_L = 500Ω	±15V	●	3.3			V/mV
		V_{OUT} = ±2.5V, R_L = 1k	±5V	●	10.0			V/mV
		V_{OUT} = ±2.5V, R_L = 500Ω	±5V	●	3.3			V/mV
		V_{OUT} = ±2.5V, R_L = 150Ω	±5V	●	0.6			V/mV
		V_{OUT} = ±1V, R_L = 500Ω	±2.5V	●	3.3			V/mV
V_{OUT}	Output Swing	R_L = 1k, V_{IN} = ±40mV	±15V	●	13.2			±V
		R_L = 500Ω, V_{IN} = ±40mV	±15V	●	12.0			±V
		R_L = 500Ω, V_{IN} = ±40mV	±5V	●	3.4			±V
		R_L = 150Ω, V_{IN} = ±40mV	±5V	●	2.8			±V
		R_L = 500Ω, V_{IN} = ±40mV	±2.5V	●	1.2			±V
I_{OUT}	Output Current	V_{OUT} = ±12V	±15V	●	24.0			mA
		V_{OUT} = ±2.8V	±5V	●	18.7			mA
I_{SC}	Short-Circuit Current	V_{OUT} = 0V, V_{IN} = ±3V	±15V	●	24			mA
SR	Slew Rate	A_V = −2, (Note 4)	±15V	●	150			V/µs
			±5V	●	60			V/µs
GBW	Gain Bandwidth	f = 200kHz, R_L = 2k	±15V	●	7.5			MHz
			±5V	●	6.0			MHz
	Channel Separation	V_{OUT} = ±10V, R_L = 500Ω	±15V	●	98			dB
I_S	Supply Current	Each Amplifier	±15V	●			1.45	mA
		Each Amplifier	±5V	●			1.40	mA

The ● denotes the specifications which apply over the temperature range $-40°C \le T_A \le 85°C$, V_{CM} = 0V unless otherwise noted. (Note 8)

SYMBOL	PARAMETER	CONDITIONS	V_{SUPPLY}		MIN	TYP	MAX	UNITS
V_{OS}	Input Offset Voltage		±15V	●			1.5	mV
			±5V	●			1.5	mV
			±2.5V	●			1.7	mV
	Input V_{OS} Drift	(Note 6)	±2.5V to ±15V	●		5	8	µV/°C
I_{OS}	Input Offset Current		±2.5V to ±15V	●			200	nA
I_B	Input Bias Current		±2.5V to ±15V	●			550	nA
CMRR	Common Mode Rejection Ratio	V_{CM} = ±12V	±15V	●	80			dB
		V_{CM} = ±2.5V	±5V	●	76			dB
		V_{CM} = ±0.5V	±2.5V	●	66			dB
PSRR	Power Supply Rejection Ratio	V_S = ±2.5V to ±15V		●	90			dB
A_{VOL}	Large-Signal Voltage Gain	V_{OUT} = ±12V, R_L = 1k	±15V	●	7.0			V/mV
		V_{OUT} = ±10V, R_L = 500Ω	±15V	●	1.7			V/mV
		V_{OUT} = ±2.5V, R_L = 1k	±5V	●	7.0			V/mV
		V_{OUT} = ±2.5V, R_L = 500Ω	±5V	●	1.7			V/mV

ELECTRICAL CHARACTERISTICS

The ● denotes the specifications which apply over the temperature range $-40°C \le T_A \le 85°C$, $V_{CM} = 0V$ unless otherwise noted. (Note 8)

SYMBOL	PARAMETER	CONDITIONS	V_{SUPPLY}		MIN	TYP	MAX	UNITS
		$V_{OUT} = \pm2.5V$, $R_L = 150\Omega$	±5V	●	0.4			V/mV
		$V_{OUT} = \pm1V$, $R_L = 500\Omega$	±2.5V	●	1.7			V/mV
V_{OUT}	Output Swing	$R_L = 1k$, $V_{IN} = \pm40mV$	±15V	●	13.0			±V
		$R_L = 500\Omega$, $V_{IN} = \pm40mV$	±15V	●	11.5			±V
		$R_L = 500\Omega$, $V_{IN} = \pm40mV$	±5V	●	3.4			±V
		$R_L = 150\Omega$, $V_{IN} = \pm40mV$	±5V	●	2.6			±V
		$R_L = 500\Omega$, $V_{IN} = \pm40mV$	±2.5V	●	1.2			±V
I_{OUT}	Output Current	$V_{OUT} = \pm11.5V$	±15V	●	23.0			mA
		$V_{OUT} = \pm2.6V$	±5V	●	17.3			mA
I_{SC}	Short-Circuit Current	$V_{OUT} = 0V$, $V_{IN} = \pm3V$	±15V	●	23			mA
SR	Slew Rate	$A_V = -2$, (Note 4)	±15V	●	120			V/µs
			±5V	●	50			V/µs
GBW	Gain Bandwidth	f = 200kHz, $R_L = 2k$	±15V	●	7.0			MHz
			±5V	●	5.5			MHz
	Channel Separation	$V_{OUT} = \pm10V$, $R_L = 500\Omega$	±15V	●	98			dB
I_S	Supply Current	Each Amplifier	±15V	●			1.50	mA
		Each Amplifier	±5V	●			1.45	mA

Note 1: Absolute Maximum Ratings are those values beyond which the life of a device may be impaired.

Note 2: Differential inputs of ±10V are appropriate for transient operation only, such as during slewing. Large, sustained differential inputs will cause excessive power dissipation and may damage the part. See Input Considerations in the Applications Information section of this data sheet for more details.

Note 3: A heat sink may be required to keep the junction temperature below absolute maximum when the output is shorted indefinitely.

Note 4: Slew rate is measured between ±10V on the output with ±6V input for ±15V supplies and ±1V on the output with ±1.75V input for ±5V supplies.

Note 5: Full power bandwidth is calculated from the slew rate measurement: $FPBW = (SR)/2\pi V_P$.

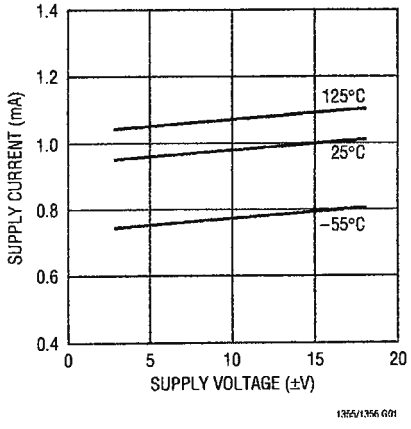
Note 6: This parameter is not 100% tested.

Note 7: The LT1355C/LT1356C are guaranteed functional over the operating temperature range of –40°C to 85°C.

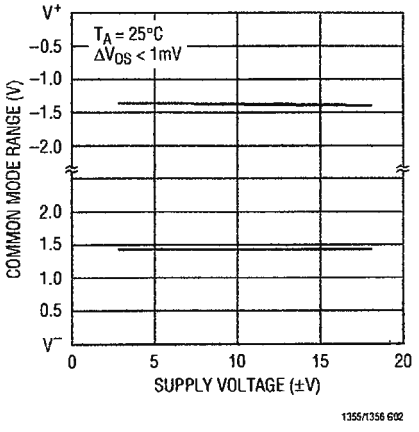
Note 8: The LT1355C/LT1356C are guaranteed to meet specified performance from 0°C to 70°C. The LT1355C/LT1356C are designed, characterized and expected to meet specified performance from –40°C to 85°C, but are not tested or QA sampled at these temperatures. For guaranteed I-grade parts, consult the factory.

TYPICAL PERFORMANCE CHARACTERISTICS

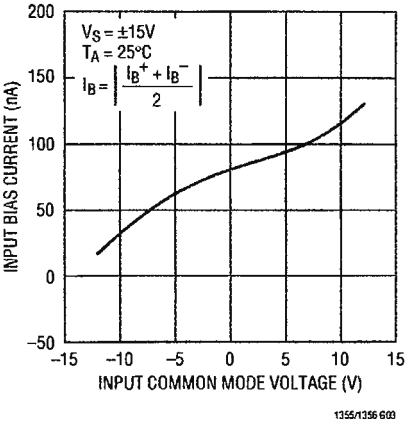
Supply Current vs Supply Voltage and Temperature



Input Common Mode Range vs Supply Voltage

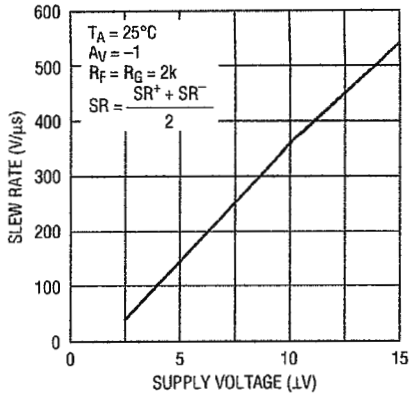


Input Bias Current vs Input Common Mode Voltage

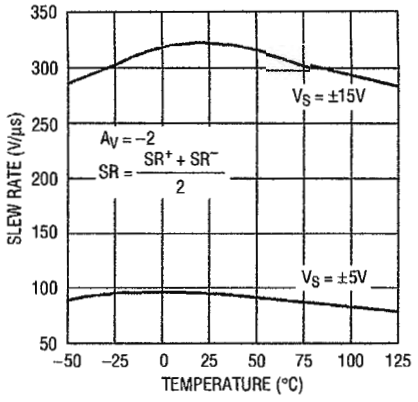


TYPICAL PERFORMANCE CHARACTERISTICS

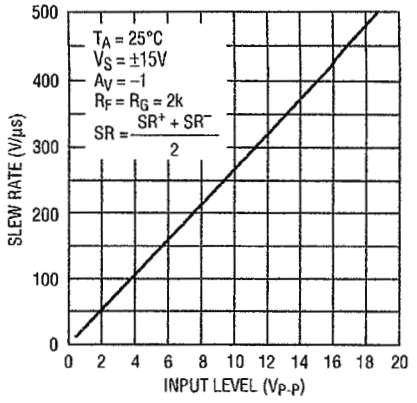
Slew Rate vs Supply Voltage



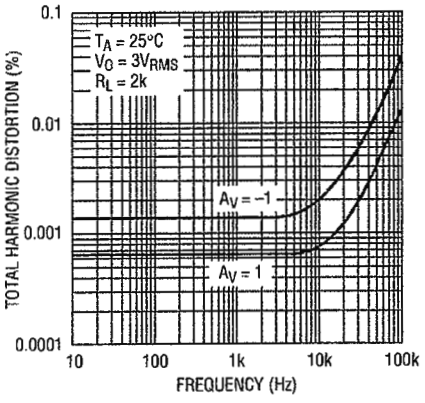
Slew Rate vs Temperature



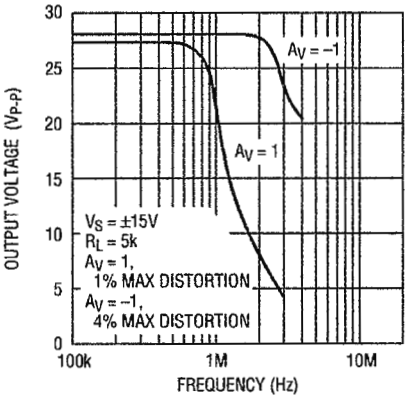
Slew Rate vs Input Level



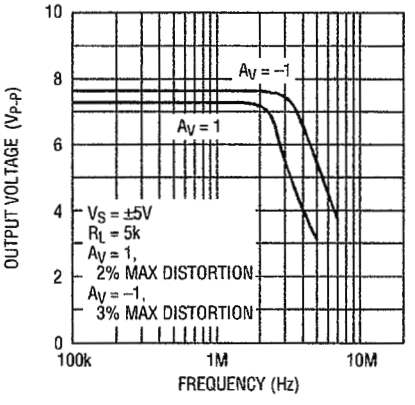
Total Harmonic Distortion vs Frequency



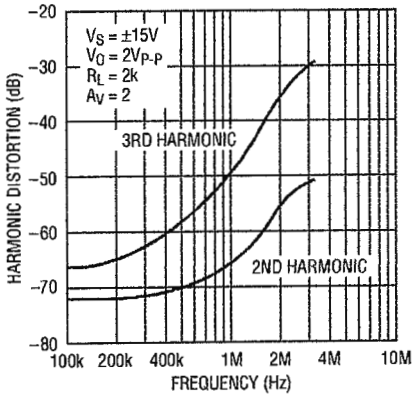
Undistorted Output Swing vs Frequency (±15V)



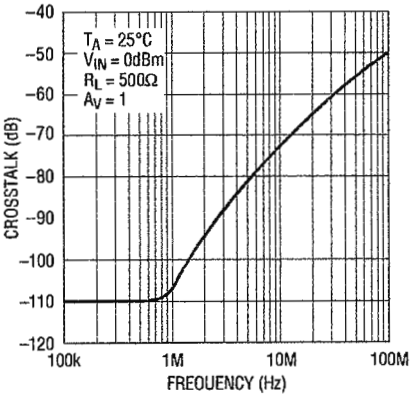
Undistorted Output Swing vs Frequency (±5V)



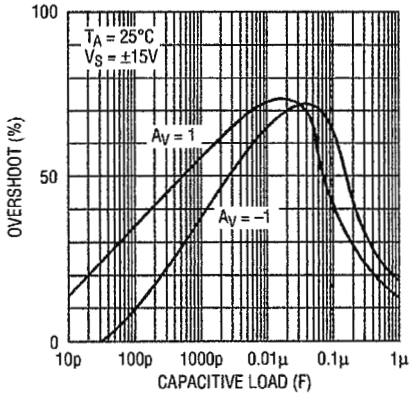
2nd and 3rd Harmonic Distortion vs Frequency



Crosstalk vs Frequency



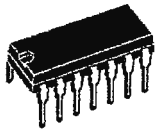
Capacitive Load Handling



TL074
TL074A - TL074B

LOW NOISE J-FET QUAD OPERATIONAL AMPLIFIERS

- WIDE COMMON-MODE (UP TO V_{CC}^+) AND DIFFERENTIAL VOLTAGE RANGE
- LOW INPUT BIAS AND OFFSET CURRENT
- LOW NOISE e_n = 15nV/√Hz (typ)
- OUTPUT SHORT-CIRCUIT PROTECTION
- HIGH INPUT IMPEDANCE J–FET INPUT STAGE
- LOW HARMONIC DISTORTION : 0.01% (typ)
- INTERNAL FREQUENCY COMPENSATION
- LATCH UP FREE OPERATION
- HIGH SLEW RATE : 13V/µs (typ)



N
DIP14
(Plastic Package)

D
SO14
(Plastic Micropackage)

DESCRIPTION

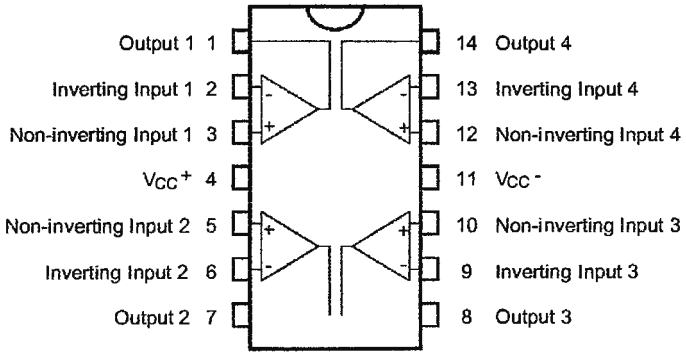
The TL074, TL074A and TL074B are high speed J–FET input quad operational amplifiers incorporating well matched, high voltage J–FET and bipolar transistors in a monolithic integrated circuit.

The devices feature high slew rates, low input bias and offset currents, and low offset voltage temperature coefficient.

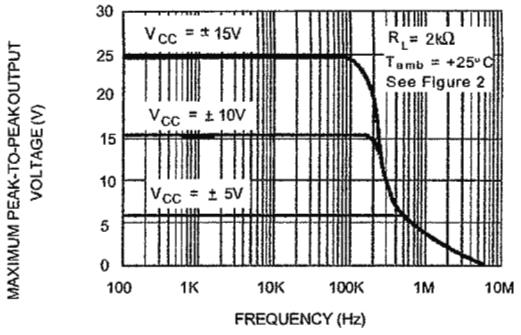
ORDER CODES

Part Number	Temperature Range	Package	
		N	D
TL074M/AM/BM	–55°C, +125°C	•	•
TL074I/AI/BI	–40°C, +105°C	•	•
TL074C/AC/BC	0°C, +70°C	•	•
Example : TL074IN			

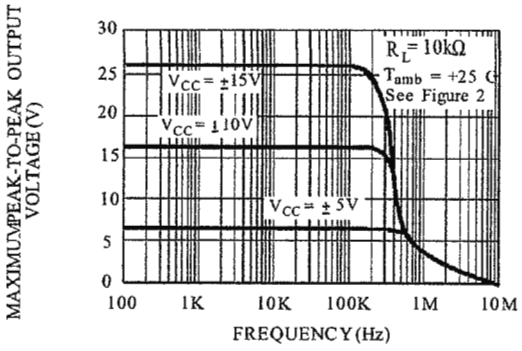
PIN CONNECTIONS (top view)



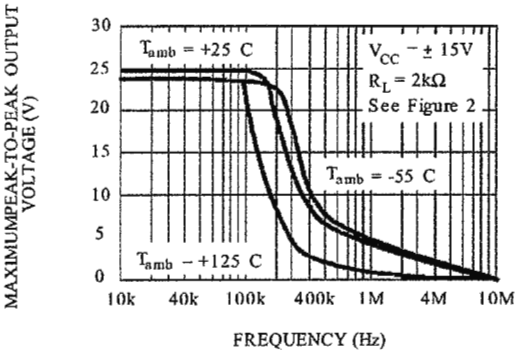
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS FREQUENCY



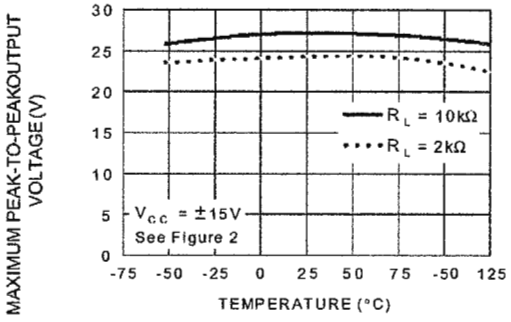
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS FREQUENCY



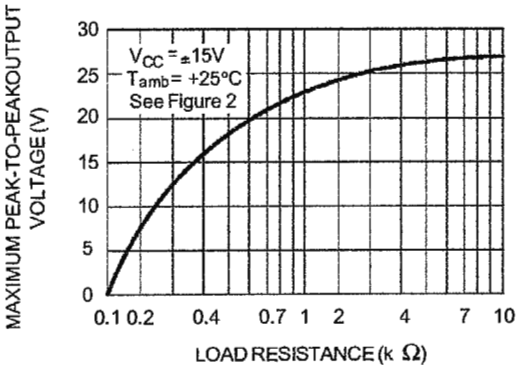
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS FREQUENCY



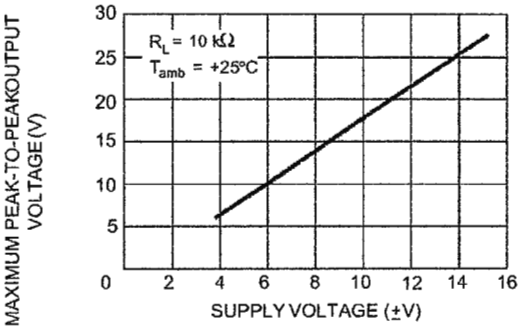
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS FREE AIR TEMP.



MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS LOAD RESISTANCE



MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE VERSUS SUPPLY VOLTAGE



TL072
TL072A - TL072B

LOW NOISE J-FET DUAL OPERATIONAL AMPLIFIERS

- WIDE COMMON-MODE (UP TO V_{CC}^+) AND DIFFERENTIAL VOLTAGE RANGE
- LOW INPUT BIAS AND OFFSET CURRENT
- LOW NOISE e_n = 15nV/√Hz (typ)
- OUTPUT SHORT-CIRCUIT PROTECTION
- HIGH INPUT IMPEDANCE J–FET INPUT STAGE
- LOW HARMONIC DISTORTION : 0.01% (typ)
- INTERNAL FREQUENCY COMPENSATION
- LATCH UP FREE OPERATION
- HIGH SLEW RATE : 16V/µs (typ)

N
DIP8
(Plastic Package)

D
SO8
(Plastic Micropackage)

DESCRIPTION

The TL072, TL072A and TL072B are high speed J–FET input dual operational amplifiers incorporating well matched, high voltage J–FET and bipolar transistors in a monolithic integrated circuit.

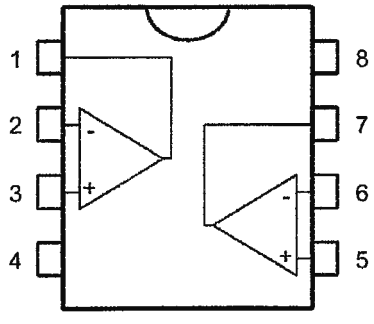
The devices feature high slew rates, low input bias and offset current, and low offset voltage temperature coefficient.

ORDER CODE

Part Number	Temperature Range	Package	
		N	D
TL072M/AM/BM	-55°C, +125°C	•	•
TL072I/AI/BI	-40°C, +105°C	•	•
TL072C/AC/BC	0°C, +70°C	•	•
Example : TL072CN			

N = Dual in Line Package (DIP)
D = Small Outline Package (SO) - also available in Tape & Reel (DT)

PIN CONNECTIONS (top view)



1 - Offset null 1
2 - Inverting input 1
3 - Non-inverting input 1
4 - V_{CC}^-
5 - Non-inverting input 2
6 -Inverting input 2
7 - Output 2
8 - V_{CC}^+

ELECTRICAL CHARACTERISTICS

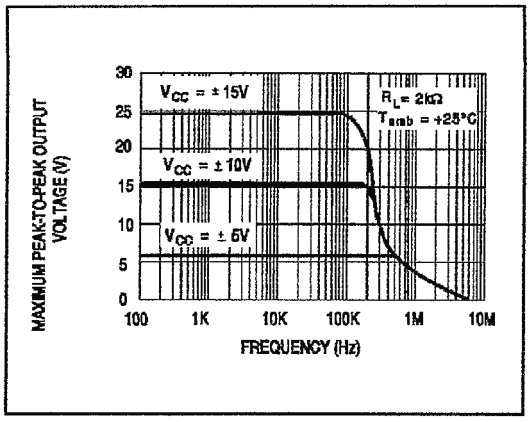
V_{CC} = ±15V, T_{amb} = +25°C (unless otherwise specified)

Symbol	Parameter		TL072I,M,AC,AI,AM, BC,BI,BM			TL072C			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{io}	Input Offset Voltage (R_s = 50Ω)								mV
	T_{amb} = +25°C	TL072		3	10		3	10	
		TL072A		3	6				
		TL072B		1	3				
	$T_{min} \le T_{amb} \le T_{max}$	TL072			13			13	
		TL072A			7				
		TL072B			5				
DV_{io}	Input Offset Voltage Drift			10			10		µV/°C
I_{io}	Input Offset Current - note 1)								
	T_{amb} = +25°C			5	100		5	100	pA
	$T_{min} \le T_{amb} \le T_{max}$				4			10	nA
I_{ib}	Input Bias Current -note 1								
	T_{amb} = +25°C			20	200		20	200	pA
	$T_{min} \le T_{amb} \le T_{max}$				20			20	nA
A_{vd}	Large Signal Voltage Gain (R_L = 2kΩ, V_o = ±10V)								V/mV
	T_{amb} = +25°C		50	200		25	200		
	$T_{min} \le T_{amb} \le T_{max}$		25			15			
SVR	Supply Voltage Rejection Ratio (R_S = 50Ω)								dB
	T_{amb} = +25°C		80	86		70	86		
	$T_{min} \le T_{amb} \le T_{max}$		80			70			
I_{CC}	Supply Current, no load								mA
	T_{amb} = +25°C			1.4	2.5		1.4	2.5	
	$T_{min} \le T_{amb} \le T_{max}$				2.5			2.5	
V_{icm}	Input Common Mode Voltage Range		±11	+15 -12		±11	+15 -12		V
CMR	Common Mode Rejection Ratio (R_S = 50Ω)								dB
	T_{amb} = +25°C		80	86		70	86		
	$T_{min} \le T_{amb} \le T_{max}$		80			70			
I_{os}	Output Short-circuit Current								mA
	T_{amb} = +25°C		10	40	60	10	40	60	
	$T_{min} \le T_{amb} \le T_{max}$		10		60	10		60	
$\pm V_{opp}$	Output Voltage Swing								V
	T_{amb} = +25°C	RL = 2kΩ	10	12		10	12		
		RL = 10kΩ	12	13.5		12	13.5		
	$T_{min} \le T_{amb} \le T_{max}$	RL = 2kΩ	10			10			
		RL = 10kΩ	12			12			
SR	Slew Rate (T_{amb} = +25°C)								V/µs
	V_{in} = 10V, R_L = 2kΩ, C_L = 100pF, unity gain		8	16		8	16		
t_r	Rise Time (T_{amb} = +25°C)								µs
	V_{in} = 20mV, R_L = 2kΩ, C_L = 100pF, unity gain			0.1			0.1		
K_{ov}	Overshoot (T_{amb} = +25°C)								%
	V_{in} = 20mV, R_L = 2kΩ, C_L = 100pF, unity gain			10			10		
GBP	Gain Bandwidth Product (T_{amb} = +25°C)								MHz
	V_{in} = 10mV, R_L = 2kΩ, C_L = 100pF, f= 100kHz		2.5	4		2.5	4		
R_i	Input Resistance			10^{12}			10^{12}		Ω

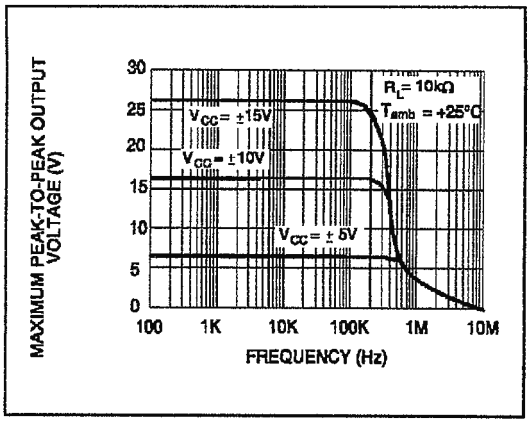
Symbol	Parameter	TL072I,M,AC,AI,AM, BC,BI,BM			TL072C			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
THD	Total Harmonic Distortion (T_{amb} = +25°C) f= 1kHz, R_L = 2kΩ,C_L = 100pF, A_v = 20dB, $V_o = 2V_{pp}$		0.01			0.01		%
e_n	Equivalent Input Noise Voltage R_S = 100Ω, f = 1KHz		15			15		$\frac{nV}{\sqrt{Hz}}$
Øm	Phase Margin		45			45		degrees
V_{o1}/V_{o2}	Channel separation A_v = 100		120			120		dB

1. The input bias currents are junction leakage currents which approximately double for every 10°C increase in the junction temperature.

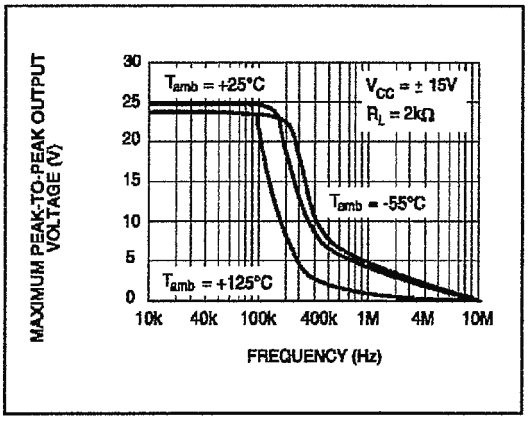
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus FREQUENCY



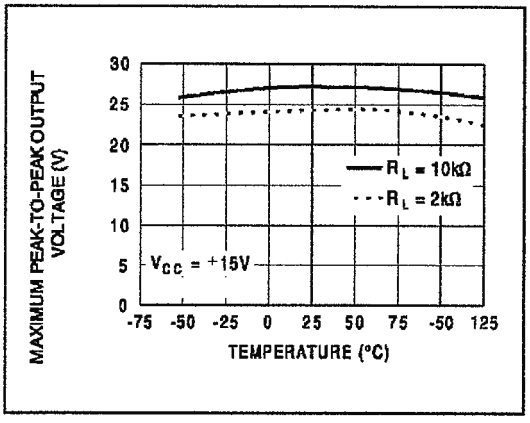
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus FREQUENCY



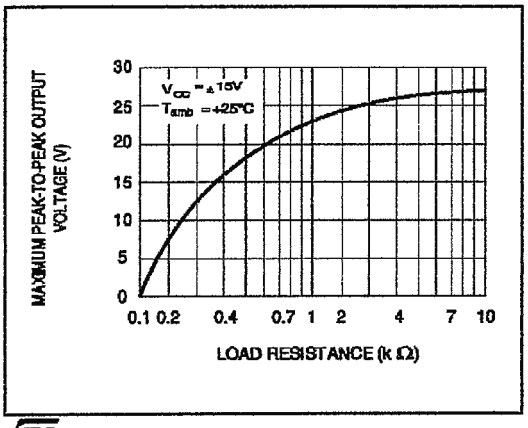
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus FREQUENCY



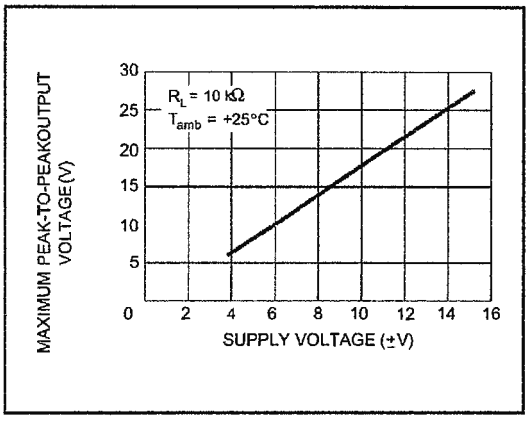
MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus FREE AIR TEMP.



MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus LOAD RESISTANCE



MAXIMUM PEAK-TO-PEAK OUTPUT VOLTAGE versus SUPPLY VOLTAGE



XR-2206

Monolithic
Function Generator

EXAR ...the analog plus company™

June 1997-3

FEATURES

- Low-Sine Wave Distortion, 0.5%, Typical
- Excellent Temperature Stability, 20ppm/°C, Typ.
- Wide Sweep Range, 2000:1, Typical
- Low-Supply Sensitivity, 0.01%V, Typ.
- Linear Amplitude Modulation
- TTL Compatible FSK Controls
- Wide Supply Range, 10V to 26V
- Adjustable Duty Cycle, 1% TO 99%

APPLICATIONS

- Waveform Generation
- Sweep Generation
- AM/FM Generation
- V/F Conversion
- FSK Generation
- Phase-Locked Loops (VCO)

GENERAL DESCRIPTION

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.

The circuit is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, AM, FM, or FSK generation. It has a typical drift specification of 20ppm/°C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage, while maintaining low distortion.

ORDERING INFORMATION

Part No.	Package	Operating Temperature Range
XR-2206M	16 Lead 300 Mil CDIP	-55°C to +125°C
XR-2206P	16 Lead 300 Mil PDIP	–40°C to +85°C
XR-2206CP	16 Lead 300 Mil PDIP	0°C to +70°C
XR-2206D	16 Lead 300 Mil JEDEC SOIC	0°C to +70°C

©1972
EXAR Corporation, 48720 Kato Road, Fremont, CA 94538 ◆ (510) 668-7000 ◆ (510) 668-7017

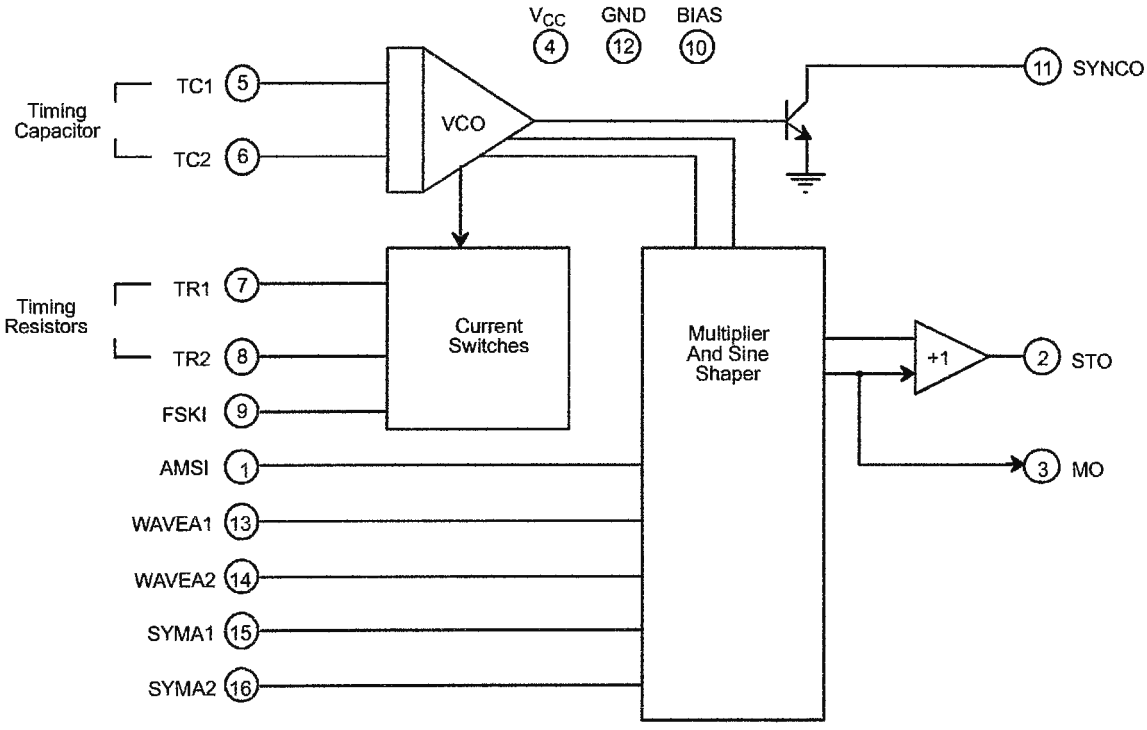
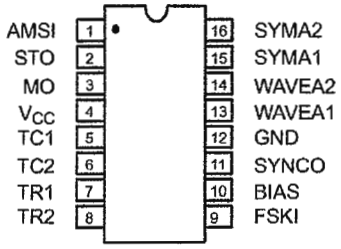
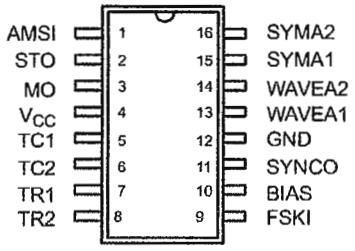


Figure 1. XR-2206 Block Diagram



16 Lead PDIP, CDIP (0.300")



16 Lead SOIC (Jedec, 0.300")

PIN DESCRIPTION

Pin #	Symbol	Type	Description
1	AMSI	I	**Amplitude Modulating Signal Input.**
2	STO	O	**Sine or Triangle Wave Output.**
3	MO	O	**Multiplier Output.**
4	V_{CC}		**Positive Power Supply.**
5	TC1	I	**Timing Capacitor Input.**
6	TC2	I	**Timing Capacitor Input.**
7	TR1	O	**Timing Resistor 1 Output.**
8	TR2	O	**Timing Resistor 2 Output.**
9	FSKI	I	**Frequency Shift Keying Input.**
10	BIAS	O	**Internal Voltage Reference.**
11	SYNCO	O	**Sync Output.** This output is a open collector and needs a pull up resistor to V_{CC}.
12	GND		**Ground pin.**
13	WAVEA1	I	**Wave Form Adjust Input 1.**
14	WAVEA2	I	**Wave Form Adjust Input 2.**
15	SYMA1	I	**Wave Symetry Adjust 1.**
16	SYMA2	I	**Wave Symetry Adjust 2.**

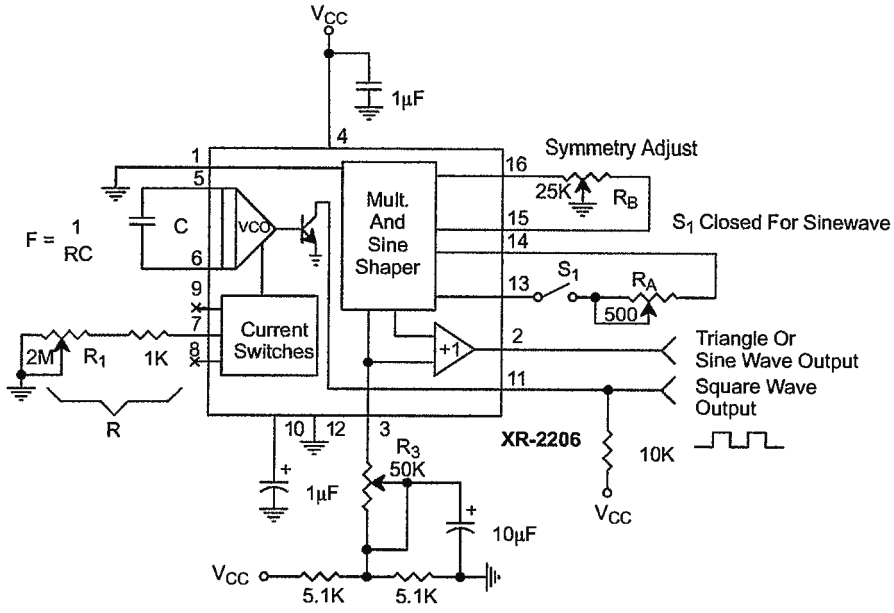


Figure 12. Circuit for Sine Wave Generation with Minimum Harmonic Distortion. (R_3 Determines Output Swing - See *Figure 3*)

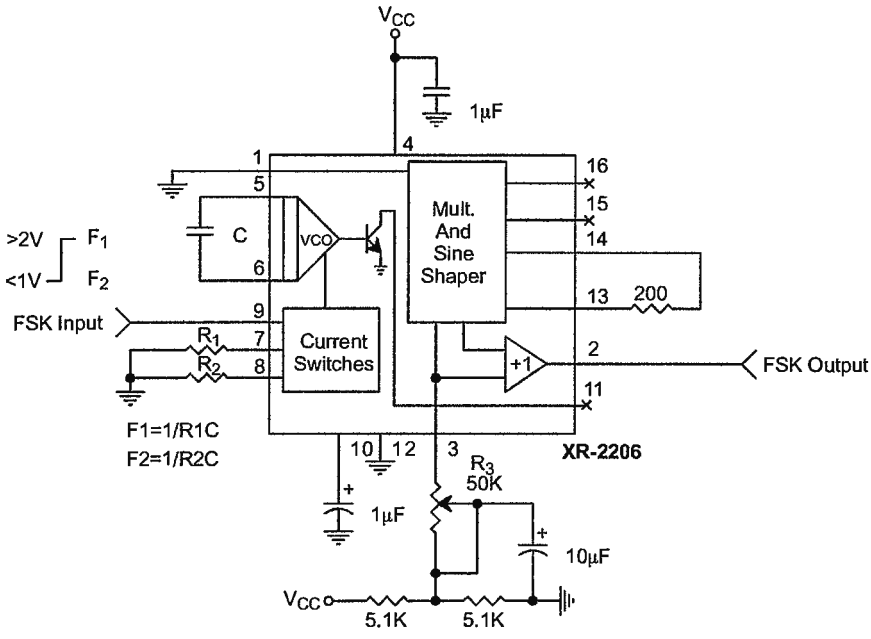


Figure 13. Sinusoidal FSK Generator

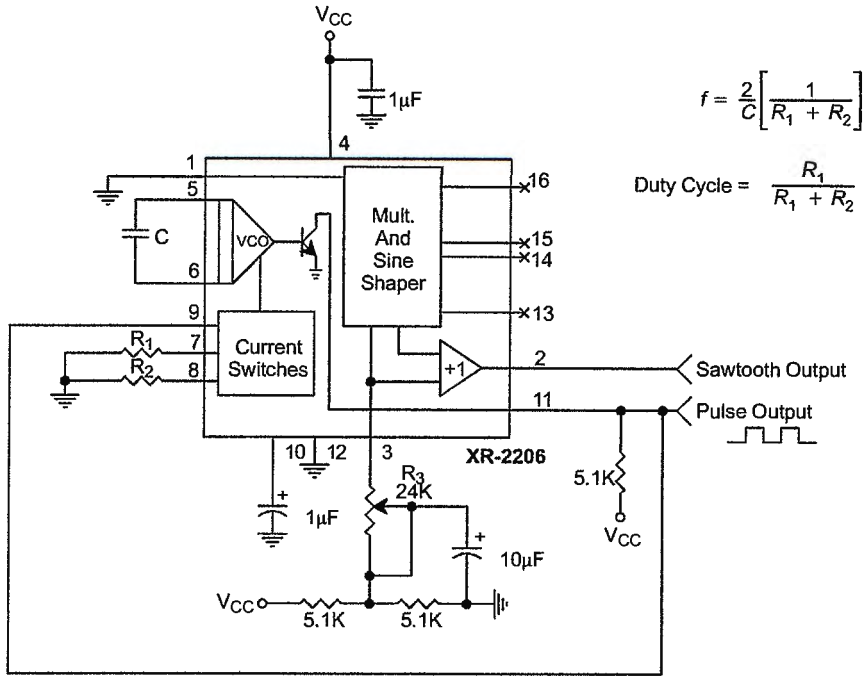


Figure 14. Circuit for Pulse and Ramp Generation.

Frequency-Shift Keying

The XR-2206 can be operated with two separate timing resistors, R_1 and R_2, connected to the timing Pin 7 and 8, respectively, as shown in *Figure 13.* Depending on the polarity of the logic signal at Pin 9, either one or the other of these timing resistors is activated. If Pin 9 is open-circuited or connected to a bias voltage $\geq$ 2V, only R_1 is activated. Similarly, if the voltage level at Pin 9 is $\leq$1V, only R_2 is activated. Thus, the output frequency can be keyed between two levels. f_1 and f_2, as:

$f_1 = 1/R_1C$ and $f_2 = 1/R_2C$

For split-supply operation, the keying voltage at Pin 9 is referenced to V^-.

Output DC Level Control

The dc level at the output (Pin 2) is approximately the same as the dc bias at Pin 3. In *Figure 11*, *Figure 12* and *Figure 13*, Pin 3 is biased midway between V+ and ground, to give an output dc level of $\approx V^+/2$.

APPLICATIONS INFORMATION

Sine Wave Generation

Without External Adjustment

Figure 11 shows the circuit connection for generating a sinusoidal output from the XR-2206. The potentiometer, R_1 at Pin 7, provides the desired frequency tuning. The maximum output swing is greater than $V^+/2$, and the typical distortion (THD) is < 2.5%. If lower sine wave distortion is desired, additional adjustments can be provided as described in the following section.

The circuit of *Figure 11* can be converted to split-supply operation, simply by replacing all ground connections with V^-. For split-supply operation, R_3 can be directly connected to ground.

With External Adjustment:

The harmonic content of sinusoidal output can be reduced to -0.5% by additional adjustments as shown in *Figure 12.* The potentiometer, R_A, adjusts the sine-shaping resistor, and R_B provides the fine adjustment for the waveform symmetry. The adjustment procedure is as follows:

1. Set R_B at midpoint and adjust R_A for minimum distortion.
2. With R_A set as above, adjust R_B to further reduce distortion.

Triangle Wave Generation

The circuits of *Figure 11* and *Figure 12* can be converted to triangle wave generation, by simply open-circuiting Pin 13 and 14 (i.e., S_1 open). Amplitude of the triangle is approximately twice the sine wave output.

FSK Generation

Figure 13 shows the circuit connection for sinusoidal FSK signal operation. Mark and space frequencies can be independently adjusted by the choice of timing resistors, R_1 and R_2; the output is phase-continuous during transitions. The keying signal is applied to Pin 9. The circuit can be converted to split-supply operation by simply replacing ground with V-.

Pulse and Ramp Generation

Figure 14 shows the circuit for pulse and ramp waveform generation. In this mode of operation, the FSK keying terminal (Pin 9) is shorted to the square-wave output (Pin 11), and the circuit automatically frequency-shift keys itself between two separate frequencies during the positive-going and negative-going output waveforms. The pulse width and duty cycle can be adjusted from 1% to 99% by the choice of R_1 and R_2. The values of R_1 and R_2 should be in the range of 1kΩ to 2MΩ.

PRINCIPLES OF OPERATION

Description of Controls

Frequency of Operation:

The frequency of oscillation, f_o, is determined by the external timing capacitor, C, across Pin 5 and 6, and by the timing resistor, R, connected to either Pin 7 or 8. The frequency is given as:

$$f_0 = \frac{1}{RC}\ Hz$$

and can be adjusted by varying either R or C. The recommended values of R, for a given frequency range, as shown in *Figure 5.* Temperature stability is optimum for 4kΩ < R < 200kΩ. Recommended values of C are from 1000pF to 100μF.

Frequency Sweep and Modulation:

Frequency of oscillation is proportional to the total timing current, I_T, drawn from Pin 7 or 8:

$$f = \frac{320 I_T(mA)}{C(\mu F)}\ Hz$$

Timing terminals (Pin 7 or 8) are low-impedance points, and are internally biased at +3V, with respect to Pin 12. Frequency varies linearly with IT, over a wide range of current values, from 1μA to 3mA. The frequency can be controlled by applying a control voltage, V_C, to the activated timing pin as shown in *Figure 10.* The frequency of oscillation is related to VC as:

$$f = \frac{1}{RC}\left(1 + \frac{R}{R_c}\left(1 - \frac{V_c}{3}\right)\right) Hz$$

where V_C is in volts. The voltage-to-frequency conversion gain, K, is given as:

$$K = \partial f / \partial V_c = -\frac{0.32}{R_c C}\ Hz/V$$

CAUTION: *For safety operation of the circuit, I_T should be limited to ≤ 3mA.*

TAN-005

High-Quality Function Generator System with the XR-2206

June 1997-3

INTRODUCTION

Waveform or function generators capable of producing AM/FM modulated sine wave outputs find a wide range of applications in electrical measurement and laboratory instrumentation. This application note describes the design, construction and performance of such a complete function generator system suitable for laboratory usage or hobbyist applications. The entire function generator is comprised of a single XR-2206 monolithic IC and a limited number of passive circuit components. It provides the engineer, student, or hobbyist with a highly versatile laboratory instrument for waveform generation, at a very small fraction of the cost of conventional function generators available today.

GENERAL DESCRIPTION

The basic circuit configuration and the external components necessary for the high-quality function generator system is shown in *Figure 1*. The circuit shown is designed to operate with either a 12V single power supply, or with 16V split supplies. For most applications, split-supply operation is preferred since it results in an output dc level which is nearly at ground potential.

The circuit configuration of *Figure 1* provides three basic waveforms: sine, triangle and square wave. There are four overlapping frequency ranges which give an overall frequency range of 1Hz to 100KHz. In each range, the frequency may be varied over a 100:1 tuning range.

The sine or triangle output can be varied from 0 to over 6V (peak to peak) from a 600Ω source at the output terminal.

A squarewave output is available at the sync output terminal for oscilloscope synchronizing or driving logic circuits.

TYPICAL PERFORMANCE CHARACTERISTICS

The performance characteristics listed are not guaranteed or warranted by EXAR Corporation. However, they represent the typical performance characteristics measured by EXAR's application engineers during the laboratory evaluation of the function generator system shown in *Figure 1*. The typical performance specifications listed below apply only when all of the recommended assembly instructions and adjustment procedures are followed:

(a) **Frequency Ranges:** The function generator system is designed to operate over four overlapping frequency ranges:

1Hz	to	100Hz
10Hz	to	1KHz
100Hz	to	10KHz
1KHz	to	100KHz

The range selection is made by switching in different timing capacitors.

(b) **Frequency Setting:** At any range setting, frequency can be varied over a 100:1 tuning range with a potentiometer (see R13 of *Figure 1*.)

(c) **Frequency Accuracy:** Frequency accuracy of the XR-2206 is set by the timing resistor R and the timing capacitor C, and is given as:

$$f = 1/RC$$

The above expression is accurate to within 15% at any range setting. The timing resistor R is the series combination of resistors R4 and R13 of *Figure 1*. The timing capacitor C is any one of the capacitors C3 through C6, shown in the figure.

(d) **Sine and Triangle Output:** The sine and triangle output amplitudes are variable from 0V to 6Vpp. The amplitude is set by an external potentiometer, R12 of *Figure 1*. At any given amplitude setting, the triangle output amplitude is approximately twice as high as the sinewave output. The internal impedance of the output is 600Ω.

©1995
EXAR Corporation, 48720 Kato Road, Fremont, CA 94538 ◆ (510) 668-7000 ◆ FAX (510) 668-7017

(e) **Sinewave Distortion:** The total harmonic distortion of sinewave is less than 1% from 10Hz to 10KHz and less than 3% over the entire frequency range. The selection of a waveform is made by the triangle/sine selector switch, S2.

(f) **Sync Output:** The sync output provides a 50% duty cycle pulse output with either full swing or upper half swing of the supply voltage depending on the choice of sync output terminals on the printed circuit board (see *Figure 1.*)

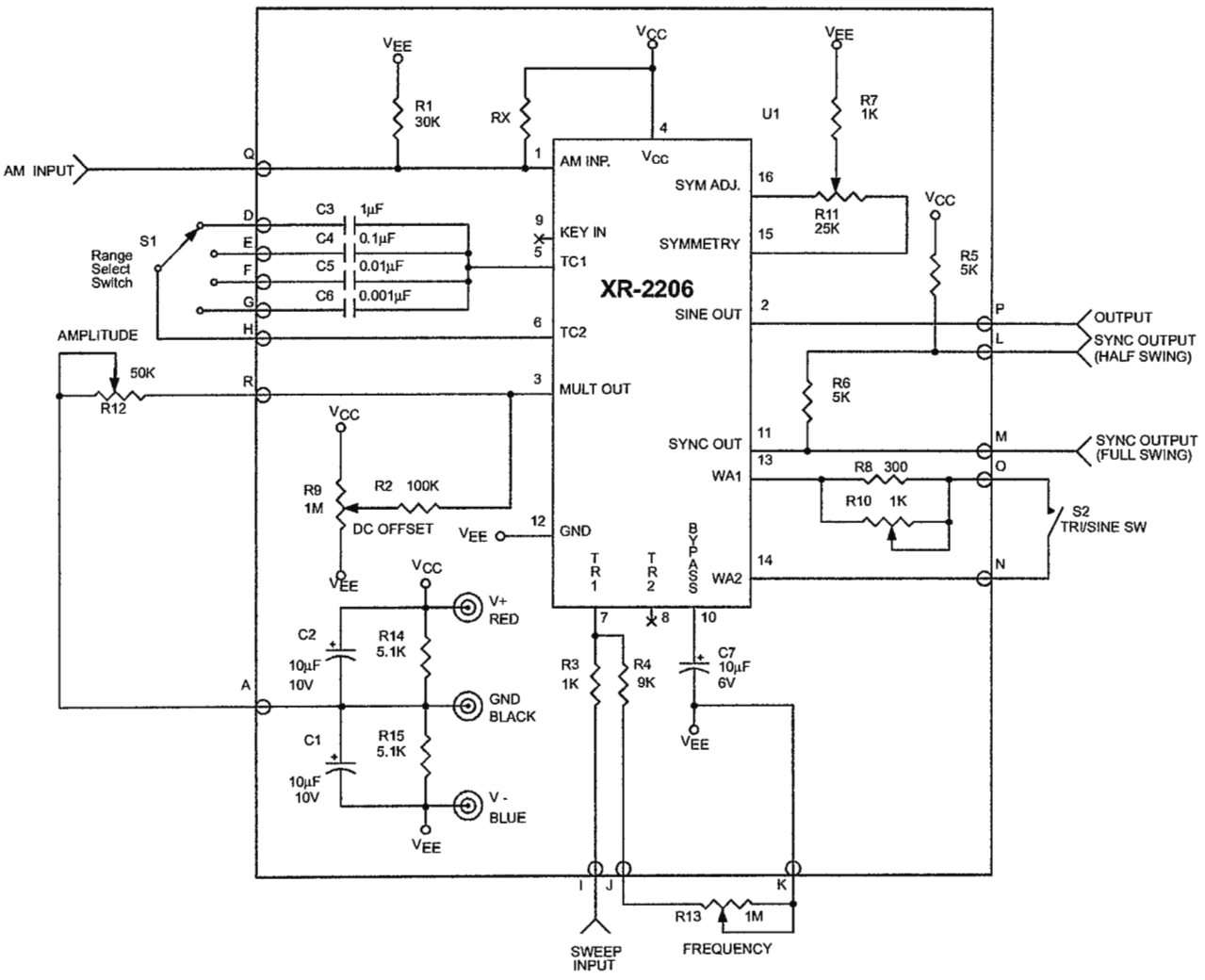


Figure 1. Circuit Connection Diagram for Function Generator (See (i) for Single Supply Operation)

g) **Frequency Modulation (External Sweep):** Frequency can be modulated or swept by applying an external control voltage to sweep terminal (Terminal I of *Figure 1.*) When not used, this terminal should be left open circuited. The open circuit voltage at this terminal is approximately 3V above the negative supply voltage and its impedance is approximately 1000Ω.

(h) **Amplitude Modulation:** The output amplitude varies linearly with modulation voltage applied to AM input (terminal Q of *Figure 1.*) The output amplitude reaches its minimum as the AM control voltage approaches the half of the total power supply voltage. The phase of the output signal reverses as the amplitude goes through its minimum value. The total dynamic range is approximately 55dB, with AM control voltage range of 4V referenced to the half of the total supply voltage. When not used, AM terminal should be left open-circuited.

(i) **Power Source: Split Supplies:** +/-6V, or single supply: +12V. Supply Current 15mA (see *Figure 2*). For single supply operation bias resistors, R14 and R15 should be added, the GND point left floating and V-tied to ground.

EXPLANATION OF CIRCUIT CONTROLS

Switches

Range Select Switch, S1: Selects the frequency range of operation for the function generator. The frequency is inversely proportional to the timing capacitor connected across Pins 5 and 6 of the XR-2206 circuit. Nominal capacitance values and frequency ranges corresponding to switch positions of S1 are as follows:

Position	Nominal Range	Timing Capacitance
1	1Hz to 100Hz	1μF
2	10Hz to 1KHz	0.1μF
3	100Hz to 10KHz	0.01μF
4	1Hz to 100kHz	0.001μF

If additional frequency ranges are needed, they can be added by introducing additional switch positions.

Triangle/Sine Waveform Switch, S2: Selects the triangle or sine output waveform.

Trimmers and Potentiometers

DC Offset Adjustment, R9: The potentiometer used for adjusting the dc offset level of the triangle or sine output waveform.

Sinewave Distortion Adjustment, R10: Adjusted to minimize the harmonic content of sinewave output.

Sinewave Symmetry Adjustment, R11: Adjusted to optimize the symmetry of the sinewave output.

Amplitude Control, R12: Sets the amplitude of the triangle or sinewave output.

Frequency Adjust, R13: Sets the oscillator frequency for any range setting of S1. Thus, R13 serves as a frequency dial on a conventional waveform generator and varies the frequency of the oscillator over an approximate 100 to 1 range.

Terminals

A. Negative Supply -6V
B. Ground
C. Positive Supply +6V
D. Range 1, timing capacitor terminal
E. Range 2, timing capacitor terminal
F. Range 3. timing capacitor terminal
G. Range 4, timing capacitor terminal
H. Timing capacitor common terminal
I. Sweep Input
J. Frequency adjust potentiometer terminal
K. Frequency adjust potentiometer negative supply terminal
L. Sync output (1/2 swing)
M. Sync output (full swing)
N. Triangle/sine waveform switch terminals
O. Triangle/sine waveform switch terminals
P. Triangle or sinewave output
Q. AM input
R. Amplitude control terminal

PARTS LIST

The following is a list of external circuit components necessary to provide the circuit interconnections shown in *Figure 1*.

Capacitors:

C1, C2, C7	Electrolytic, 10µF, 10V
C3	Mylar, 1µF, nonpolar, 10%
C4	Mylar, 0.1µF, 10%
C5	Mylar, 0.01µF, 10%
C6	Mylar, 1000pF, 10%

Resistors:

R1	30KΩ, 1/4 W, 10%
R2	100KΩ, 1/4 W, 10%
R3, R7	1KΩ, 1/4 W, 10%
R4	9KΩ, 1/4 W, 10%
R5, R6	5KΩ, 1/4 W. 10%
R8	300KΩ,1/4 W, 10%
RX	62KΩ, 1/4 W, 10% (RX can be eliminated for maximum output)

The following two resistors are used in single supply applications:

R14, R15	5.1KΩ, 1/4 W 10%

Potentiometers:

R9	Trim, 1MΩ, 1/4 W
R10	Trim, 1KΩ, 1/4 W
R11	Trim, 25KΩ, 1/4 W

The following additional items are recommended to convert the circuit of *Figure 1* to a complete laboratory instrument:

Potentiometers:

R12	Amplitude control, linear, 50KΩ
R13	Frequency control, audio taper, 1MΩ

Switches:

S1	Rotary switch, 1-pole, 4 positions
S2	Toggle or slide, SPST

Power Supply:

Dual supplies 16V or single +12V

Batteries or power supply unit

(See *Figure 2 (a)* and *Figure 2 (b)*.)

Miscellaneous:

Knobs, solder, wires, terminals, etc.

BOARD LAYOUT

Care should be given to the layout of the board, to prevent noise from the supplies from affecting the XR-2206 performance.

Any simple power supply having reasonable regulation may be used. *Figure 2* gives some recommended power supply configurations.

Precaution: Keep the lead lengths small for the range selector switch. This will reduce stray capacitance.

ADJUSTMENT PROCEDURE

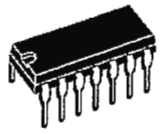
When assembly is completed and you are ready to put the function generator into operation, make sure that the polarity of power supply and the orientation of the IC unit are correct. Then apply the dc power to the unit.

To adjust for minimum distortion, connect the scope probe to the triangle/sine output. Close S2 and adjust the amplitude control to give non-clipping maximum swing. Then adjust R10 and R11 alternately for minimum distortion by observing the sinusoidal waveform. If a distortion meter is available, you may use it as a final check on the setting of sine-shaping trimmers. The minimum distortion obtained in this manner is typically less than 1% from 1Hz to 10KHz and less than 3% over the entire frequency range.

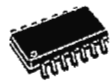
LM124
LM224 - LM324

LOW POWER QUAD OPERATIONAL AMPLIFIERS

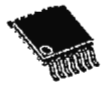
- WIDE GAIN BANDWIDTH : 1.3MHz
- INPUT COMMON-MODE VOLTAGE RANGE INCLUDES GROUND
- LARGE VOLTAGE GAIN : 100dB
- VERY LOW SUPPLY CURRENT/AMPLI : 375µA
- LOW INPUT BIAS CURRENT : 20nA
- LOW INPUT OFFSET VOLTAGE : 5mV max.
  (for more accurate applications, use the equivalent parts LM124A-LM224A-LM324A which feature 3mV max)
- LOW INPUT OFFSET CURRENT : 2nA
- WIDE POWER SUPPLY RANGE :
  SINGLE SUPPLY : +3V TO +30V
  DUAL SUPPLIES : ±1.5V TO ±15V



N
DIP14
(Plastic Package)



D
SO14
(Plastic Micropackage)



P
TSSOP14
(Thin Shrink Small Outline Package)

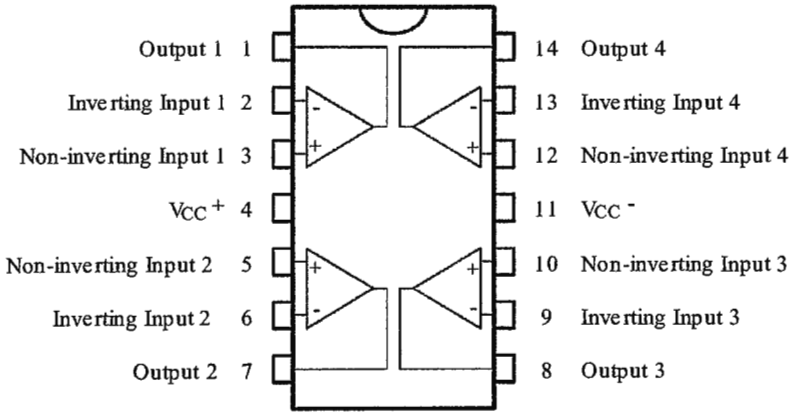
DESCRIPTION

These circuits consist of four independent, high gain, internally frequency compensated operational amplifiers. They operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

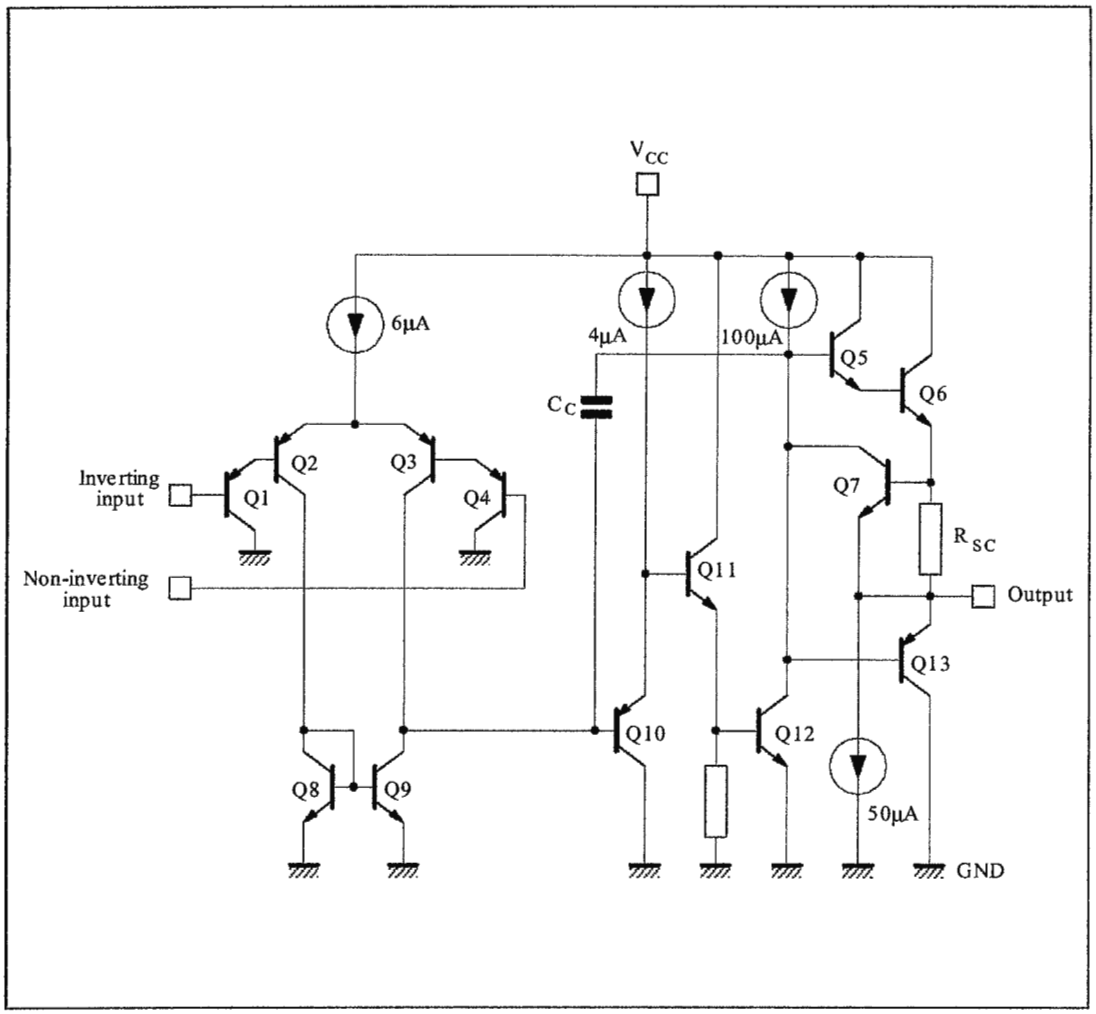
ORDER CODES

Part Number	Temperature Range	Package		
		N	D	P
LM124	-55°C, +125°C	•	•	•
LM224	-40°C, +105°C	•	•	•
LM324	0°C, +70°C	•	•	•
Example : LM224N				

PIN CONNECTIONS (top view)



SCHEMATIC DIAGRAM (1/4 LM124)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	LM124	LM224	LM324	Unit
V_{cc}	Supply Voltage	±16 or 32			V
V_i	Input Voltage	-0.3 to +32			V
V_{id}	Differential Input Voltage - (*)	+32	+32	+32	V
P_{tot}	Power Dissipation N Suffix D Suffix	500 -	500 400	500 400	mW mW
-	Output Short-circuit Duration - (note 1)	Infinite			
I_{in}	Input Current – (note 6)	50	50	50	mA
T_{oper}	Operating Free Air Temperature Range	-55 to +125	-40 to +105	0 to +70	°C
T_{stg}	Storage Temperature Range	-65 to +150	-65 to +150	-65 to +150	°C

ELECTRICAL CHARACTERISTICS

V_{CC}^{+} = +5V, V_{CC}^{-} = Ground, V_O = 1.4V, T_{amb} = +25°C (unless otherwise specified)

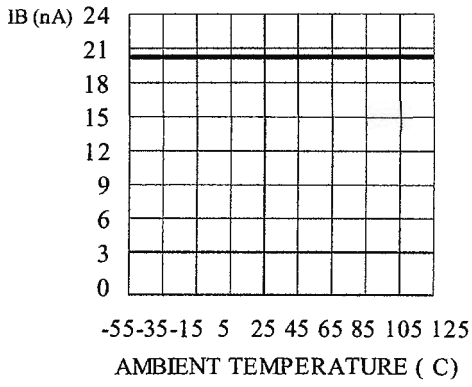
Symbol	Parameter	LM124 - LM224 - LM324			Unit
		Min.	Typ.	Max.	
V_{io}	Input Offset Voltage (note 3) T_{amb} = +25°C LM324 $T_{min.} \le T_{amb} \le T_{max.}$ LM324		2	 5 7 7 9	mV
I_{io}	Input Offset Current T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$		2	 30 100	nA
I_{ib}	Input Bias Current (note 2) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$		20	 150 300	nA
A_{vd}	Large Signal Voltage Gain (V_{CC}^{+} = +15V, R_L = 2kΩ, V_O = 1.4V to 11.4V) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$	 50 25	 100		V/mV
SVR	Supply Voltage Rejection Ratio ($R_S \le$ 10kΩ) (V_{CC}^{+} = 5V to 30V) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$	 65 65	 110		dB
I_{CC}	Supply Current, all Amp, no load T_{amb} = +25°C V_{CC} = +5V V_{CC} = +30V $T_{min.} \le T_{amb} \le T_{max.}$ V_{CC} = +5V V_{CC} = +30V		 0.7 1.5 0.8 1.5	 1.2 3 1.2 3	mA
V_{icm}	Input Common Mode Voltage Range (V_{CC} = +30V) - (note 4) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$	 0 0		 V_{CC} -1.5 V_{CC} -2	V
CMR	Common-mode Rejection Ratio ($R_S \le$ 10kΩ) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max}$	 70 60	 80		dB
I_{source}	Output Current Source (V_{id} = +1V) V_{CC} = +15V, V_o = +2V	 20	 40	 70	mA
I_{sink}	Output Sink Current (V_{id} = -1V) V_{CC} = +15V, V_o = +2V V_{CC} = +15V, V_o = +0.2V	 10 12	 20 50		 mA µA

ELECTRICAL CHARACTERISTICS (continued)

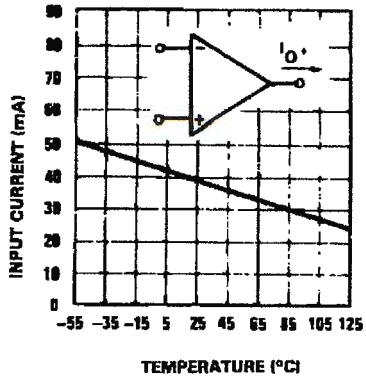
Symbol	Parameter	LM124 - LM224 - LM324			Unit
		Min.	Typ.	Max.	
V_{OH}	High Level Output Voltage (V_{CC} = +30V) T_{amb} = +25°C R_L = 2kΩ $T_{min.} \le T_{amb} \le T_{max.}$ T_{amb} = +25°C R_L = 10kΩ $T_{min.} \le T_{amb} \le T_{max.}$ (V_{CC} = +5V, R_L = 2kΩ) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$	 26 26 27 27 3.5 3	 27 28		V
V_{OL}	Low Level Output Voltage (R_L = 10kΩ) T_{amb} = +25°C $T_{min.} \le T_{amb} \le T_{max.}$		5	20 20	mV
SR	Slew Rate V_{CC} = 15V, V_i = 0.5 to 3V, R_L = 2kΩ, C_L = 100pF, unity gain)		0.4		V/µs
GBP	Gain Bandwidth Product V_{CC} = 30V, f = 100kHz, V_{in} = 10mV R_L = 2kΩ, C_L = 100pF		1.3		MHz
THD	Total Harmonic Distortion f = 1kHz, A_V = 20dB, R_L = 2kΩ, V_O = $2V_{pp}$ C_L = 100pF, V_{CC} = 30V		0.015		%
e_n	Equivalent Input Noise Voltage f = 1kHz, R_S = 100Ω, V_{CC} = 30V		40		$\frac{nV}{\sqrt{Hz}}$
DV_{io}	Input Offset Voltage Drift		7	30	µV/°C
DI_{IO}	Input Offset Current Drift		10	200	pA/°C
V_{o1}/V_{o2}	Channel Separation (note 5) 1kHz ≤ f ≤ 20kHz		120		dB

Notes :
1. Short-circuits from the output to V_{CC} can cause excessive heating if V_{CC} > 15V. The maximum output current is approximately 40mA independent of the magnitude of V_{CC}. Destructive dissipation can result from simultaneous short-circuit on all amplifiers.
2. The direction of the input current is out of the IC. This current is essentially constant, independent of the state of the output so no loading change exists on the input lines.
3. V_o = 1.4V, R_s = 0Ω, 5V < V_{CC}^+ < 30V, 0 < V_{ic} < V_{CC}^+ - 1.5V
4. The input common-mode voltage of either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is V_{CC}^+ - 1.5V, but either or both inputs can go to +32V without damage.
5. Due to the proximity of external components insure that coupling is not originating via stray capacitance between these external parts. This typically can be detected as this type of capacitance increases at higher frequences.
6. This input current only exists when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistor becoming forward biased and thereby acting as input diodes clamps. In addition to this diode action, there is also NPN parasitic action on the IC chip. this transistor action can cause the output voltages of the Op-amps to go to the V_{CC} voltage level (or to ground for a large overdrive) for the time duration than an input is driven negative.
This is not destructive and normal output will set up again for input voltage higher than -0.3V.

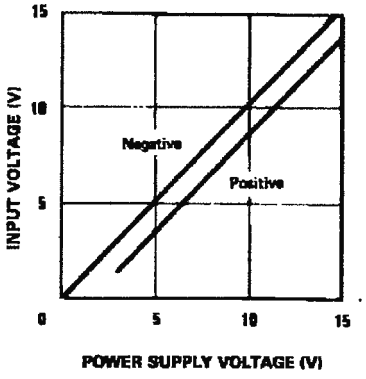
INPUT BIAS CURRENT
versus AMBIENT TEMPERATURE



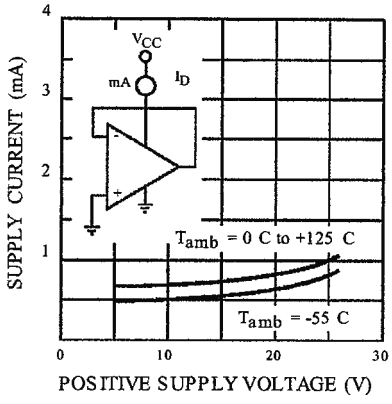
CURRENT LIMITING (Note 6)



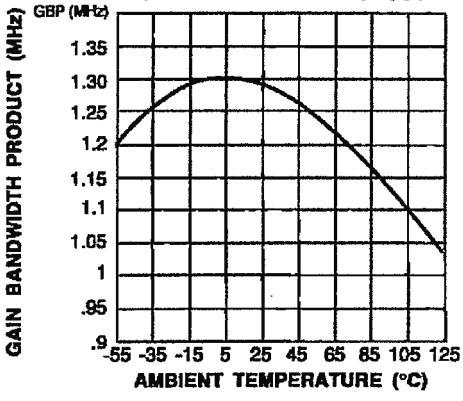
INPUT VOLTAGE RANGE



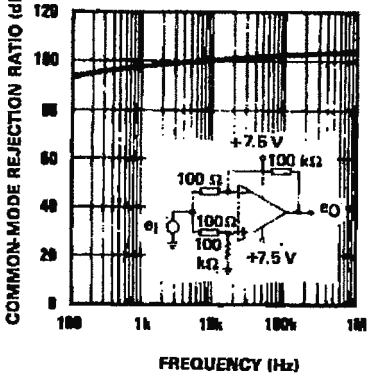
SUPPLY CURRENT



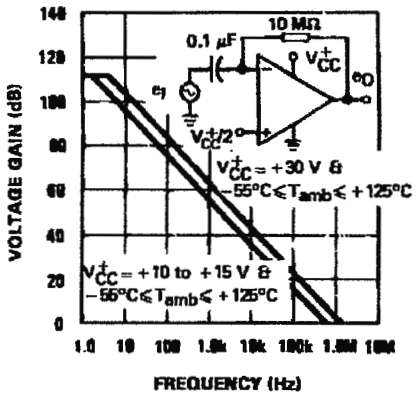
GAIN BANDWIDTH PRODUCT



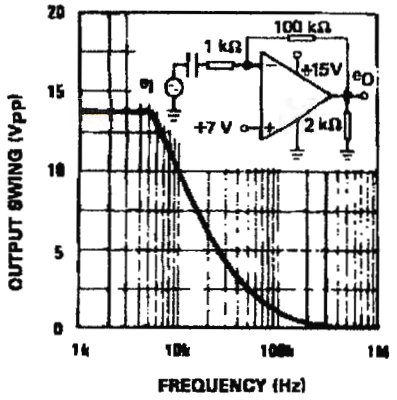
COMMON-MODE REJECTION RATIO



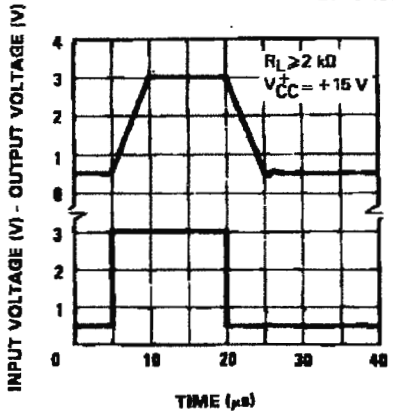
OPEN LOOP FREQUENCY RESPONSE



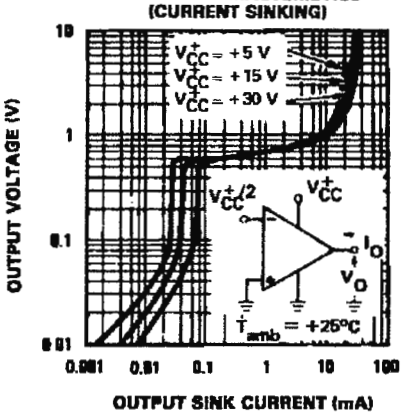
LARGE SIGNAL FREQUENCY RESPONSE



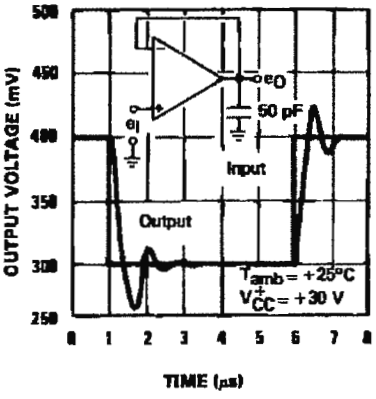
VOLTAGE FOLLOWER PULSE RESPONSE



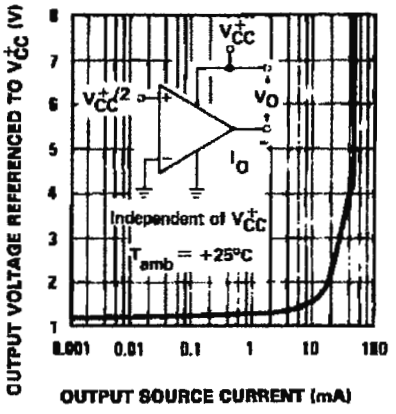
OUTPUT CHARACTERISTICS (CURRENT SINKING)



VOLTAGE FOLLOWER PULSE RESPONSE (SMALL SIGNAL)



OUTPUT CHARACTERISTICS (CURRENT SOURCING)



National Semiconductor

January 1995

DAC0808/DAC0807/DAC0806 8-Bit D/A Converters

General Description

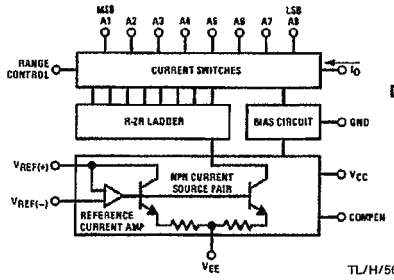
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 I_{REF}/ 256. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 μA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

Features

- Relative accuracy: ±0.19% error maximum (DAC0808)
- Full scale current match: ±1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/μs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

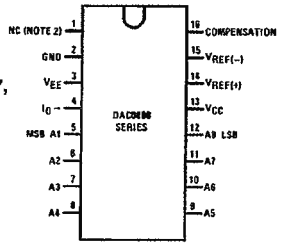
Block and Connection Diagrams



TL/H/5687–1

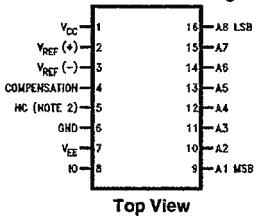
Order Number DAC0808, DAC0807, or DAC0806 See NS Package Number J16A, M16A or N16A

Dual-In-Line Package



TL/H/5687–2

Small-Outline Package



Top View

TL/H/5687–13

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS				
		J PACKAGE (J16A)*		N PACKAGE (N16A)*		SO PACKAGE (M16A)
				DAC0808LCN	MC1408P8	DAC0808LCM
7-bit	0°C ≤ T_A ≤ +75°C	DAC0807LCJ	MC1408L7	DAC0807LCN	MC1408P7	DAC0807LCM
6-bit	0°C ≤ T_A ≤ +75°C	DAC0806LCJ	MC1408L6	DAC0806LCN	MC1408P6	DAC0806LCM

*Note. Devices may be ordered by using either order number.

©1995 National Semiconductor Corporation TL/H/5687

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage	
V_{CC}	$+18\ V_{DC}$
V_{EE}	$-18\ V_{DC}$
Digital Input Voltage, V5–V12	$-10\ V_{DC}$ to $+18\ V_{DC}$
Applied Output Voltage, V_O	$-11\ V_{DC}$ to $+18\ V_{DC}$
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V14, V15	V_{CC}, V_{EE}
Power Dissipation (Note 3)	1000 mW
ESD Susceptibility (Note 4)	TBD
Storage Temperature Range	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (Plastic)	260°C
Dual-In-Line Package (Ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C

Operating Ratings

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
DAC0808LC Series	$0 \le T_A \le +75°C$

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, V_{REF}/R14 = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E_r	Relative Accuracy (Error Relative to Full Scale I_O)	*(Figure 4)*				%
	DAC0808LC (LM1408-8)				**±0.19**	%
	DAC0807LC (LM1408-7), (Note 5)				±0.39	%
	DAC0806LC (LM1408-6), (Note 5)				±0.78	%
	Settling Time to Within ½ LSB (Includes t_{PLH})	T_A = 25°C (Note 6), *(Figure 5)*		150		ns
t_{PLH}, t_{PHL}	Propagation Delay Time	T_A = 25°C, *(Figure 5)*		30	100	ns
TCI_O	Output Full Scale Current Drift			±20		ppm/°C
MSB	Digital Input Logic Levels	*(Figure 3)*				
V_{IH}	High Level, Logic "1"		2			V_{DC}
V_{IL}	Low Level, Logic "0"				0.8	V_{DC}
MSB	Digital Input Current	*(Figure 3)*				
	High Level	V_{IH} = 5V		0	0.040	mA
	Low Level	V_{IL} = 0.8V		−0.003	−0.8	mA
I_{15}	Reference Input Bias Current	*(Figure 3)*		−1	−3	μA
	Output Current Range	*(Figure 3)*				
		V_{EE} = −5V	0	2.0	2.1	mA
		V_{EE} = −15V, T_A = 25°C	0	2.0	4.2	mA
I_O	Output Current	V_{REF} = 2.000V, R14 = 1000Ω, *(Figure 3)*	1.9	1.99	2.1	mA
	Output Current, All Bits Low	*(Figure 3)*		0	4	μA
	Output Voltage Compliance (Note 2)	$E_r \le 0.19\%$, T_A = 25°C				
	V_{EE} = −5V, I_{REF} = 1 mA				−0.55, +0.4	V_{DC}
	V_{EE} Below −10V				−5.0, +0.4	V_{DC}

Electrical Characteristics (Continued)

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, V_{REF}/R14 = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
SRI_{REF}	Reference Current Slew Rate	*(Figure 6)*	4	8		mA/μs
	Output Current Power Supply Sensitivity	−5V ≤ V_{EE} ≤ −16.5V		0.05	2.7	μA/V
	Power Supply Current (All Bits Low)	*(Figure 3)*				
I_{CC}				2.3	22	mA
I_{EE}				−4.3	−13	mA
	Power Supply Voltage Range	T_A = 25°C, *(Figure 3)*				
V_{CC}			4.5	5.0	5.5	V_{DC}
V_{EE}			−4.5	−15	−16.5	V_{DC}
	Power Dissipation					
	All Bits Low	V_{CC} = 5V, V_{EE} = −5V		33	170	mW
		V_{CC} = 5V, V_{EE} = −15V		106	305	mW
	All Bits High	V_{CC} = 15V, V_{EE} = −5V		90		mW
		V_{CC} = 15V, V_{EE} = −15V		160		mW

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: Range control is not required.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX}, θ_{JA}, and the ambient temperature, T_A. The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maximum Ratings, whichever is lower. For this device, T_{JMAX} = 125°C, and the typical junction-to-ambient thermal resistance of the dual-in-line J package when the board mounted is 100°C/W. For the dual-in-line N package, this number increases to 175°C/W and for the small outline M package this number is 100°C/W.

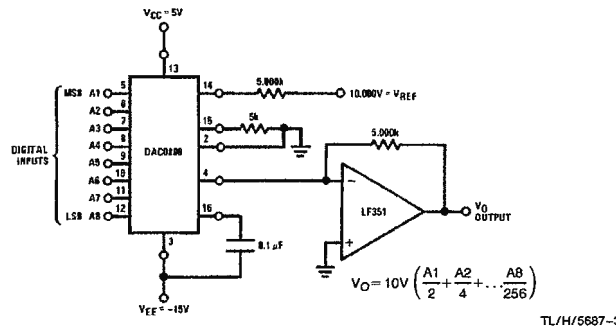
Note 4: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 5: All current switches are tested to guarantee at least 50% of rated current.

Note 6: All bits switched.

Note 7: Pin-out numbers for the DAL080X represent the dual-in-line package. The small outline package pinout differs from the dual-in-line package.

Typical Application



TL/H/5687–3

FIGURE 1. + 10V Output Digital to Analog Converter (Note 7)

1N4728A - 1N4764A Series

1 Watt DO-41 Hermetically Sealed Glass Zener Voltage Regulator Diodes

This is a complete series of 1 Watt Zener diode with limits and excellent operating characteristics that reflect the superior capabilities of silicon–oxide passivated junctions. All this in an axial–lead hermetically sealed glass package that offers protection in all common environmental conditions.

Specification Features:

- Zener Voltage Range — 3.3 V to 91 V
- ESD Rating of Class 3 (>16 KV) per Human Body Model
- DO–41 (DO–204AL) Package
- Double Slug Type Construction
- Metallurgical Bonded Construction
- Oxide Passivated Die

Mechanical Characteristics:
CASE: Double slug type, hermetically sealed glass
FINISH: All external surfaces are corrosion resistant and leads are readily solderable
MAXIMUM LEAD TEMPERATURE FOR SOLDERING PURPOSES: 230°C, 1/16″ from the case for 10 seconds
POLARITY: Cathode indicated by polarity band
MOUNTING POSITION: Any

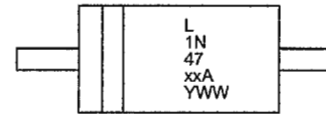
MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Max. Steady State Power Dissipation @ $T_L \leq 50°C$, Lead Length = 3/8″	P_D	1.0	Watt
Derated above 50°C		6.67	mW/°C
Operating and Storage Temperature Range	T_J, T_{stg}	– 65 to +200	°C

ON Semiconductor™

http://onsemi.com

AXIAL LEAD
CASE 59
GLASS



L = Assembly Location
1N47xxA = Device Code
Y = Year
WW = Work Week

ORDERING INFORMATION (1.)(NO TAG)

Device	Package	Shipping
1N47xxA	Axial Lead	2000 Units/Box
1N47xxARL	Axial Lead	6000/Tape & Reel
1N47xxARL2	Axial Lead	6000/Tape & Reel
1N47xxATA	Axial Lead	4000/Ammo Pack
1N47xxATA2	Axial Lead	4000/Ammo Pack

NOTES:
1. The "2" suffix refers to 26 mm tape spacing.

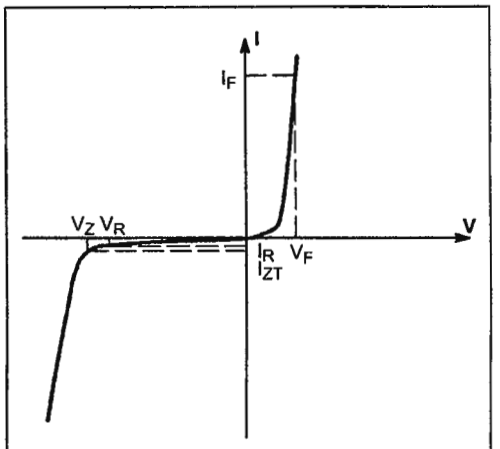
Devices listed in ***bold, italic*** are ON Semiconductor **Preferred** devices. **Preferred** devices are recommended choices for future use and best overall value.

© Semiconductor Components Industries, LLC, 2000

November, 2000 – Rev. 1
Publication Order Number:
1N4728A/D

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted, V_F = 1.2 V Max., I_F = 200 mA for all types)

Symbol	Parameter
V_Z	Reverse Zener Voltage @ I_{ZT}
I_{ZT}	Reverse Current
Z_{ZT}	Maximum Zener Impedance @ I_{ZT}
I_{ZK}	Reverse Current
Z_{ZK}	Maximum Zener Impedance @ I_{ZK}
I_R	Reverse Leakage Current @ V_R
V_R	Breakdown Voltage
I_F	Forward Current
V_F	Forward Voltage @ I_F
I_r	Surge Current @ T_A = 25°C



Zener Voltage Regulator

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted, V_F = 1.2 V Max, I_F = 200 mA for all types)

JEDEC Device (2.)	Zener Voltage (3.)(4.)				Zener Impedance (5.)			Leakage Current		I_r (6.)
	V_Z (Volts)			@ I_{ZT}	Z_{ZT} @ I_{ZT}	Z_{ZK} @ I_{ZK}		I_R @ V_R		
	Min	Nom	Max	(mA)	(Ω)	(Ω)	(mA)	(μA Max)	(Volts)	(mA)
1N4728A	3.14	3.3	3.47	76	10	400	1	100	1	1380
1N4729A	3.42	3.6	3.78	69	10	400	1	100	1	1260
1N4730A	3.71	3.9	4.10	64	9	400	1	50	1	1190
1N4731A	4.09	4.3	4.52	58	9	400	1	10	1	1070
1N4732A	4.47	4.7	4.94	53	8	500	1	10	1	970
1N4733A	***4.85***	***5.1***	***5.36***	***49***	***7***	***550***	***1***	***10***	***1***	***890***
1N4734A	***5.32***	***5.6***	***5.88***	***45***	***5***	***600***	***1***	***10***	***2***	***810***
1N4735A	***5.89***	***6.2***	***6.51***	***41***	***2***	***700***	***1***	***10***	***3***	***730***
1N4736A	***6.46***	***6.8***	***7.14***	***37***	***3.5***	***700***	***1***	***10***	***4***	***660***
1N4737A	7.13	7.5	7.88	34	4	700	0.5	10	5	605
1N4738A	***7.79***	***8.2***	***8.61***	***31***	***4.5***	***700***	***0.5***	***10***	***6***	***550***
1N4739A	8.65	9.1	9.56	28	5	700	0.5	10	7	500
1N4740A	***9.50***	***10***	***10.50***	***25***	***7***	***700***	***0.25***	***10***	***7.6***	***454***
1N4741A	***10.45***	***11***	***11.55***	***23***	***8***	***700***	***0.25***	***5***	***8.4***	***414***
1N4742A	***11.40***	***12***	***12.60***	***21***	***9***	***700***	***0.25***	***5***	***9.1***	***380***

TOLERANCE AND TYPE NUMBER DESIGNATION

2. The JEDEC type numbers listed have a standard tolerance on the nominal zener voltage of ±5%.

SPECIALS AVAILABLE INCLUDE:

3. Nominal zener voltages between the voltages shown and tighter voltage tolerances. For detailed information on price, availability, and delivery, contact your nearest ON Semiconductor representative.

ZENER VOLTAGE (V_Z) MEASUREMENT

4. ON Semiconductor guarantees the zener voltage when measured at 90 seconds while maintaining the lead temperature (T_L) at 30°C ± 1°C, 3/8″ from the diode body.

ZENER IMPEDANCE (Z_Z) DERIVATION

5. The zener impedance is derived from the 60 cycle ac voltage, which results when an ac current having an rms value equal to 10% of the dc zener current (I_{ZT} or I_{ZK}) is superimposed on I_{ZT} or I_{ZK}.

SURGE CURRENT (I_R) NON-REPETITIVE

6. The rating listed in the electrical characteristics table is maximum peak, non-repetitive, reverse surge current of 1/2 square wave or equivalent sine wave pulse of 1/120 second duration superimposed on the test current, I_{ZT}, per JEDEC registration; however, actual device capability is as described in Figure 5 of the General Data — DO-41 Glass.

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted, V_F = 1.2 V Max, I_F = 200 mA for all types) (continued)

JEDEC Device (2.)	Zener Voltage (3.)(4.)				Zener Impedance (5.)			Leakage Current		I_r (6.)
	V_Z (Volts)			@ I_{ZT}	Z_{ZT} @ I_{ZT}	Z_{ZK} @ I_{ZK}		I_R @ V_R		
	Min	Nom	Max	(mA)	(Ω)	(Ω)	(mA)	(μA Max)	(Volts)	(mA)
1N4743A	12.4	13	13.7	19	10	700	0.25	5	9.9	344
1N4744A	***14.3***	***15***	***15.8***	***17***	***14***	***700***	***0.25***	***5***	***11.4***	***304***
1N4745A	***15.2***	***16***	***16.8***	***15.5***	***16***	***700***	***0.25***	***5***	***12.2***	***285***
1N4746A	***17.1***	***18***	***18.9***	***14***	***20***	***750***	***0.25***	***5***	***13.7***	***250***
1N4747A	***19.0***	***20***	***21.0***	***12.5***	***22***	***750***	***0.25***	***5***	***15.2***	***225***
1N4748A	***20.9***	***22***	***23.1***	***11.5***	***23***	***750***	***0.25***	***5***	***16.7***	***205***
1N4749A	***22.8***	***24***	***25.2***	***10.5***	***25***	***750***	***0.25***	***5***	***18.2***	***190***
1N4750A	***25.7***	***27***	***28.4***	***9.5***	***35***	***750***	***0.25***	***5***	***20.6***	***170***
1N4751A	***28.5***	***30***	***31.5***	***8.5***	***40***	***1000***	***0.25***	***5***	***22.8***	***150***
1N4752A	***31.4***	***33***	***34.7***	***7.5***	***45***	***1000***	***0.25***	***5***	***25.1***	***135***
1N4753A	34.2	36	37.8	7	50	1000	0.25	5	27.4	125
1N4754A	37.1	39	41.0	6.5	60	1000	0.25	5	29.7	115
1N4755A	40.9	43	45.2	6	70	1500	0.25	5	32.7	110
1N4756A	44.7	47	49.4	5.5	80	1500	0.25	5	35.8	95
1N4757A	48.5	51	53.6	5	95	1500	0.25	5	38.8	90
1N4758A	53.2	56	58.8	4.5	110	2000	0.25	5	42.6	80
1N4759A	58.9	62	65.1	4	125	2000	0.25	5	47.1	70
1N4760A	64.6	68	71.4	3.7	150	2000	0.25	5	51.7	65
1N4761A	71.3	75	78.8	3.3	175	2000	0.25	5	56	60
1N4762A	77.9	82	86.1	3	200	3000	0.25	5	62.2	55
1N4763A	86.5	91	95.6	2.8	250	3000	0.25	5	69.2	50

TOLERANCE AND TYPE NUMBER DESIGNATION

2. The JEDEC type numbers listed have a standard tolerance on the nominal zener voltage of ±5%.

SPECIALS AVAILABLE INCLUDE:

3. Nominal zener voltages between the voltages shown and tighter voltage tolerances. For detailed information on price, availability, and delivery, contact your nearest ON Semiconductor representative.

ZENER VOLTAGE (V_Z) MEASUREMENT

4. ON Semiconductor guarantees the zener voltage when measured at 90 seconds while maintaining the lead temperature (T_L) at 30°C ± 1°C, 3/8″ from the diode body.

ZENER IMPEDANCE (Z_Z) DERIVATION

5. The zener impedance is derived from the 60 cycle ac voltage, which results when an ac current having an rms value equal to 10% of the dc zener current (I_{ZT} or I_{ZK}) is superimposed on I_{ZT} or I_{ZK}.

SURGE CURRENT (I_R) NON-REPETITIVE

6. The rating listed in the electrical characteristics table is maximum peak, non-repetitive, reverse surge current of 1/2 square wave or equivalent sine wave pulse of 1/120 second duration superimposed on the test current, I_{ZT}, per JEDEC registration; however, actual device capability is as described in Figure 5 of the General Data — DO-41 Glass.