

104
25j

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA



DISEÑO DE UN SISTEMA PARA DETECCION DE GASES COMBUSTIBLES

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA
EN EL AREA DE INGENIERIA ELECTRICA
Y ELECTRONICA

P R E S E N T A N :
JORGE LUIS ORTEGA ARJONA
ERNESTO RUBIO ACOSTA
FERNANDO SANCHEZ RODRIGUEZ

DIRECTOR DE TESIS:
M. en I. ABEL CLEMENTE REYES

MEXICO, D. F.

AGOSTO 1992



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INTRODUCCIÓN.

En la actualidad, el manejo de los materiales en estado gaseoso tiene lugar en gran parte de los procesos industriales. Esta actividad representa, en mayor o menor grado, un peligro latente para la vida humana y la propiedad. Lo anterior se debe a la susceptibilidad de los materiales gaseosos a presentar violentamente cambios físicos. Durante muchos años, los accidentes por falta de conocimiento y precaución en el manejo de los gases, ha provocado grandes pérdidas, tanto en vidas como en bienes materiales.

Los métodos que se han utilizado para resolver este problema han ido evolucionando con el tiempo. Como en todos los *sistemas de seguridad*, dichos métodos pueden clasificarse en *preventivos* y *correctivos*. Los primeros consideran las causas que anteceden a un desastre; mientras que los segundos, a las consecuencias que resultan del mismo.

El *objetivo* del presente diseño es realizar un sistema para sensar gases explosivos, que sea capaz de adquirir y almacenar datos en forma digital, y de comunicarse con una computadora personal para un análisis posterior de dichos datos.

Este diseño es un sistema de seguridad del tipo preventivo. En cualquier situación donde se involucren vidas humanas, es indispensable tomar medidas que minimicen las causas de posibles accidentes.

Este trabajo se divide en:

El *primer capítulo* trata la definición de gas, sus diferentes clasificaciones por sus propiedades y uso, los riesgos que éstos representan, así como algunos casos particulares.

En el *segundo capítulo*, se explican las tres principales técnicas para el análisis de gases: la fotometría, la cromatografía y la espectroscopía, así como el equipo relacionado con ellas.

En el *tercer capítulo*, se exponen los métodos fundamentales para el muestreo de gases: adsorción, absorción, condensación y desplazamiento, así como las precauciones en el muestreo de gases.

En el *cuarto capítulo*, se dan a conocer los principales transductores asociados a la detección de gases considerando variables físicas como: conductividad térmica, calor de combustión, detección de luz infrarroja, conducción electrolítica, detección de iones específicos, y adsorción en semiconductores.

En el *quinto capítulo*, se presenta la estructura de un sistema digital de adquisición de datos, los elementos que los conforman, y sus ventajas.

En el *sexto capítulo*, se muestra el proceso de diseño considerando sus principios, factores, y limitantes. Se trata la técnica de diseño funcional descendente, y una técnica para la selección de una familia de microprocesadores. Además, se expone el planteamiento para el diseño del Sistema para Detección de Gases Combustibles.

En el *séptimo capítulo*, se ilustran los aspectos en el diseño del Sistema para Detección de Gases Combustibles, tanto en el "*hardware*" como en el "*software*". Se expone la circuitería y la programación desarrolladas para la realización del proyecto.

En el *octavo capítulo*, se proporciona un seguimiento de las pruebas realizadas durante el diseño, a fin de verificar el correcto funcionamiento del mismo.

CAPÍTULO I.

PRINCIPALES GASES LETALES DE USO INDUSTRIAL. CASOS DE ESTUDIO.

I. Introducción.

El término *gas* describe el estado físico de una sustancia que no posee forma ni volumen por sí misma; pero que toma la forma y llena completamente el volumen de cualquier contenedor en que se encuentre. En contraste, las sustancias en estado *líquido* no poseen forma propia, tomando la misma del recipiente que los contiene; pero sí poseen un volumen definido. Además, las sustancias en estado gaseoso son compresibles mientras que las líquidas no lo son. Por otro lado, las sustancias en estado *sólido* poseen ambas características de forma y volumen definidos.

Los gases se componen de partículas extremadamente diminutas, que se encuentran en constante movimiento debido a la cantidad de energía que en ellas se encuentra. Este movimiento afecta de uno u otro modo a las propiedades y al comportamiento de los gases, y se encuentra en función de las condiciones de temperatura, presión y volumen del propio gas.

Este capítulo presenta una definición del estado gaseoso de las substancias, los lineamientos utilizados para la clasificación de los diferentes gases, y los peligros que presentan tanto en su almacenamiento como en el escape de los mismos a la atmósfera. El capítulo concluye con algunos casos de estudio de gases de uso común. Para cada gas, se considera información sobre su clasificación, propiedades químicas, propiedades físicas y su utilización.

II. Definición de Gas.

El término *gas* hace referencia al estado físico de la materia que no posee forma ni volumen definidos.

Si se considera que todas las substancias pueden existir en estado gaseoso dependiendo de las condiciones de temperatura y presión en que se encuentren, el término de gas es muy amplio, reservándose en este capítulo a sólo las substancias que existen en dicho estado en condiciones conocidas como *condiciones normales de temperatura y presión*. Estas condiciones normales corresponden a **21 °C** ó **70 °F** en temperatura, y a **101 kPa** ó **14.7 psia** (libras por pulgada cuadrada) en presión¹. Sin embargo, aún en condiciones normales, o cerca de ellas, varias substancias pueden existir tanto en estado líquido como gaseoso.

El término de gas no se define con precisión en las normas dictaminadas por la "National Fire Protection Association" (NFPA). Sin embargo, dichas normas definen a un líquido como aquél que tenga una

¹ Una atmósfera equivale a 101 kPa.

presión de vapor que no exceda los 275 kPa² a una temperatura de 38 °C. Por exclusión, cualquier sustancia ó mezcla de sustancias que en su estado líquido presente una presión de vapor mayor a los 275 kPa a una temperatura de 38 °C, puede ser considerada como un gas.

La *presión de vapor* de un líquido es la presión ejercida por las moléculas de vapor en equilibrio con el líquido a una determinada temperatura. Es decir, es la presión externa (presión del vapor) que deben vencer las moléculas del líquido para escapar al medio externo.

III. Clasificación de los Gases.

Los gases se encuentran clasificados en base a ciertas denominaciones comunes, donde se reflejan las propiedades químicas y físicas de los mismos, así como sus usos principales. A continuación se consideran las clasificaciones por propiedades químicas, por propiedades físicas y por el uso que presentan.

A. Clasificación por propiedades químicas.

Las *propiedades químicas* de un gas se refieren principalmente a los cambios que experimenta dicho gas en contacto directo con el fuego, y a su capacidad de reaccionar químicamente con otras sustancias o consigo mismo. Lo anterior produciría cantidades potencialmente peligrosas de calor y/o sustancias como producto de la reacción, que podrían afectar a los seres humanos.

A continuación se presenta la siguiente clasificación de los gases atendiendo a sus propiedades químicas:

² Equivale a 2.72 atm (atmósferas).

1. Gases inflamables.

Un *gas inflamable*³ es cualquier gas que enciende en concentraciones normales de oxígeno en el aire.

A partir de lo anterior, se define el *rango de inflamabilidad* de un gas como el intervalo específico de concentraciones de mezcla aire-gas dentro del cual presenta características de inflamabilidad. Este se define a partir de límites superior e inferior. A la concentración mínima se le considera el *límite inferior* de inflamabilidad⁴; mientras que a la concentración máxima, el *límite superior* de inflamabilidad. Además, cada gas arderá a partir de cierta temperatura, que se conoce como *temperatura de ignición*.

2. Gases no inflamables.

Los *gases no inflamables* son aquellos que no encienden en cualquier concentración de aire u oxígeno. Sin embargo, un gran número de gases no inflamables pueden soportar la combustión; esto es que no la extienden. También es posible encontrar otros gases que suprimen la combustión; esto significa que la extinguen.

Aquellos gases que soportan la combustión, se conocen como *gases oxidantes*. Generalmente se componen de mezclas de oxígeno y otros gases; por ejemplo: la mezcla de helio y oxígeno, la de nitrógeno y oxígeno, o algunos óxidos gaseosos

³ Fuente: norma de la "National Fire Protection Association" (NFPA).

⁴ También conocido como Límite Inferior de Explosividad.

como el óxido nitroso. Estas mezclas contienen una cantidad considerablemente mayor de oxígeno que la mezcla de nitrógeno y oxígeno presente en el aire.

Aquellos gases que no soportan la combustión generalmente se denominan *gases inertes*. Entre los más comunes encontramos el nitrógeno, argón, helio y otros gases poco abundantes en la atmósfera tales como el dióxido de carbono y el dióxido de azufre. Sin embargo, existen algunos metales como el magnesio que son combustibles y que pueden reaccionar vigorosamente con el dióxido de carbono o el nitrógeno.

3. Gases reactivos.

El término de *gas reactivo* se utiliza para hacer distinción de aquellos gases que son susceptibles de reaccionar con otros materiales o consigo mismos mediante una reacción puramente química.

Cabe aclarar que la mayoría de los gases pueden ser inducidos a reaccionar químicamente con otras sustancias bajo condiciones específicas como son combustión, calentamiento e impacto físico, aunque no se consideren como gases reactivos propiamente.

Un ejemplo típico de un gas reactivo es el flúor. Este gas es altamente reactivo con prácticamente cualquier sustancia orgánica o inorgánica a condiciones normales de temperatura y presión. El flúor reacciona lo suficientemente rápido como para producir flama. Otro ejemplo lo tenemos en el cloro (un gas no inflamable) que en combinación con el hidrógeno (un gas inflamable), reacciona desprendiendo generalmente flama.

Varios gases pueden reaccionar químicamente consigo mismos cuando se someten a ciertas condiciones ambientales. Ejemplos de estos gases son el acetileno, metil-acetileno, propano, y cloruro de vinilo. Para su transporte y almacenamiento, estos gases generalmente se mezclan con otras sustancias o se manejan en contenedores especiales, para evitar su reacción.

4. Gases tóxicos.

Los gases *tóxicos* son venenosos o irritantes cuando se inhalan o si se entran en contacto directo con el cuerpo. Estos gases representan un serio peligro para la vida si se liberan a la atmósfera; por ejemplo: cloro, ácido sulfhídrico, dióxido de azufre, amoníaco, monóxido de carbono, entre otros.

B. Clasificación por propiedades físicas.

Para propósitos de este capítulo, las *propiedades físicas* de un gas se refieren específicamente al estado que éste presenta en función de las condiciones de temperatura y presión dentro de un contenedor.

El control de las propiedades físicas de un gas es de crucial importancia cuando éste se encuentra confinado en un contenedor durante su transporte, transferencia o almacenamiento. Además, siempre es necesario protegerlo del contacto con el fuego y evitar fugas accidentales. Durante dichos eventos, las propiedades físicas del gas pueden alterarse en gran medida suscitando situaciones peligrosas. En cualquier situación, la cantidad (peso) de un gas es importante, ya que los gases son inherentemente más ligeros que los

líquidos y los sólidos, ocupando un volumen mayor. Esto último dificulta su manejo.

Ahora bien, por razones económicas y por facilidad de empleo, los gases se almacenan en contenedores cuya capacidad permita el manejo de la mayor cantidad de gas posible. En consecuencia, los gases se almacenan preferentemente en estado líquido, aunque también se hace en estado gaseoso.

A continuación se presentan la siguiente clasificación de los gases atendiendo a sus propiedades físicas:

1. Gases comprimidos.

Un *gas comprimido* es aquel que existe en estado gaseoso dentro de un contenedor a temperatura normal (temperatura ambiente) y a una presión mayor que la atmosférica.

La presión de un gas comprimido dentro de un contenedor depende de la temperatura del gas, de la presión inicial de carga, y de la cantidad de gas presente en el contenedor. No existen límites superior o inferior establecidos para la presión dentro de un contenedor. Como referencia, en EE. UU. se acostumbra colocar el límite inferior a 273 kPa^5 , a una temperatura entre $21 \text{ }^\circ\text{C}$ y $38 \text{ }^\circ\text{C}$. El límite superior lo restringe únicamente la resistencia de la estructura del contenedor y, generalmente, se encuentra entre los $12\,512 \text{ kPa}^6$ y los $24\,923 \text{ kPa}^7$.

⁵ Equivale a 2.72 atm (atmósferas).

⁶ Equivale a 123.9 atm (atmósferas).

⁷ Equivale a 247 atm (atmósferas).

La cantidad de gas que puede almacenar un contenedor para gas comprimido es muy limitada. Por ejemplo, el cilindro portátil más grande y comúnmente usado de oxígeno comprimido es de solo 9 kg de oxígeno o, expresando en unidades de volumen, de 6.81 m³ de oxígeno a temperatura y presión normales (véase apéndice A).

2.Gases licuados.

Un *gas licuado* es aquel que existe parcialmente en estado líquido y parcialmente en estado gaseoso dentro de un contenedor, a una temperatura normal (temperatura atmosférica) y a una presión tal que lo obligue a permanecer en dicho estado.

La presión de un gas licuado dentro de un contenedor depende básicamente de la temperatura de la porción líquida.

Un gas licuado es mucho más concentrado que un gas comprimido. Por ejemplo, el antes mencionado cilindro de oxígeno podría contener aproximadamente 53 kg de oxígeno líquido o, en unidades de volumen, 40.1 m³ de oxígeno en condiciones normales (véase apéndice A); lo cual representa seis veces más que como gas comprimido. Esta comparación es válida únicamente para fines ilustrativos, ya que el oxígeno comprimido y el oxígeno líquido no se almacenan en el mismo tipo de contenedores.

3.Gases criogénicos.

Un *gas criogénico* es un gas licuado que existe en un contenedor a una temperatura mucho más baja que la temperatura atmosférica -generalmente por encima de su punto

de ebullición en condiciones normales- y a presiones moderadamente bajas.

El *punto de ebullición* de una sustancia en estado líquido, es la temperatura a la cual la presión de su vapor es igual a la presión del medio exterior.

La diferencia entre un gas criogénico y un gas licuado, es que el primero no se puede retener indefinidamente dentro de un contenedor, ya que el flujo de calor proveniente de la atmósfera tiende a aumentar la presión del contenedor. Dicho flujo de calor puede ser controlado; pero, no eliminado. Si el gas se encuentra confinado totalmente, la presión resultante puede exceder en gran medida la capacidad de esfuerzo de cualquier contenedor.

C. Clasificación por su uso.

Las organizaciones encargadas de estandarizar y regularizar el manejo de los gases, así como las industrias en general, frecuentemente clasifican los gases por sus principales usos. Este esquema de clasificación no es tan preciso como los esquemas precedentes, ya que algunos gases con características poco similares pueden presentar un uso muy parecido.

1. Gases combustibles.

Estos gases son generalmente del tipo inflamable. Se utilizan para quemarse en presencia de aire como una fuente de calor o luz, con fines industriales o domésticos. Hasta ahora, los gases combustibles principalmente utilizados son el gas natural

y los gases licuados derivados del petróleo, como el butano y el propano.

2.Gases Industriales.

Dentro de esta categoría se incluyen toda la gama de gases clasificados según sus propiedades químicas, y que se utilizan en procesos industriales de tratamientos de calor, tratamientos de refrigeración, tratamientos de aguas y procesos químicos.

3.Gases médicos.

La clasificación por uso de estos gases es algo más especializada, ya que éstos se utilizan con propósitos médicos como son la anestesia y la terapia respiratoria. Los gases más comunes para estos tratamientos son el oxígeno y el óxido nitroso.

IV.Peligros en el manejo de los gases.

Es posible hacer una distinción entre los peligros que representan los gases encerrados en un contenedor, de aquellos que significan los gases en estado de fuga; aún cuando ambas acciones puedan presentarse al mismo tiempo.

En cuanto a los gases confinados en un contenedor, el principal riesgo que constituyen es la falla del contenedor donde se encuentran. Por otro lado, refiriéndose a los gases que se fugan a la atmósfera, es la provocación de incendios y explosiones, en el caso de gases inflamables;

envenenamiento, en el caso de gases tóxicos; y en cualquier caso, la asfixia por desplazamiento de oxígeno.

A. Gases encerrados en un contenedor.

Los gases se expanden cuando se les aplica calor, generando un incremento de presión dentro del contenedor. Lo anterior puede ocasionar que falle el contenedor y el gas se fugue. También es importante aclarar que los gases comprimidos y los gases licuados reaccionan de diferente manera cuando se calientan.

1. Gases comprimidos.

Un gas comprimido simplemente trata de expandirse, siguiendo las leyes clásicas del comportamiento de los gases. Ningún gas cumple estas leyes exactamente, aunque las leyes de *Boyle* y de *Charles* son lo suficientemente precisas para predecir el comportamiento de los gases comprimidos bajo las condiciones en que comúnmente se encuentran.

Para aplicar las leyes de Boyle y Charles, es necesario utilizar un conjunto de unidades consistentes en los cálculos, utilizando los valores de presión y temperatura absolutos. La *presión absoluta* se calcula mediante la siguiente ecuación:

$$P_{\text{absoluta}} = P_{\text{manométrica}} + P_{\text{atmosférica}}$$

y; la temperatura absoluta, con:

$$K = ^\circ C + 273$$

En el Sistema Internacional de Unidades bastará considerar: en cuanto a temperatura los Kelvin; en cuanto a presión los Pascales; y en cuanto a volumen los metros cúbicos.

a.Ley de Boyle.

La Ley de Boyle establece que:

"El volumen ocupado por una masa dada de gas varía inversamente con la presión absoluta, si no son permitidos los cambios de temperatura."

Esto es que:

$$P V = cte$$

b.Ley de Charles.

La Ley de Charles establece que:

"El volumen de una masa dada de gas es directamente proporcional a la temperatura absoluta, si se mantiene la presión constante."

Esto es que:

$$\frac{V}{T} = \text{cte}$$

Por lo tanto, si se consideran los límites prácticos de trabajo para la mayoría de los gases, la relación entre temperatura, presión y volumen se puede calcular con precisión mediante la siguiente fórmula:

$$\frac{T_1}{T_2} = \frac{P_1 V_1}{P_2 V_2}$$

donde T_1 , P_1 y V_1 se refieren a las condiciones iniciales; y T_2 , P_2 y V_2 , a las condiciones finales que se desean determinar de un gas.

2. Gases licuados y criogénicos.

Ahora bien, considerando el caso de los gases licuados, incluyendo los gases criogénicos, éstos exhiben un comportamiento más complicado, ya que el resultado neto del calentamiento es la combinación de tres efectos:

- Primero, la fase que se encuentra en estado gaseoso es susceptible a comportarse como un gas comprimido.

- Segundo, la fase que se encuentra en estado líquido se dilata, comprimiendo la fase gaseosa.
- Tercero, la presión de vapor de la fase líquida se incrementa al aumentar la temperatura de ésta.

La combinación final de los tres efectos anteriores da como resultado que la presión total dentro del contenedor aumente cuando se expone a una fuente de calor externa.

Un incremento de presión aún más serio puede originarse si la dilatación de la fase líquida provoca que ésta ocupe todo el volumen disponible dentro del contenedor, condensando la fase gaseosa. Si lo anterior ocurre, una pequeña cantidad de calor aplicada provoca un enorme incremento de presión, ya que la fase líquida tiende a seguir dilatándose. Por esta razón, si la temperatura de la fase líquida es susceptible a elevarse, es muy importante nunca introducir en un contenedor una cantidad de gas licuado (en fase líquida) mayor que aquella que podría almacenarse a temperatura ambiente. Así, siempre habrá espacio para la fase gaseosa.

La cantidad apropiada de la fase líquida depende considerablemente del gas licuado en cuestión y de factores que incrementan la temperatura, como son la temperatura inicial de carga del gas licuado, y el tamaño y ubicación del contenedor. Es decir, si el contenedor se encuentra aislado, instalado a nivel de piso o es subterráneo. Dicha cantidad permitida de gas licuado dentro de un contenedor se conoce como *densidad de llenado* o *densidad de carga*⁸. Esta se puede expresar en términos de masa, o en términos de volumen. Si se especifica en términos de volumen, debe indicarse necesariamente la temperatura a la cual está referida.

⁸ Masa total del gas licuado entre volumen total del contenedor.

B. Fallas en contenedores.

Una *falla* en un contenedor se refiere a la ruptura de éste debido a la sobrepresión interna o a la pérdida de rigidez del contenedor.

La sobrepresión dentro del contenedor se debe al incremento de presión que resulta de la expansión de los gases debido al aumento en la temperatura.

Además, los contenedores pueden perder su rigidez, ya que los materiales con los que se fabrican modifican sus propiedades debido al contacto directo con fuego; a la fatiga estructural producto de esfuerzos a los que se han sometido durante cierto tiempo; a impactos físicos; al deterioro, corrosión u otras causas.

1. Ruptura de contenedores.

Los contenedores para gases comprimidos y gases licuados soportan altos niveles de energía potencial, que puede liberarse en función de la concentración de la materia comprimida o licuada. Una falla en un contenedor libera rápida y violentamente dicha energía potencial, junto con una fuga simultánea de gas hacia los alrededores, y una propulsión del contenedor o de sus partes.

a. Gases comprimidos.

Las fallas en contenedores con gases comprimidos se distinguen más por el peligro que representan los trozos de material impulsados, que por el que significa la

presencia del gas, ya que la cantidad de gas en este tipo de contenedores es poca.

b. Gases licuados.

Las fallas en contenedores con gases licuados, debido a la concentración del gas, sí son susceptibles a liberar una gran cantidad de éste. Un ejemplo de este tipo de fallas es la explosión tipo BLEVE.

Las *Explosiones de Vapores en Expansión y Líquidos en Ebullición (BLEVE[®])* se refieren a los casos en que los contenedores de gas licuado fallan. Principalmente son explosiones de contenedores que liberan grandes cantidades de presión.

Todos los gases licuados se encuentran almacenados a temperaturas sobre su punto de ebullición (a presión normal), y permanecen en estado líquido debido a la alta presión dentro del contenedor.

Si la presión se reduce a la atmosférica, como cuando falla el contenedor, se provoca una rápida vaporización de una porción del gas. Dicha porción es directamente proporcional a la diferencia existente entre la temperatura presente al instante de la falla y la correspondiente al punto normal de ebullición del líquido.

La vaporización del líquido generalmente se acompaña de una enorme expansión, causada por el paso de líquido a vapor. Este proceso de expansión es el que proporciona la energía para la generación de rupturas en

[®] BLEVE es el acrónimo de "Boiling Liquid-Expanding Vapor Explosion".

la estructura del contenedor, y para la propulsión de los fragmentos del mismo.

En el caso de los gases inflamables, al mezclarse rápidamente sus vapores con el aire en presencia de fuego, se incrementa la cantidad de éste. Además, la rápida expansión provoca una atomización del líquido restante que no se ha vaporizado. Muchas de las pequeñas gotas resultantes tienden a encandecer mientras flotan por el aire, aumentando más aún la cantidad de fuego. Sin embargo, también es posible que dichas gotas sean lanzadas fuera de la zona de fuego tan rápidamente que no se encienden, precipitándose en forma líquida.

Aunque la mayoría de las explosiones tipo BLEVE son causadas por la exposición directa del contenedor al fuego, también pueden ocurrir debido a diversas causas, como son la corrosión o impactos contra el contenedor.

2. Combustión de gases dentro de contenedores.

Otro peligro poco frecuente en los gases encerrados dentro de un contenedor, pero no menos significativo, es aquél que resulta de la combustión del propio gas. Lo anterior provoca que falle el contenedor por sobrepresión.

Las mezclas de gases inflamables con oxígeno, que intencionalmente se encierran dentro de un contenedor, son poco comunes; pero es posible que se generen accidentalmente.

La mayoría de las *explosiones por combustión*¹⁰ de gases confinados en contenedores, han ocurrido en aplicaciones industriales o médicas donde el uso de oxígeno o aire comprimido junto con gases inflamables es frecuente.

C. Gases que escapan de un contenedor.

En cuanto a los peligros que representan los gases que escapan de un contenedor, varían según sus propiedades químicas y físicas, así como respecto a la naturaleza del medio en donde se liberan.

Todos los gases, a excepción del aire y oxígeno, representan un cierto peligro para la vida cuando se respiran. El nitrógeno, helio, argón y otros gases incoloros e inodoros, son particularmente peligrosos para la vida cuando escapan de su confinamiento, ya que no son detectados por los sentidos. En algunos casos, el peligro radica en su concentración respecto al oxígeno. La concentración mínima de oxígeno en el aire para asegurar la supervivencia es entre el 6% y 10%; en comparación con el 21% de concentración normal. Sin embargo, a ciertas concentraciones, la coordinación y la percepción se afectan.

1. Gases tóxicos o venenosos.

Los gases tóxicos o venenosos representan un peligro obvio para la vida. Son especialmente peligrosos cuando se liberan a la atmósfera durante una emergencia de fuego, ya que impiden que el auxilio se lleve a cabo de una manera segura.

¹⁰ Las explosiones por combustión se explican más adelante en este capítulo.

2.Oxígeno y otros gases oxidantes.

Aún cuando el oxígeno y los gases oxidantes no son inflamables, pueden acelerar la combustión o provocar igniciones al combinarse con sustancias inflamables, incluso a bajas temperaturas.

3.Gases licuados y criogénicos.

Debido a sus bajas temperaturas, los gases licuados y criogénicos representan un peligro para la vida y la propiedad cuando escapan en forma líquida.

El contacto directo con un líquido frío puede provocar quemaduras y congelación que pueden ser graves si la exposición es prolongada.

Por otro lado, algunos materiales, como el acero y los plásticos, modifican sus propiedades estructurales a bajas temperaturas. Esto podría provocar fallas peligrosas en construcciones o estructuras cercanas a la fuga.

4.Gases inflamables.

Debido a su uso común, el comportamiento de los gases inflamables al ser liberados de sus contenedores son de una mayor preocupación. Al escapar de su contenedor, los gases inflamables representan dos peligros básicamente: las explosiones por combustión y los incendios.

a. Explosiones por combustión.

Una *explosión por combustión* requiere de una mezcla de gas inflamable con aire (oxígeno) acumulada en una zona cerrada, la cual se expone a una fuente de energía.

La presión generada por la mezcla debe superar la resistencia al esfuerzo de la zona cerrada en algún punto. Si la resistencia de dicha zona es suficiente para soportar la presión, no existe posibilidad de una explosión por combustión, ya que la ocurrencia de ésta depende directamente de la naturaleza de la zona cerrada.

Ahora bien, si una zona cerrada se llena de una mezcla de aire con gas inflamable a presión atmosférica, ésta zona deberá soportar una presión hasta de 400 kPa a 750 kPa, de manera que pueda permanecer estable, evitando una explosión por combustión. La presión pueden aumentar si la mezcla involucra un gas reactivo, o si está enriquecida con oxígeno. Sin embargo, las estructuras convencionales sólo son capaces de soportar una presión del orden de 3.5 kPa a 7 kPa. Es esta disparidad entre la resistencia de las estructuras convencionales y la presión generada por la mezcla de aire con gas inflamable, lo que hace posible la ocurrencia de las explosiones por combustión.

Es posible hacer un análisis de la mecánica de la acumulación de los gases en estructuras cerradas, ya que depende de la cantidad, fase y densidad del gas liberado, así como de la ventilación en la estructura.

Se utilizan varios métodos para detectar la presencia de gases en ambientes cerrados. Un ejemplo consiste en la *odorización* de los gases inodoros e incoloros. Este

método se aplica particularmente a los gases combustibles, como lo es el gas natural y el gas LP. Las normas establecen que este tipo de gases deben ser odorizados, de manera que su presencia sea detectada por el ser humano, a concentraciones no mayores a la quinta parte del límite inferior de inflamabilidad. Los odorizantes utilizados son, en general, líquidos volátiles orgánicos químicamente combinados con azufre, cuyo olor es muy penetrante.

Sin embargo, aunque la odorización es un buen método para la detección de gases, posee sus limitaciones. Si no hay personas dentro de la zona cerrada, o se encuentran dormidas, simplemente no se detecta el gas. Además, el olfato tiende a acostumbrarse después de un cierto tiempo a los olores presentes, y la capacidad de percepción de olores varía de una persona a otra.

b. Incendios de Gases Inflamables.

Como es de esperar, la fuga de un gas inflamable en la atmósfera, generalmente provoca un incendio.

Los incendios provocados por gases inflamables pueden considerarse como intentos de explosiones por combustión, en los cuales no se presenta una acumulación de mezcla aire-gas inflamable. Esto ocurre porque no existe un ambiente cerrado donde se acumule dicha mezcla, o bien, porque ésta se enciende prematuramente sin permitir la acumulación.

En el caso de una enorme fuga, es posible que el propio aire o los edificios circundantes compriman el gas lo suficiente para generar una especie de explosión por

combustión. Este fenómeno se conoce como *explosión a campo abierto* o *explosión al aire libre*.

Los gases licuados no criogénicos pueden provocar explosiones a campo abierto, así como también el hidrógeno, etileno, y algunos gases reactivos que poseen una alta velocidad de propagación de llama. Para prevenir este peligro, se emplean sistemas equipados con pilotos que queman el gas presente en el ambiente, la cual suele ser pequeña.

V.Casos de Estudio.

A continuación se presenta una lista de los principales gases de uso industrial, su clasificación, principales propiedades químicas, y utilización.

A.Acetileno.

1.Clasificación.

Reactivo, inflamable, comprimido, industrial.

2.Propiedades químicas.

Se compone de carbono e hidrógeno. A altas presiones se descompone formando carbono e hidrógeno y produciendo calor. Reacciona con ciertos metales, como el cobre, produciendo carburos metálicos, que generalmente son compuestos explosivos. No es tóxico y es inodoro.

3.Utilización.

Se utiliza en procesos industriales químicos y como gas combustible en operaciones de soldadura y corte.

B.Amoníaco anhidro.

1.Clasificación.

Inflamable, licuado (incluso criogénico), industrial.

2.Propiedades químicas.

Se compone de nitrógeno e hidrógeno. Se le conoce comúnmente como *amoníaco*. Tiene una limitada inflamabilidad. El amoníaco húmedo ataca al cobre y al zinc, y a muchas de sus aleaciones como el latón y el bronce. Es poco tóxico. Es irritante y tiene un olor característico.

3.Utilización.

Se le utiliza como fertilizante, como refrigerante y como fuente de hidrógeno para la formación de atmósferas especiales.

C.Dióxido de carbono.

1.Clasificación.

No inflamable (inerte), licuado, industrial.

2. Propiedades químicas.

Se compone de carbono y oxígeno. No es inflamable. No es tóxico; pero puede producir asfixia debido al desplazamiento de aire respirable.

3. Utilización.

Se utiliza para gasificar bebidas, para la formación de atmósferas inertes y como agente extintor de incendios.

D. Cloro.

1. Clasificación.

Reactivo, no inflamable, licuado, tóxico, industrial.

2. Propiedades químicas.

Elemento químico básico. No es inflamable. Es muy reactivo. En contacto con el agua forma ácidos como el hipocloroso y el clorhídrico. Es muy tóxico y se le considera venenoso. Produce quemaduras en la piel. Tiene un olor característico.

3. Utilización.

Se utiliza en procesos químicos industriales, como blanqueador, como purificador de agua potable y para el saneamiento de aguas negras.

E. Etileno.

1. Clasificación.

Inflamable, comprimido, criogénico, industrial.

2. Propiedades químicas.

Se compone de carbono e hidrógeno. Es estable, excepto cuando se encuentra a muy altas presiones. Es muy reactivo. No es tóxico, pero puede producir asfixia. Es anestésico.

3. Utilización.

Se utiliza en la manufactura de plástico de polietileno y en el proceso de maduración de frutas.

F. Hidrógeno.

1. Clasificación.

Inflamable, comprimido, criogénico, industrial.

2. Propiedades químicas.

Elemento químico básico. Posee la mayor *velocidad de combustión* de todos los gases (velocidad de propagación de la flama). No es tóxico.

3. Utilización.

Se utiliza para la hidrogenación de aceites comestibles, en el proceso de soldadura y corte, para la creación de atmósferas especiales y como agente refrigerante.

G. Gas natural licuado (GNL).

1. Clasificación.

Inflamable, criogénico, combustible.

2. Propiedades químicas.

Mezcla de compuestos que tienen carbono e hidrógeno. Su principal componente es el metano; y en menores proporciones, el etano, propano y butano. No es tóxico, pero puede producir asfixia.

3. Utilización.

Se utiliza como combustible y como suministro básico de gas natural.

H. Gas licuado de petróleo (GLP).

1. Clasificación.

Inflamable, licuado (incluso criogénico), combustible.

2. Propiedades químicas.

Mezcla de compuestos que tienen carbono e hidrógeno. Sus principales compuestos son el propano o el butano.

3. Utilización.

Se utiliza como combustible doméstico, comercial, agrícola e industrial y para motores.

I. Oxígeno.

1. Clasificación.

No inflamable (oxidante), comprimido; criogénico, industrial y medicinal.

2. Propiedades químicas.

Elemento químico básico. Reacciona con prácticamente todos los materiales; a esta reacción se le conoce como *oxidación*. La combustión es un tipo especial de reacción de oxidación.

3. Utilización.

Se utiliza en la producción de acero, en el proceso de soldadura y corte, en procesos químicos industriales y para aplicaciones medicinales.

Gas	Límite de inflamabilidad en porcentaje en volumen en aire		Densidad relativa (aire=1)	Cantidad de aire necesaria para que arda 1m ³ de gas. en m ³	Temperatura de ignición en °C
	Inferior	Superior			
Acetileno	2.5	81.0	0.91	11.9	304
Hidrógeno	4.0	75.0	0.07	2.4	400
Amoníaco anhidro	16.0	25.0	0.60	8.3	851
Monóxido de carbono	12.5	74.0	0.97	2.4	609
Etileno	2.7	36.0	0.98	14.3	490

Tabla 1.1: Propiedades de combustión de los gases inflamables más comunes.

CAPÍTULO II.

SISTEMAS PARA EL ANÁLISIS DE GASES.

I. Introducción.

Las técnicas más comunes que se emplean en laboratorio para el análisis e identificación de los componentes de una mezcla de gases son la fotometría, la espectroscopía y la cromatografía.

Este capítulo describe las técnicas fotométrica, espectroscópica y cromatográfica para el análisis de gases, así como los equipos correspondientes que utilizan.

II. Fotometría.

La *fotometría* es el método que se basa en la medición de la *transmitancia* o *absorbancia* propias de la presencia de una solución salina con propiedades absorbentes, de un compuesto, o de una reacción producida por la sustancia a ser determinada (gas).

La transmitancia o absorbancia se definen a partir de la relación entre la potencia transmitida a través de una solución de referencia, con la potencia transmitida a través de una solución absorbente (muestra), que inciden ambas en un mismo detector.

A. Principio del método.

El método se fundamenta en la ley empírica de Beer:

$$P = P_0 \cdot 10^{-abc}$$

donde:

P = potencia radiada recibida a través de una solución absorbente (muestra).

P₀ = potencia radiada recibida a través de una solución de referencia.

- a = *absorbidividad*, que es una constante característica de la solución absorbente, y de la frecuencia de la fuente de energía radiante.
- b = longitud de la celda interna (generalmente en centímetros), que alberga el material absorbente.
- c = concentración de la sustancia absorbente en gramos por litro.

A partir de la razón entre las potencias recibidas se define la *transmitancia (T)* como:

$$T = \frac{P}{P_0}$$

A partir de la transmitancia se define la *absorbancia* como:

$$A = \log_{10} \frac{1}{T} = \log_{10} \frac{P_0}{P}$$

Considerando la ley de Beer, la absorbancia es función de la *absorbidividad (a)*, de la longitud de la celda interna (*b*) y de la concentración de la sustancia absorbente (*c*):

$$A = abc$$

Ya que la absorbitividad (a) y la longitud de la celda interna (b) son constantes, la absorbancia (A) es una función lineal de la concentración de la sustancia absorbente (c).

De la ecuación anterior interesa conocer el valor de la concentración de la sustancia absorbente (c), de tal modo que:

$$c = \frac{A}{ab}$$

es evidente que una vez determinada la absorbitividad (a) para el sistema, la concentración de la sustancia absorbente (c) puede ser obtenida, ya que la longitud de la celda interna (b) se conoce y la absorbancia (A) puede ser medida.

Para determinar el valor de la constante de absorbitividad (a), se emplea la ecuación:

$$a = \frac{A}{bc}$$

sustituyendo el valor medido de la absorbancia (A) para un valor dado de la longitud de la celda interna (b) y de la concentración de la sustancia absorbente (c). Teóricamente, en la determinación de la constante de absorbitividad (a) para un sistema de absorción, será suficiente considerar una sola medición para una longitud de onda dada de la fuente de luz, en una solución de concentración conocida. Sin embargo, es más seguro utilizar el valor promedio obtenido con

tres o más concentraciones, cubriendo así el rango posible sobre el cual serán efectuadas las mediciones, y realizando varias lecturas para cada concentración.

La validez de la ley de Beer para un sistema particular puede ser probada demostrando que la absorbitividad (a) permanece constante cuando la longitud de la celda interna (b) y la concentración de la sustancia absorbente (c) varían.

Además, es necesario determinar la región espectral involucrada, ya que la absorbitividad está en función de la frecuencia de la fuente de energía radiante.

En general, es deseable utilizar un filtro óptico tal que la porción espectral que aísle se encuentre en la región de máxima absorbitividad. En el mejor de los casos, la curva de absorbitividad contra frecuencia es lo suficientemente plana y amplia en la región de máxima absorbitividad, para que las desviaciones de la ley de Beer resultantes del uso de fuentes de energía radiante heterogéneas sean despreciables.

Sin embargo, en algunas ocasiones no es posible o deseable trabajar en el punto de máxima absorbitividad, debido a la interferencia de otras sustancias absorbentes en la solución, o a que la región de máxima absorbitividad es estrecha. En estos casos es recomendable encontrar otra porción plana de la curva donde las mediciones estén libres de interferencia. Cuando no exista una porción plana libre de interferencia, entonces será necesario trabajar en una porción de la curva con cierta pendiente. Entonces, la ley de Beer será válida solamente si la porción espectral aislada es estrecha.

B.El Fotómetro.

Un *fotómetro* consta básicamente de cuatro elementos:

- La fuente de energía radiante.
- Filtros o monocromadores.
- Celdas de absorción.
- Detectores fotoeléctricos.

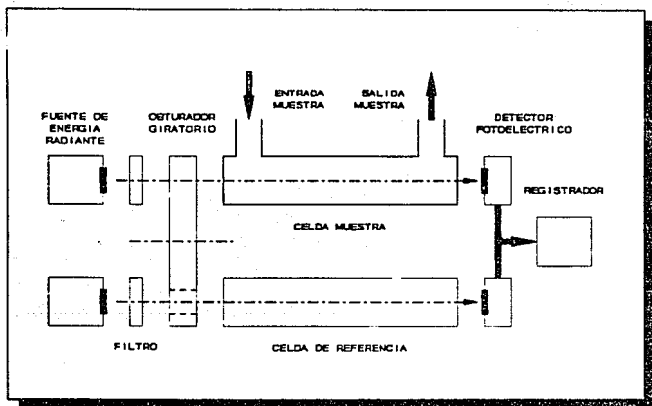


Figura 2.1: Esquema funcional de un *fotómetro*.

Se muestra el esquema funcional de un fotómetro en la Figura 2.1.

1. La fuente de energía radiante.

La *fuentes de energía radiante* genera el haz de luz que se utiliza en las mediciones de potencia. Generalmente se trata de una lámpara incandescente con filamento de tungsteno. Las características deseables en la lámpara incandescente son: bajo consumo de potencia, largo tiempo de vida, independencia con respecto a las fluctuaciones de la línea, distribución estable y uniforme de energía y bajo costo de operación.

2. Filtros o monocromadores.

Los *filtros* y los *monocromadores* son dispositivos que seleccionan cierta banda angosta de frecuencias de la energía radiada, de tal manera que ésta resulte monocromática.

Los *filtros* atenúan ciertas componentes del haz luminoso; se fabrican de vidrio, soluciones acuosas o sustancias gelatinosas.

Los *monocromadores* dispersan las diversas componentes del haz luminoso; por ejemplo, las rejillas de dispersión y los prismas.

3. Celdas de absorción.

Las *celdas de absorción* constituyen el volumen de control en donde se realiza el fenómeno de absorción. Sirven para mantener la solución en estudio (muestra y solvente), y la solución de referencia (solvente).

4. Detectores fotoeléctricos.

Los *detectores fotoeléctricos* son elementos pasivos o activos que en función de la luz incidente, varían alguna de sus características (resistencia, cantidad de portadores) o generan un voltaje entre sus terminales. Se emplean para detectar la cantidad de potencia en la luz incidente.

Finalmente, el *espectrofotómetro* es un fotómetro de precisión que emplea monocromadores capaces de proporcionar energía radiada de alta pureza en cualquier longitud de onda elegida dentro de su rango de funcionamiento.

III. Cromatografía.

La *cromatografía* es una técnica para la separación, cuantificación e identificación de compuestos estrechamente relacionados.

En la cromatografía de gases, la muestra se vaporiza (si todavía no se encuentra en este estado), y es impulsada por un flujo de gas inerte (*gas portador*) para pasar a través de un contenedor rígido (*columna*) que alberga un material de relleno. Este material de relleno tiene diferentes afinidades para cada componente en particular de la mezcla, y permite que cada componente pase a través de él a diferente velocidad. Entonces cada componente emerge de la columna en tiempos diferentes, y se detectan mediante un dispositivo sensible.

La cromatografía de gases es apropiada para la separación y análisis de mezclas de gases y materiales volátiles. Se aplica principalmente al análisis de hidrocarburos líquidos y gaseosos del petróleo, y al análisis de los productos de reacciones orgánicas.

A. Principio del método.

El método cromatográfico posee dos componentes: el componente estacionario y el componente en movimiento.

Un componente es fijo y se conoce como *componente estacionario*, puede ser un sólido o un líquido suspendido por un sólido. Su función principal radica en separar la muestra en sus diferentes compuestos.

El otro componente es móvil y se conoce como *componente en movimiento*, puede ser un gas, un líquido o un sólido volátil. Su función principal es transportar la muestra a través del componente estacionario.

Cuando ocurre el equilibrio de fases entre los compuestos de la muestra, éstos se distribuyen entre los componentes estacionario y en movimiento.

Este método se lleva a cabo de la siguiente manera: una pequeña muestra de la mezcla a separar se inyecta en un flujo de gas inerte (nitrógeno, hidrógeno, dióxido de carbono, argón o helio), que corre por una columna que contiene un medio apropiado capaz de retardar gradualmente el flujo de cada uno de los compuestos individuales de la muestra. En consecuencia, los compuestos separados emergen a intervalos discretos (característicos de cada compuesto) y pasan a través de algún tipo de detector. Las diferencias en la capacidad de adsorción, o la distribución de los compuestos de la muestra en el material de la columna, es el factor que hace posible la separación.

Como se puede apreciar, el principio de la cromatografía de gases es muy simple. Sin embargo, se requiere de un aparato y un control más sofisticado.

B.El Cromatógrafo.

El *cromatógrafo* de gases consta de los siguientes elementos:

- El gas portador.
- El sistema de introducción de la muestra.
- La columna.
- El detector.

Se muestra el esquema funcional de un cromatógrafo en la Figura 2.2.

1.El gas portador.

El componente en movimiento que se emplea para transportar la muestra a través de la columna se conoce como *gas portador*.

Los gases portadores empleados se encuentran de preferencia en estado puro y envasados a presión. Para obtener un resultado óptimo es necesario eliminar el vapor de agua del gas portador.

La elección del gas portador está en función de la naturaleza de la muestra, así como del tipo de detector que se va a emplear. Por ejemplo, el hidrógeno y el helio son los gases más apropiados cuando se emplea un detector basado en conductividad térmica (*catarómetro*); debido a que ambos tienen una buena conductividad térmica, el detector ofrecerá una rápida respuesta.

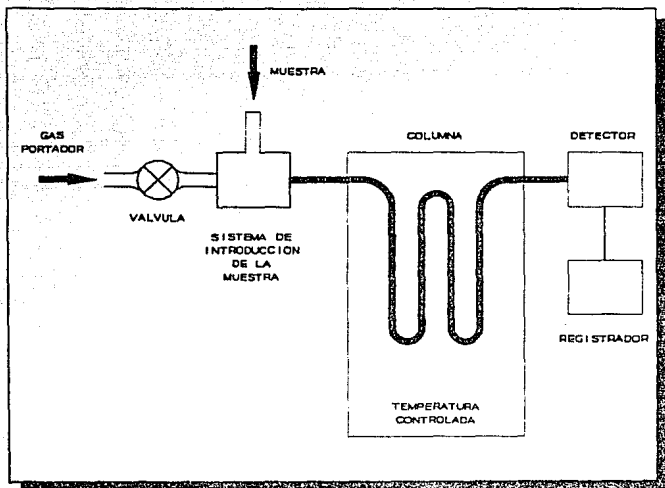


Figura 2.2: Esquema funcional de un cromatógrafo.

2. Sistema de introducción de la muestra.

El sistema de introducción debe ser capaz de depositar instantáneamente la muestra, sin que sufra ninguna alteración a la entrada de la columna.

La cantidad de muestra que se debe introducir se encuentra en función de la naturaleza de ésta, del tamaño de la columna y del tipo de detector. La muestra puede introducirse

por medio de una pequeña jeringa calibrada o a través de un válvula de admisión de dicha muestra. Sin embargo, en la cromatografía de gases se emplean muestras pequeñas.

3.La columna.

La *columna* es un contenedor rígido que sostiene el componente estacionario del cromatógrafo. Puede tener varias formas dependiendo del espacio disponible. Se fabrica con diversos materiales inertes, tales como metal, vidrio y plástico.

Las columnas de metal son las más apropiadas debido a que son inertes, resistentes y además cuentan con excelentes propiedades térmicas. Las columnas de vidrio son también inertes, pero son frágiles y difíciles de calentar. En cuanto a las columnas de plástico (polietileno o nylon) son relativamente sencillas de calentar y enfriar; sin embargo, no es posible trabajar con ellas a altas temperaturas.

4.Detectores.

El *detector* es un aparato que mide los cambios en el componente en movimiento del cromatógrafo.

Existen diversas clases de detectores dependiendo de su principio de funcionamiento:

a. Conductividad térmica (catarómetro). Mide los cambios en la capacidad térmica de la muestra.

b.Densidad del gas. Mide los cambios en la densidad de la muestra.

c.Ionización por flama. Mide las diferencias en la ionización debido a la combustión de la muestra.

d.Ionización por rayos beta. Mide el flujo de corriente entre dos electrodos debido a la ionización del gas por radioactividad.

e.Ionización por luz. Mide el flujo de corriente entre dos electrodos debido a la ionización del gas por radiación ultravioleta.

f.Descarga luminiscente. Mide la variación en la diferencia de potencial entre dos electrodos, causada por el cambio de descarga luminiscente debida a las diferencias en la composición del gas.

g.Temperatura de flama. Mide los cambios en la temperatura de la flama causados por las diferencias en la composición del gas.

h.Constante dieléctrica. Mide los cambios en la constante dieléctrica entre las placas de un capacitor causados por las diferencias en la composición del gas.

IV. Espectroscopía.

La *espectroscopía* se encarga de obtener los posibles niveles de energía o estados característicos de un sistema atómico o molecular a partir de las frecuencias medidas en el espectro.

El estudio de las espectroscopía se divide en *espectroscopía de emisión* y *espectroscopía de absorción*.

Se obtiene un *espectro de emisión* por el análisis espectroscópico de una fuente de luz, como puede ser una llama o un arco eléctrico. Este fenómeno es causado, fundamentalmente, por la excitación de átomos por medios térmicos o eléctricos; la energía absorbida induce los electrones, que se encuentran en un estado fundamental a un estado de mayor energía. El tiempo de vida de los electrones en esta situación es corto y vuelven a un estado de excitación más bajo o al estado fundamental; la energía se libera en forma de luz.

Un *espectro de absorción* se obtiene colocando la sustancia entre el espectrómetro y una fuente de energía que proporciona radiación electromagnética en el intervalo de frecuencias que se desea estudiar. El espectrómetro analiza, para una determinada frecuencia, la energía transmitida con referencia a la energía incidente.

A. Principio del Método.

El fundamento de la medición espectroscópica es el hecho que muchas moléculas interactúan con determinado tipo de energía radiante en forma predecible. El término que comúnmente se utiliza para designar este fenómeno es *absorción de luz*, esto significa que la sustancia irradiada por energía luminosa absorbe algunas de sus frecuencias y refleja otras. Esto se manifiesta en cualquier objeto, cuando se observa que tiene color.

La *luz* es un tipo común de energía. Como otros tipos de energía, se manifiesta mediante partículas en movimiento, capaces de transmitir energía a otras partículas.

La luz, como entidad física, se comporta como una onda y como una partícula. Las partículas de luz se conocen como *quanta*¹ o *fotones*. Los quanta viajan como una onda en movimiento; la distancia entre las crestas adyacentes de la onda (longitud de onda) es característica para cada tipo de quantum.

De esta manera, el *espectro electromagnético* representa las formas de radiación que la luz posee, y que difieren entre sí únicamente por su longitud de onda; por ejemplo: infrarrojo, luz visible, ultravioleta, rayos X, etc.

Partiendo de la *ecuación de Max Planck*, la energía que posee cada componente del espectro es inversamente proporcional a su longitud de onda:

$$E = \frac{hc}{\lambda}$$

donde:

- λ : longitud de onda.
- h : constante de Planck ($6.624 \cdot 10^{-27}$ ergio/segundo).
- c : velocidad de la luz en el vacío ($3 \cdot 10^{10}$ cm/s).

¹ Singular quantum.

Los efectos de la incidencia de los diversos tipos de radiación sobre la materia serán progresivamente más severos a medida que la longitud de onda sea más corta. Así tenemos que:

1. Infrarrojo. Produce un incremento en la vibración de los átomos y de las moléculas generando calor; no altera la estructura electrónica.

2. Luz visible. Los electrones de valencia son llevados a un estado de energía más alto, dando lugar a varios fenómenos como la fluorescencia, fosforescencia y calor. Los cambios en la estructura electrónica no son permanentes.

3. Ultravioleta. Para el caso de las longitudes de onda mayores se comporta de igual modo que la luz visible; pero para el caso de las longitudes de onda menores pueden liberar los electrones de valencia completamente fuera del átomo, causando ionización y, subsecuentemente, otras alteraciones químicas.

4. Rayos X. Causa la liberación de los electrones más cercanos al núcleo (que no son electrones de valencia), produciendo una alteración molecular considerable (destructiva). Estos espectros son característicos de cada átomo y prácticamente no dependen de los enlaces químicos, esto es, de la molécula en que se encuentran los átomos. De este modo, proporcionan información acerca de los elementos que conforman las moléculas del gas.

La mecánica cuántica indica que partes de los quanta no pueden excitar los electrones. Solamente los quanta con un contenido específico de energía (igual a la cantidad necesaria para llevar un electrón de un estado fundamental a un estado de excitación), son capaces de excitar los electrones de una molécula o átomo. Esta cantidad de energía no es la misma para todos los átomos y moléculas, sino que varía respecto al átomo o molécula involucrado. Cuando los quanta de un contenido energético adecuado inciden sobre una molécula determinada, estos parecen "desaparecer" del espectro, es decir, que una longitud de onda de luz particular se absorbe por el compuesto en cuestión. Por lo tanto, la absorción de luz de cierta longitud de onda es característica para cada molécula, lo que permite identificarla.

B.El Espectroscopio.

Cualquier instrumento que se utiliza para producir o estudiar un espectro, en cualquier región espectral se puede denominar *espectroscopio*. Específicamente esta palabra designa aquellos instrumentos que permiten la visión directa del espectro por el ojo. Si el aparato registra el espectro sobre una placa fotográfica (espectrograma), se denomina *espectrógrafo*. En el caso que proporcione directamente una lectura de la longitud de onda o frecuencia de la radiación, el aparato se llama *espectrómetro*; si al mismo tiempo da también una lectura de la intensidad de esa radiación, se le suele llamar *espectrofotómetro*.

Un espectroscopio consta de:

- Una fuente de radiación.
- Un sistema analizador.
- Un detector.

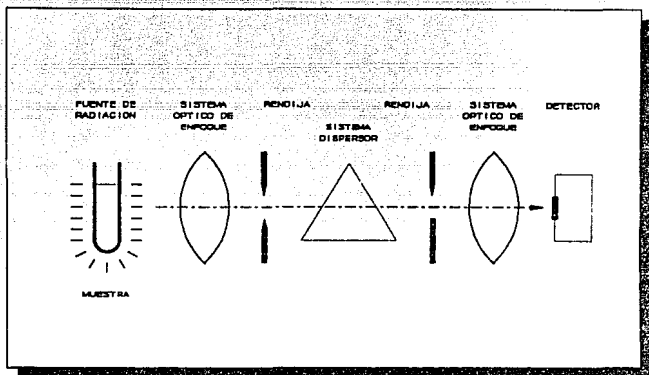


Figura 2.3: Esquema funcional de un *espectrómetro de emisión*.

Se muestra un esquema funcional de un *espectrómetro de emisión* en la Figura 2.3; mientras que en la Figura 2.4, el de un *espectrómetro de absorción*.

La configuración de los componentes anteriormente mencionados puede ser variable, según la zona espectral y la naturaleza del fenómeno que se estudia.

El *sistema analizador*, o espectroscopio propiamente dicho, suele constar de tres elementos principales. Primero, una *rendija* por la que penetra en el aparato un fino haz de la radiación que se pretende analizar. Segundo, un *sistema dispersor*, constituido generalmente por uno o más prismas o redes de difracción, que produce una separación de las distintas longitudes de onda que

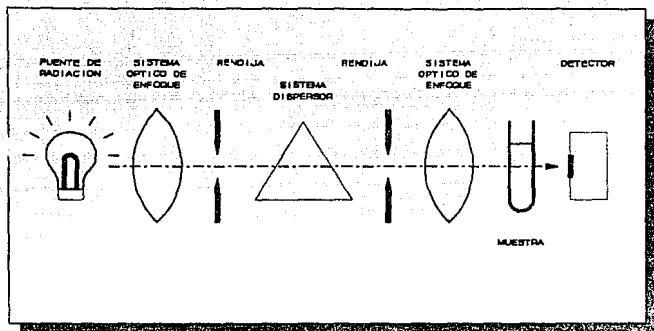


Figura 2.4: Esquema funcional de un *espectrómetro de absorción*.

componen el haz de radiación. Y tercero, un *sistema óptico de enfoque*, constituido casi siempre por combinaciones de lentes o espejos, que da lugar a que se formen en su superficie focal tantas imágenes de la rendija de entrada como radiaciones de distinta longitud de onda componen a la luz que se analiza.

CAPÍTULO III.

MÉTODOS PARA MUESTREO DE GASES.

I. Introducción.

Los gases y vapores siguen las leyes normales de *difusión*¹, mezclándose libremente con la atmósfera que los rodea. Esto es, las fuerzas inerciales y electrostáticas externas no afectan notablemente el movimiento de las partículas en los gases. Debido a lo anterior, deben tomarse ciertas precauciones en el muestreo de gases y, adicionalmente, acondicionar la muestra.

¹ Ley de difusión de Graham.

Este capítulo presenta acciones para acondicionar la muestra de gases, así como precauciones que deben tomarse durante esta acción. Además, menciona las principales técnicas para el muestreo de gases: la adsorción, la absorción, la condensación y el desplazamiento de sustancias.

II. Acondicionamiento de la muestra.

El *acondicionamiento de la muestra* consiste en la alteración de la muestra de modo que sea más fácil captar el componente gaseoso deseado. El acondicionamiento puede consistir en:

A. Filtrado.

El *filtrado* consiste en la eliminación de partículas sólidas o líquidas, que se encuentran suspendidas en el gas.

B. Secado.

El *secado* consiste en la eliminación de la humedad contenida en la muestra.

C. Reacciones químicas.

Existen una gran variedad de *reacciones químicas* para eliminar contaminantes o gases no deseados en la muestra que puedan intervenir en mediciones posteriores.

III. Precauciones en la recolección de la muestra.

En el intervalo existente entre la recolección y el análisis de una muestra de gases, deben considerarse principalmente los siguientes factores para conservar la integridad de la misma:

A. Cambios químicos.

Cambios químicos tales como interacciones químicas entre los componentes de la muestra recolectada, o debidos a la incidencia de luz.

B. Adsorción.

Adsorción de los gases de la muestra en las paredes de contenedores.

C. Fugas.

Fugas en el contenedor donde se almacena la muestra por encontrarse mal sellado, agrietado o no ser el adecuado. También, al trasladar la muestra de un contenedor a otro.

En general, se procura que el intervalo transcurrido entre la recolección de la muestra y su análisis sea lo más breve posible, protegiéndola contra la luz y el calor, y evitando las fugas.

IV. Métodos de recolección de la muestra.

Existen varios *métodos para recolectar muestras de gases*. Los principales se basan en la adsorción, absorción o condensación del compuesto gaseoso de interés, o bien, simplemente confinan la muestra por desplazamiento en un volumen cerrado.

A. Recolección de la muestra por adsorción.

Cualquier gas o vapor se adherirá a la superficie de un sólido con el que se encuentre en contacto. Este fenómeno se conoce como *adsorción* y se debe principalmente a la porosidad de los materiales sólidos.

Los sólidos porosos exponen además de una superficie exterior visible, una intrincada red de canales internamente diminutos con poros submicroscópicos dentro de ellos. Esto último determina las *características adsorbentes* de cada material.

Los *materiales adsorbentes* más comunes son el carbono activado², el óxido de silicio gelatinoso, el óxido de aluminio activado y ciertas tierras activadas. Todos estos difieren ampliamente en el número y tipo de sustancias que adsorben, así como en la cantidad de sustancia adsorbida que pueden retener.

Principalmente, los materiales adsorbentes se pueden clasificar en dos grandes grupos.

² El carbono activado es carbono poroso sin otros materiales que lo contaminen.

1. Clasificación de los materiales adsorbentes.

Los materiales adsorbentes pueden ser de dos tipos: *polares* y *no polares*. Ambas clases se pueden impregnar de ciertas sustancias para obtener un mejor rendimiento durante la adsorción de gases y vapores específicos.

a. Materiales adsorbentes polares.

En general, los tipos de materiales *adsorbentes* que contienen silicio, óxidos metálicos y ciertas tierras activadas son eléctricamente *polares*; esto es, que su estructura molecular contiene una distribución eléctrica no simétrica.

Puesto que las sustancias polares tienden a atraerse fuertemente entre sí y debido a que el agua es un compuesto altamente polar, entonces los adsorbentes polares poseen la capacidad de retener la humedad atmosférica que los rodea.

Compuestos polares como el óxido de silicio gelatinoso y el óxido de aluminio activado se utilizan en muestreos de corta duración en atmósferas que contienen altas concentraciones de gas o vapor, o que están lo suficientemente secas, de tal manera que el adsorbente no se sature de humedad antes de completar el muestreo.

b. Materiales adsorbentes no polares.

Los materiales *adsorbentes no polares*, por otro lado, poseen una estructura molecular tal que exhiben una

simetría en su distribución eléctrica. Tal es el caso del carbono activado.

Ya que el carbono activado es no polar, es capaz de adsorber adecuadamente gases orgánicos y vapores en presencia de atmósferas húmedas. De hecho, el carbono activado con humedad previamente adsorbida, la perderá al ser desplazada por los gases orgánicos y vapores que adsorba de la atmósfera. Tal desplazamiento de humedad no se presenta en los materiales adsorbentes polares.

2.Comparación entre materiales adsorbentes polares y no polares. Caso particular.

La siguiente comparación entre el *carbono activado* y el *óxido de silicio gelatinoso* señala las diferencias entre estos dos tipos de adsorbentes utilizados en el muestreo de gases y vapores atmosféricos.

- a. El carbono activado posee una mayor capacidad para adsorber y retener muestras de diferentes gases y vapores; por otro lado, el óxido de silicio gelatinoso presenta una mayor selectividad en cuanto a los gases y vapores que adsorbe.
- b. El carbono activado es más adecuado para recolectar mezclas de gases y vapores, especialmente cuando éstas sean de poca concentración y, por lo tanto, deban ser recolectadas en largos periodos de tiempo de muestreo, ya que no adsorbe humedad.
- c. La *deadsorción* de gases y vapores contenidos en el óxido de silicio gelatinoso se realiza más fácilmente que en el carbono activado; por esto, es posible

obtener algunas separaciones de gases y vapores por deadsorción selectiva en el óxido de silicio gelatinoso.

3. Deadsorción.

Al proceso inverso a la adsorción, por el cual se obtiene de un material adsorbente los gases y vapores adsorbidos con anterioridad, se le conoce como *deadsorción*.

El proceso de deadsorción se utiliza para "limpiar" al material adsorbente, ya que el material muestreado reduce gradualmente la superficie disponible del material adsorbente, inhibiendo la adsorción de los gases y vapores en estudio. Esto último se conoce como *envenenamiento* del material adsorbente.

a. Deadsorción en carbono activado.

Para el caso del carbono activado, los gases y vapores adsorbidos se deadsorben mediante:

- (1) Desplazamiento del material adsorbido por un flujo de material sobrecalentado.
- (2) Calentamiento del carbono activado en el vacío, destilando el material adsorbido a través de trampas de enfriamiento.

b. Deadsorción en óxido de silicio gelatinoso.

Para el caso del óxido de silicio gelatinoso, es posible lograr la deadsorción utilizando:

- (1) Una corriente de aire libre de contaminantes, calentada a 350°C.
- (2) Solventes polares, tales como agua o alcoholes.

B. Recolección de la muestra por absorción.

El fenómeno de *absorción* se lleva a cabo cuando se pone en contacto una mezcla de gases con un líquido y consiste en el paso de uno o más componentes de la fase³ gaseosa a la fase líquida, llamada solvente.

La *solubilidad* de un gas en un líquido obedece a la ley de Henry que dice: "a temperatura constante la concentración de un gas disuelto en un líquido es proporcional a la presión parcial del gas sobre el líquido".

El concepto de *presión parcial* se ilustra en la ley de Dalton, la cual dice: "a temperatura constante la presión ejercida por una mezcla de gases en un volumen definido es igual a la suma de las presiones individuales que cada gas ejerce si ocupase él solo⁴ el volumen total".

³ Aquí, *fase* es sinónimo de estado.

⁴ Léase: *en forma individual*.

Cabe hacer notar que el fenómeno de absorción es consecuencia de la difusión. La *difusión* es la tendencia de un gas a extenderse y distribuirse homogéneamente a lo largo de todo el volumen que ocupa sin que ocurran reacciones químicas. Dicha tendencia se debe al movimiento continuo de las partículas del gas.

Finalmente, los fenómenos de absorción y de adsorción son diferentes. En la **absorción** un fluido "atrapa"⁵ a un gas y éste último acaba por difundirse por todo el volumen del líquido. Por otro lado, en la **adsorción** un sólido "atrapa"⁶ a un gas y este fenómeno solamente tiene lugar en la superficie del sólido.

C. Recolección de la muestra por condensación.

Cuando las moléculas de un material en estado gaseoso ceden la energía cinética que poseen, bajo ciertas condiciones de temperatura y presión, es posible que se realice un cambio de estado gaseoso a líquido. Al proceso por el cual se extrae calor de un gas, a fin de obtenerlo en estado líquido, se le conoce como *condensación*.

La concentración de gases y vapores atmosféricos por condensación a bajas temperaturas posee algunas ventajas sobre otros métodos de recolección. Primero, el material que se recolecta está inmediatamente disponible para un análisis o separación subsecuente, sin que se requiera remoción por solventes o deadsorción de materiales adsorbentes. Segundo, la condensación es el método más seguro para la preservación de las muestras de gases y vapores, ya que no ocurre ninguna reacción química. Sin embargo, la principal desventaja que presenta es que, adicionalmente, se

⁵ Léase absorbe

⁶ Léase adsorbe.

obtienen grandes cantidades de agua que se condensan junto con los gases y vapores recolectados.

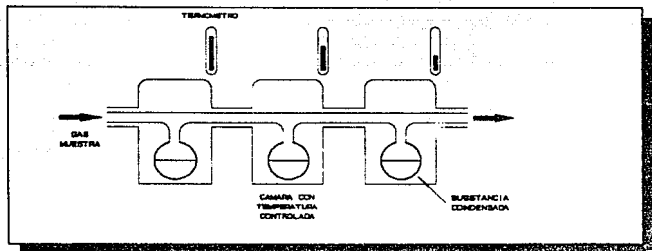


Figura 3.1: Esquema funcional de un dispositivo para recolectar muestras por condensación.

El método más utilizado emplea una serie de cámaras a temperaturas progresivamente más bajas, logrando una condensación paulatina de la muestra a medida que avanza a través de ellas (Ver Figura 3.1). De esta manera, los componentes de la muestra se van separando, dado que cada uno se condensa a diferente temperatura.

D.Recolección de la muestra por desplazamiento.

El método de *desplazamiento* consiste en confinar una muestra de gases dentro de un contenedor, a causa de la extracción del contenido de éste. La muestra ingresa al contenedor por diferencia de presiones. Véase Figura 3.2.

Para obtener una muestra de gases o vapores de la atmósfera por desplazamiento, se utilizan dos dispositivos:

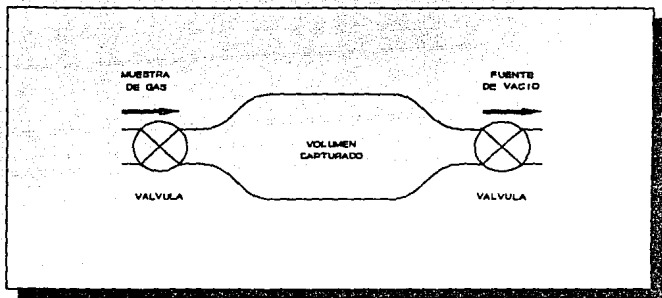


Figura 3.2: Esquema funcional de un dispositivo para recolectar muestras por desplazamiento.

1. Contenedores rígidos al vacío.

Previamente se extrae el aire del interior del contenedor mediante una bomba de vacío. En tales condiciones, éste se sella. Una vez que el contenedor se ubica en el lugar donde se desea tomar la muestra se rompe su sello, de tal manera que la muestra de gas entra al contenedor por la diferencia de presiones. Después que se ha tomado la muestra, se cierra el contenedor con un tapón especial.

Adicionalmente el contenedor podría llenarse con un fluido absorbente si se desea muestrear un gas específico.

2.Contenedores rígidos a presión atmosférica.

Este método consiste en extraer un fluido que se encuentra previamente dentro del contenedor, al mismo tiempo que por otro punto entra la muestra del gas. Generalmente el fluido es aire o agua.

En el caso del aire, éste se extrae por medio de un aspirador o de una bomba de vacío. Para el caso del agua, ésta se extrae a través de un grifo por efecto de la gravedad.

El muestreo por desplazamiento se utiliza para gases que no sean muy reactivos, ya que éstos podrían reaccionar con las paredes del contenedor, con su tapón, con la humedad dentro de él o consigo mismos.

CAPÍTULO IV.

TRANSDUCTORES PARA DETECCIÓN DE GASES.

I. Introducción.

La detección de gases y sus mezclas, se ha convertido en un importante campo de acción para las técnicas industriales relacionadas con la medición electrónica.

Para poder determinar la presencia de un gas, se consideran algunas de sus propiedades físicas y químicas, tales como:

1. Propiedades térmicas: conductividad térmica, calor de combustión.
2. Propiedades relacionadas con la emisión o absorción de radiación infrarroja.
3. Propiedades eléctricas (métodos electroquímicos).
4. Propiedades relacionadas con la adsorción.

Considerando las propiedades anteriores, en éste capítulo se describirán diferentes tipos de transductores útiles en la detección de gases. Además, se hablará de las principales unidades de medición relacionadas con la presencia de un gas.

II. Unidades de medición para gases.

El Sistema Internacional de Unidades establece la unidad fundamental *mol* para cuantificar la cantidad de sustancia. Un *mol* se define como:

" La cantidad de sustancia de un sistema que contiene tantas entidades fundamentales como átomos existentes en 0.012 kilogramos de carbono 12. "

En la práctica, la cantidad de sustancia que posee un gas disuelto en el aire, se puede expresar en función de su masa o relacionándola con el volumen que ocupa. Si se cuantifica la cantidad de sustancia en función

de su masa, se utiliza la unidad *gramo por metro cúbico* (10^{-3}Kg/m^3), o más frecuentemente, *microgramo por metro cúbico* (10^{-9}Kg/m^3). Ahora bien, si se cuantifica la cantidad de sustancia relacionándola con el volumen que ocupa el gas, se emplea la unidad *partes por millón (ppm)*:

$$1 \text{ ppm} = \frac{1 \text{ volumen de gas}}{10^6 \text{ volúmenes de gas más aire}}$$

o bien, expresado en porcentaje:

$$1 \text{ ppm} = 0.0001\% \text{ por volumen}$$

Para el caso de gases combustibles, también se mide su concentración basándose en un *porcentaje* de su *límite inferior de explosividad* (%LEL). El límite inferior de explosividad (LEL) es la concentración mínima a partir de la cual una mezcla de gas combustible con aire presenta características de inflamabilidad, es decir, que la mezcla es capaz de mantenerse encendida por sí misma, propagando la flama. Si se considera el límite inferior de explosividad de un gas como la unidad, la concentración del gas se puede expresar como una fracción porcentual de éste.

III. Transductores basados en la conductividad térmica.

El fenómeno de la *conductividad térmica* consiste en la transferencia de calor causada por la diferencia de temperatura entre dos partes adyacentes de una sustancia.

El transductor asociado con la conductividad térmica es el *catarómetro*. Se utiliza para la detección de gases puesto que compara la conductividad térmica de una muestra de gas, con la conductividad térmica de un gas de referencia. Debido a que cada gas posee una conductividad térmica específica, es posible determinar de cuál gas se trata.

A. El catarómetro.

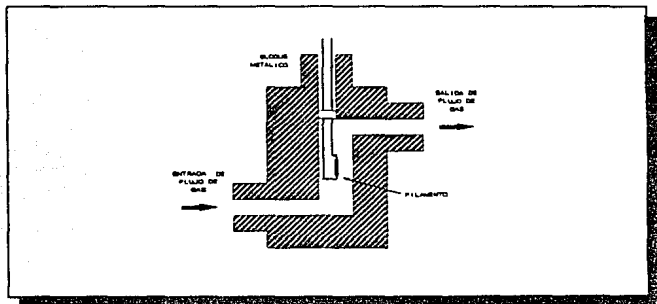


Figura 4.1: Esquema funcional de la *unidad fundamental* de un catarómetro.

La *unidad fundamental del catarómetro* consta de una celda que alberga un filamento metálico, por la cual fluye un gas. (Véase Figura 4.1). Al filamento se le aplica una diferencia de potencial constante que ocasiona la circulación de una corriente eléctrica. La intensidad de corriente eléctrica depende de la resistencia del filamento y ésta, a su vez, varía de acuerdo a su temperatura. Finalmente, la temperatura del filamento está en función de la conductividad térmica del gas que lo rodea.

Por tanto, la conductividad térmica del gas albergado en la celda es proporcional a la intensidad de corriente eléctrica que circula por el filamento.

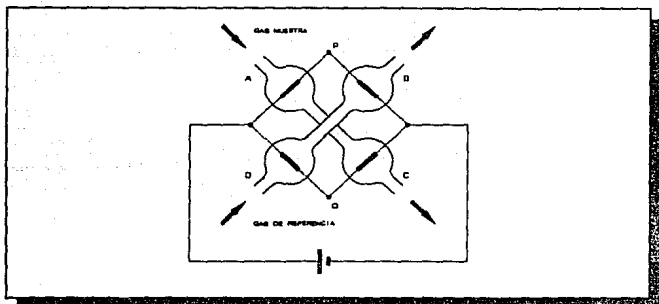


Figura 4.2: Esquema funcional de un catarómetro.

En el catarómetro, cuatro filamentos de platino con la misma resistencia, se encierran dentro de celdas separadas en el interior de un bloque metálico común. Cada celda forma un brazo de un puente de Wheatstone. En la Figura 4.2 se muestra un sistema simplificado.

Al fluir el mismo gas por las cuatro celdas, cada filamento alcanza igual temperatura y resistencia. Entonces circula la misma intensidad de corriente por ambas ramas del puente. Por lo tanto, la diferencia de potencial entre los puntos P y Q es nula. Lo anterior constituye un mecanismo de calibración.

Ahora bien, se introduce en el puente dos mezclas de gases con conductividades térmicas diferentes; por ejemplo: aire en las celdas de referencia D y B, y el gas problema (CO_2) en las celdas A y C. Puesto que la constante térmica del aire es mayor que la del CO_2 , los

filamentos **B** y **C** perderán más calor que los filamentos **A** y **C**. Esta diferencia de temperatura y, en consecuencia, la desigual resistencia de los filamentos, desequilibra el puente, apareciendo una diferencia de potencial entre los puntos **P** y **Q**. Dicha diferencia de potencial es una indicación de la presencia del gas problema.

B. Aplicaciones.

La *principal aplicación* de los transductores basados en la conductividad térmica consiste en la detección de gases puros, ya que cada gas posee una conductividad térmica característica; aunque también podría detectar mezclas de gases. Cuando se emplea para detectar mezclas de gases, se debe considerar la superposición de los efectos de la conductividad térmica de cada gas involucrado.

IV. Transductores basados en el calor de combustión.

El *calor de combustión* de una sustancia se define como la cantidad de calor liberada por unidad de masa cuando ocurre una combustión completa en presencia de oxígeno.

El transductor asociado con el calor de combustión es el *transductor catalítico*. Se utiliza para la detección de gases, puesto que compara el calor de combustión de una muestra de gas con el calor de combustión de un gas de referencia. Esta comparación proporciona información referente a la concentración de gas combustible presente en la muestra; pero no identifica los gases involucrados.

A. Transductor catalítico.

La *unidad fundamental del transductor catalítico* consiste de un filamento de platino revestido con cerámica. La cerámica se trata con un catalizador¹ que favorece la oxidación (combustión con oxígeno). Al filamento se le aplica una diferencia de potencial constante que produzca un calentamiento en él. Cuando el gas combustible entra en contacto con el catalizador, el gas se oxida y eleva aún más la temperatura. Dicho incremento en la temperatura aumenta la resistencia del filamento.

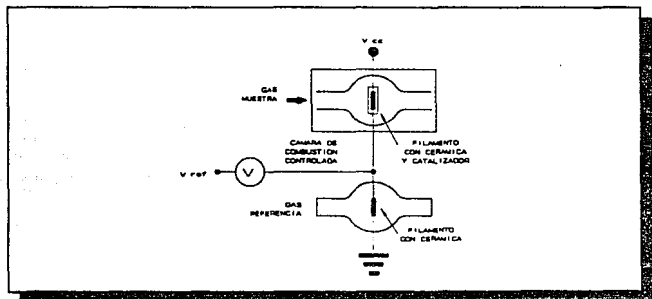


Figura 4.3: Esquema funcional de un *transductor catalítico*.

Por lo tanto, la resistencia del filamento depende de la cantidad de gas combustible presente en la muestra.

¹ **Catalizador:** substancia que acelera una reacción, sin tomar parte en ella.

En el transductor catalítico se utilizan dos filamentos de platino de la misma resistencia revestidos con cerámica. (Véase Figura 4.3). Sin embargo, sólomente la cerámica de uno de ellos se trata con el catalizador y, únicamente, éste se pone en contacto con la muestra de gas. El otro filamento se mantiene aislado de la muestra y se utiliza como referencia. La diferencia en las resistencias de ambos filamentos es una indicación de la concentración de gas combustible en la muestra.

Es necesario que el transductor catalítico cuente con un dispositivo que detenga la llama, a fin de evitar la propagación de la combustión del gas. Dicho dispositivo puede controlar la cantidad de mezcla gaseosa inflamable, o la cantidad de oxígeno que se suministre.

B. Aplicaciones.

La *principal aplicación* de los transductores basados en el calor de combustión, consiste en la determinación de la concentración de gas combustible en una mezcla gaseosa, sin importar cuál fuere éste. Lo anterior es posible gracias a que el calor de combustión por debajo del límite inferior de inflamabilidad es aproximadamente el mismo para la mayor parte de los gases combustibles².

V. Transductores de luz infrarroja.

Cada gas *absorbe* una longitud de onda característica de la región infrarroja del espectro electromagnético.

² Dato experimental proporcionado por la NFPA.

Entiéndase por *absorción* al fenómeno por el cual un cuerpo retiene parte de la energía que incide sobre él.

Los transductores relacionados con la detección de luz infrarroja se conocen genéricamente como *detectores fotoeléctricos*.

A. Detectores fotoeléctricos.³

Un *detector fotoeléctrico* es un elemento que, en función de la luz que incide sobre él, genera un voltaje entre sus terminales (elemento activo), o bien, varía alguna de sus características que afecten su conducción eléctrica como su resistencia o su cantidad de portadores (elemento pasivo).

Los detectores fotoeléctricos por sí mismos no son selectivos al detectar gases. Es por esto que, para detectar selectivamente la presencia u ausencia de un gas específico, se requiere utilizar filtros que seleccionen el ancho de banda de frecuencias de la región infrarroja que absorba dicho gas.

B. Aplicaciones.

La *principal aplicación* de los transductores de luz infrarroja consiste en la detección de gases puros y de compuestos específicos. Esto es posible ya que cada gas absorbe una longitud de onda determinada de la región infrarroja del espectro electromagnético.

También se utilizan para la detección de explosiones, ya que cuando éstas ocurren, emiten luz infrarroja; y para la detección de humo, puesto que impide el paso de luz.

³ En el Capítulo II, Métodos para el Análisis de Gases, los detectores fotoeléctricos se abordan con más detalle.

VI. Transductores basados en la conducción electrofítica.

El fenómeno de la *conducción electrofítica* consiste en el paso de una corriente eléctrica a través de un electrólito. Cabe aclarar que ésta se presenta con efectos químicos secundarios como la *electrólisis*, que consiste en la separación de un compuesto químico en sus elementos que lo conforman.

Un *electrólito* es una solución que puede conducir una corriente eléctrica debido a la presencia de iones de su soluto. El término *ion* se utiliza para designar átomos o moléculas que poseen una carga eléctrica debida a la pérdida (iones cargados positivamente o *cationes*) o adquisición (iones cargados negativamente o *aniones*) de electrones. Por otro lado, una sustancia que se disuelve en otra forma una *solución*; la sustancia que disuelve recibe el nombre de *solvente*, mientras que la sustancia disuelta se denomina *soluto*.

El transductor asociado con la conducción electrofítica es la *celda electrofítica*. Esta se utiliza en la detección de gases puesto que cuantifica la concentración de una mezcla de gases.

A. La celda electrofítica.

Una celda electrofítica consta de dos electrodos inmersos en un electrólito, a los cuales se les aplica externamente una diferencia de potencial. (Véase Figura 4.4). El circuito eléctrico entre los dos electrodos se cierra precisamente a través del electrólito. La intensidad de corriente eléctrica que puede circular por el electrólito, está en función de la diferencia de potencial aplicada y de su resistencia. A su vez, la resistencia del electrólito es inversamente

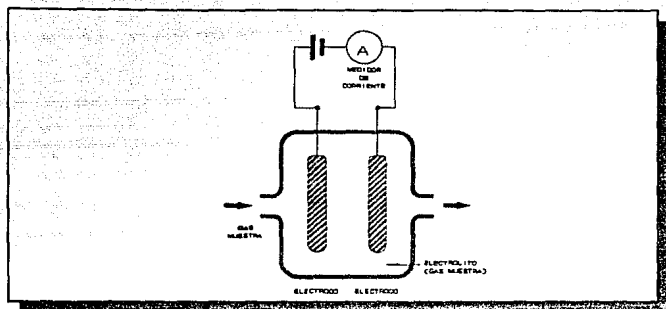


Figura 4.4: Esquema funcional de una celda electrolítica.

proporcional a la cantidad presente de iones del soluto, a saber el gas en cuestión. Finalmente, la intensidad de corriente eléctrica que circula por la celda electrolítica depende de la concentración de la mezcla de gases que se desea detectar.

Un *electrodo* es un elemento conductor ubicado en el extremo de un circuito metálico, el cual suministra o recibe corriente eléctrica de un electrolito. El electrodo que tiene un potencial positivo respecto al otro recibe el nombre de *ánodo*; mientras que el electrodo restante se conoce como *cátodo*.

Los portadores de carga en la conducción electrolítica, provienen de la ionización del soluto en el electrolito. El fenómeno de la *ionización* consiste en la ruptura de todos o algunos enlaces electrónicos de una molécula, con la consecuente formación de aniones y cationes.

El fenómeno de la conducción electrolítica consiste en lo siguiente. La corriente eléctrica que produce la ionización del soluto

en el electrolito se origina en los electrodos. Ahora bien, los cationes del soluto se dirigen al cátodo, en donde reciben electrones de la fuente de voltaje para formar átomos o moléculas neutros. A su vez, los aniones del soluto se dirigen al ánodo, en donde liberan sus electrones a la fuente de voltaje para formar átomos o moléculas neutros. De este modo se cierra el circuito. Los átomos y las moléculas neutros son el producto de la electrólisis. Principalmente se utilizan electrodos de níquel, carbono, acero inoxidable y aleaciones de níquel con hierro.

La forma de onda del voltaje que se aplica es importante. Si se suministra una diferencia de potencial continua, la conductividad electrolítica presenta cuatro etapas. La primera es un pequeño transitorio debido a la capacitancia entre los electrodos. La segunda es un corto periodo en el cual la intensidad de corriente es estrictamente proporcional a la *conductividad* del electrolito (la conductividad es el inverso de la resistencia). La tercera es un largo periodo en el cual la intensidad de corriente disminuye gradualmente debido a la electrólisis en la superficie de los electrodos. Finalmente, la cuarta ocurre cuando la intensidad de corriente alcanza su valor mínimo y se mantiene constante debido a la saturación de los electrodos por parte de los productos de la electrólisis (*polarización del electrodo*⁴).

La intensidad de corriente de interés es la que ocurre durante la segunda etapa. Es por esto que se suministra una diferencia de potencial alterna, con forma de onda cuadrada. Su frecuencia depende de la duración de dicha etapa y el cambio en su polaridad minimiza los productos de la electrólisis.

La relación entre la conductividad y la concentración es lineal a bajas concentraciones; pero a altas concentraciones deja de ser lineal, alcanzando en muchos casos un máximo seguido de una pendiente negativa. Si la celda electrolítica opera en la zona lineal, la intensidad

⁴ No confundir con polarización de un dieléctrico en un capacitor.

de corriente eléctrica que circula por el transductor es proporcional a la concentración del gas que se desea detectar.

B.Aplicaciones.

La *principal aplicación* de una celda electrolítica consiste en la determinación de la concentración de una solución formada por un gas.

Cuando se requiere detectar la concentración de un componente específico en una solución formada por varios elementos, se requiere que se mantengan constantes las concentraciones de los otros elementos, o que la conductividad del componente de interés sea mayor que la de los demás.

VII.Transductores basados en la detección de iones específicos.

En los transductores basados en la detección de iones específicos, se verifica una reacción química entre ellos y el gas que se desea detectar, la cual genera una diferencia de potencial. Dicha diferencia de potencial es la que se utiliza para determinar la ausencia o presencia del gas en cuestión.

El transductor asociado con la detección de iones específicos es la *celda voltaica*.

Las celdas voltaicas se fabrican de materiales específicos, dependiendo del gas problema. Es decir, existe una celda voltaica especial para detectar cada gas.

A. La celda voltaica.

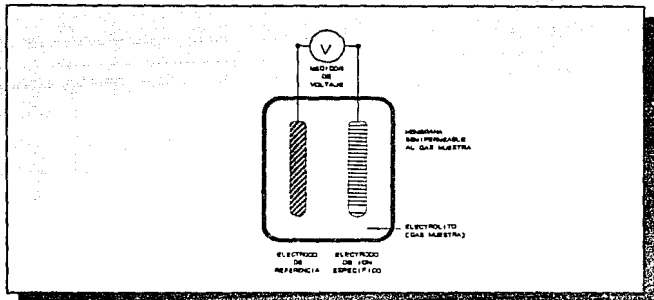


Figura 4.5: Esquema funcional de una celda voltaica.

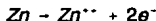
La celda voltaica consiste de dos electrodos que se sumergen en la disolución que se desea caracterizar. (Véase Figura 4.5). El primero es un *electrodo de ion específico*, el cual es sensible a ciertos iones, y el segundo es un *electrodo de referencia*, el cual sirve para cerrar el circuito. Entre ambos electrodos aparecerá una diferencia de potencial cuando estén presentes aquellos iones ante los cuales responde el electrodo de ion específico.

El electrodo de ion específico puede ser *catión específico*, si responde a la actividad de iones cargados positivamente; o *anión específico*, si responde a la actividad de iones cargados negativamente.

Como se había mencionado, el término *ion* designa átomos o moléculas que poseen una carga eléctrica debida a la pérdida (*iones*

positivos o cationes) o adquisición (*iones negativos o aniones*) de electrones; y un *electrodo* es un elemento conductor ubicado en el extremo de un circuito metálico, el cual suministra o recibe corriente eléctrica de un medio conductor diferente.

El principio de operación de un electrodo de ion selectivo se describirá con un electrodo de zinc sumergido en agua por ser muy ilustrativo. Al introducir una placa de zinc en agua pura, inmediatamente emigran iones de zinc a la disolución, teniendo lugar la siguiente reacción:



lo anterior se debe a que las moléculas polares del agua vencen las fuerzas de cohesión que mantienen a los iones de zinc en su red cristalina. Al parecer, habría un flujo continuo de iones hacia la disolución. Sin embargo, puesto que cada ion positivo de zinc que entra en la solución deja libres dos electrones en la placa de zinc, esta última adquiere un potencial negativo con respecto a la solución. Consecuentemente, la placa atrae cierta cantidad de iones de la solución, capturándolos y neutralizándolos. A medida que mayor sea la carga negativa de la placa de zinc, mayor será la tendencia de los iones positivos de zinc a regresar a ella. En el estado de equilibrio, la cantidad de iones que abandonan la placa y la de los que regresan por unidad de tiempo es igual, manteniéndose un potencial constante. El potencial que se genera en la placa se denomina *potencial de electrodo*.

Adicionalmente, se requiere de una *membrana* que sirva como interfaz entre el electrodo de ion específico y la solución, para evitar la entrada de otras sustancias que puedan interferir con el proceso de transferencia electrónica. Dicha membrana, ya sea líquida o sólida, debe ser permeable sólo a la sustancia de interés, o bien, a cierto

producto de alguna reacción química entre la membrana y la sustancia de interés.

B. Aplicación.

La *principal aplicación* de los detectores de iones específicos (celdas voltaicas) consiste en detectar la presencia de un gas específico, ya que responden solamente a la actividad de los iones libres de dicho gas, independientemente de su concentración. Lo anterior se debe a que la relación entre la cantidad de partículas ionizadas y neutras (las que no están ionizadas) no siempre es lineal respecto a la concentración. Sin embargo, para soluciones diluidas, la concentración es proporcional a la actividad iónica.

VIII. Detectores basados en la adsorción. Principio de funcionamiento.

El fenómeno de *adsorción* consiste en la adherencia de un gas o vapor a la superficie de un sólido.

El transductor asociado con la adsorción consiste de un semiconductor (una película delgada de algún óxido metálico), que exhibe una variación en su conductividad eléctrica al adsorber un gas específico. La cantidad de gas que se adsorbe es proporcional a su concentración en la atmósfera. De este modo, la conductividad eléctrica del semiconductor es proporcional a la concentración del gas.

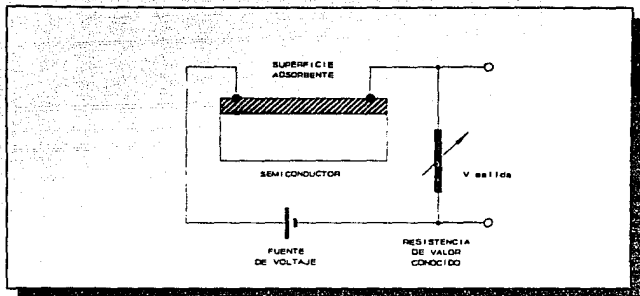


Figura 4.6: Esquema funcional de un transductor asociado al fenómeno de adsorción.

La adsorción del gas en el semiconductor puede producir tres efectos:

1. *Adsorción débil*, sin consecuencias en la conductividad eléctrica.
2. *Adsorción fuerte donadora*.
3. *Adsorción fuerte aceptora*.

La adsorción fuerte, donadora o aceptora, inmoviliza los portadores tipo *p* y tipo *n* respectivamente, alterando la conductividad del semiconductor.

El circuito de medición consiste del sensor conectado en serie con una resistencia de valor conocido, a los cuales se les aplica una diferencia de potencial constante. La corriente que fluye a través del circuito es proporcional a la conductividad eléctrica del transductor. Como se indicó

anteriormente, la conductividad eléctrica del transductor está en función de la concentración del gas (Véase Figura 4.6).

Por lo tanto, la diferencia de potencial presente en la resistencia de valor conocido es proporcional a la concentración de un gas específico en la atmósfera.

CAPÍTULO V.

SISTEMA DE ADQUISICIÓN DE DATOS.

I. Introducción.

Un *sistema de adquisición de datos (SAD)* es un conjunto de dispositivos cuya finalidad es la recolección, procesamiento y almacenamiento de una o varias señales para obtener uno o varios resultados, que pueden ser utilizados en procesos subsiguientes.

Una *señal* es el modelo del comportamiento de una variable (variable dependiente) que depende otra variable (variable independiente). Dicha señal contiene información y, generalmente, se asocia a la variación de una magnitud física con respecto al tiempo.

En el caso de un sistema electrónico de adquisición de datos, las señales se pueden obtener de dos formas: primero, por medición directa de parámetros eléctricos; por ejemplo: voltaje, corriente, resistencia y frecuencia; segundo, por medición indirecta de parámetros no eléctricas por medio de un transductor.

Ahora bien, un *transductor* es un dispositivo que alimentado por la energía de un sistema, suministra proporcionalmente energía de la misma o de otra naturaleza a un segundo sistema. Esto es, transforma una variable física a otra de igual o diferente naturaleza.

II. Sistema digital de adquisición de datos.

Un *sistema digital de adquisición de datos* captura una o varias señales analógicas, las convierte en digitales, y posteriormente las codifica para almacenarlas y procesarlas mediante una unidad digital de proceso. Es decir, un sistema digital de adquisición de datos lleva al cabo el procesamiento y almacenamiento de la información en forma digital.

Una *señal digital* es aquella que se discretiza en tiempo y se cuantiza en magnitud. La *discretización* significa que la señal solamente se considera en ciertos instantes de tiempo; mientras que la *cuantización*, que la señal solamente puede tomar determinadas magnitudes (Véase Figura 5.1).

El propósito de la *codificación* consiste en generar una señal digital binaria que sea representativa de la forma de onda de la señal analógica que se muestrea.

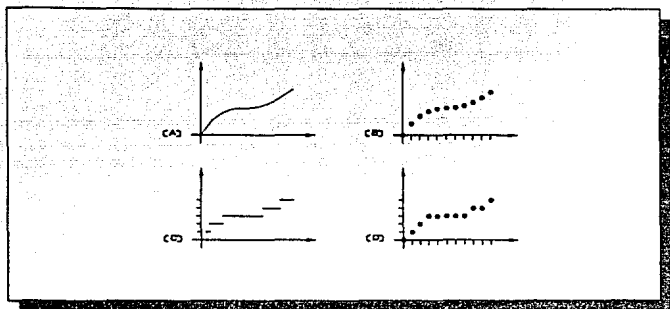


Figura 5.1: Tipos de señales: (A) analógica; (B) discreta; (C) cuantizada; (D) digital.

El sistema digital de adquisición de datos obtiene una señal digital binaria a partir de la señal analógica mediante las siguientes operaciones:

- Muestra la señal y retiene el nivel muestreado. Esta operación la realiza el circuito de muestreo y retención.
- Cuantiza y codifica el nivel anteriormente muestreado. Esta operación la realiza el convertidor analógico digital.

A continuación se explican los elementos que constituyen un sistema digital de adquisición de datos.

A.Elementos de un sistema digital de adquisición de datos.

Los elementos que conforman un sistema de adquisición de datos son los siguientes (véase Figura 5.2):

1.Transductor eléctrico y circuitería de soporte.

El *transductor eléctrico* se encarga de transformar variables físicas a variables eléctricas aceptables para el sistema. La *circuitería de soporte* controla el transductor para su correcta operación; además incluye dispositivos para su calibración y le proporciona potencia de excitación.

2.Circuito acondicionador de señal del transductor.

Proporciona a la señal que proviene del transductor niveles de voltaje y de potencia adecuados para la entrada al multiplexor analógico.

3.Filtro acondicionador de señal del transductor.

Proporciona a la señal que proviene del transductor un ancho de banda adecuado para el sistema.

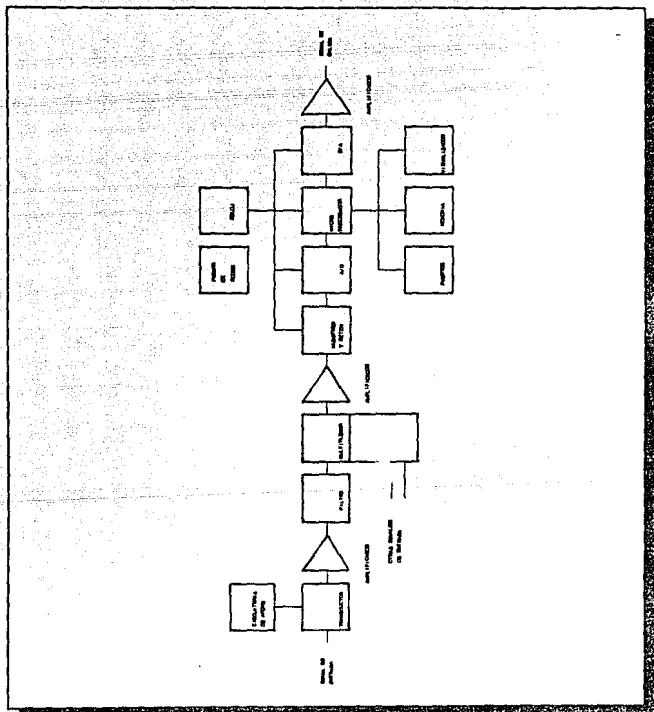


Figura 5.2: Diagrama a bloques de un sistema digital de adquisición de datos.

4. Canal analógico multiplexado y líneas de control.

Permite la selección de una señal de entre varias provenientes de múltiples transductores, para introducirla al sistema.

5. Circuito acondicionador de señal del multiplexor.

Proporciona a la señal que proviene del multiplexor analógico niveles de voltaje y de potencia adecuados para los circuitos de muestreo-retención y de conversión analógica-digital.

6. Circuito para muestreo y retención.¹

Este circuito analógico captura y retiene el nivel de una señal analógica para que sea alimentada a un convertidor analógico-digital.

7. Sistema de conversión analógica-digital (A/D).

Proporciona la representación digital en forma de pulsos de voltaje de una señal analógica.

¹ Circuito "Sample and Hold".

8.Unidad de procesamiento.

Se encarga de buscar, decodificar y ejecutar las instrucciones de un *programa de control*, de generar las señales de control y temporización para que los dispositivos que conforman al sistema de adquisición de datos operen adecuadamente; de procesar los datos adquiridos por el sistema; y de controlar el flujo de datos entre *puertos de entrada/salida y memoria*.

a.Programa de control.

Es la secuencia de instrucciones que dirigen las acciones del procesador para que realice su función.

b.Memoria.

Dispositivo en el cual se puede introducir, mantener y recuperar datos; éstos pueden ser datos propiamente dichos, o bien, el programa de control.

c.Puerto de entrada/salida.

Dispositivo por el cual se puede recibir o transmitir datos desde o hacia otros dispositivos externos.

9.Sistema de conversión digital-analógica (D/A).

Se utiliza para transformar datos digitales del sistema a su forma analógica equivalente, ya sea para obtener una representación aproximada de la señal analógica original, o bien, para controlar dispositivos analógicos externos.

10.Circuito acondicionador de señal del convertidor digital-analógico.

Proporciona a la señal que proviene del convertidor digital-analógico niveles de voltaje y de potencia adecuados para un sistema analógico externo.

III.Características de un sistema digital de adquisición de datos.

Los datos de la señal digital codificados se representan mediante una secuencia de variaciones entre los dos niveles permitidos de la señal digital binaria, propios de la electrónica digital. Este sistema se caracteriza por poseer un ancho de banda pequeño, pero mucha precisión. En consecuencia los sistemas digitales presentan las siguientes características:

A.Inmunidad a ruido.

La *inmunidad al ruido* es la capacidad de un sistema a no perder información por la presencia de señales externas ajenas al sistema.

Los sistemas digitales son más inmunes al ruido que los sistemas analógicos. Esto se debe a que, en todo momento, se conocen los posibles valores que una señal digital binaria puede tomar; mientras que no es posible conocer de antemano el valor preciso que una señal analógica puede tomar. En consecuencia, es más probable estimar correctamente los valores reales alterados por ruido en una señal digital binaria que en una señal analógica.

B. Detección y corrección de errores.

La *detección de errores* es la capacidad de un sistema de reconocer la alteración en un conjunto de datos; mientras que la *corrección de errores* es la capacidad de un sistema de regenerar a su forma original un conjunto de datos alterados.

Existen algoritmos para detectar y corregir errores dentro de una señal digital binaria.

C. Comunicación.

La *comunicación* se refiere a la capacidad de un sistema para intercambiar datos con otro sistema.

Un sistema digital puede comunicarse a unidades de proceso digitales externas para procesar con mayor detalle los datos adquiridos. Esto es muy importante debido a la poca capacidad de procesamiento de un sistema de adquisición de datos.

CAPÍTULO VI.

CONSIDERACIONES DE DISEÑO.

I. Introducción.

Este capítulo presenta el proceso de diseño, junto con sus principios, factores técnicos y normas que lo auxilian. Además muestra la técnica de diseño funcional descendente, así como una técnica para seleccionar una familia de microprocesadores. También se hace mención de las limitantes del diseño. Finalmente, se hace el planteamiento de las especificaciones y la partición del proyecto de diseño en módulos funcionales, tanto del "hardware" como del "software".

II.El proceso del diseño.

El *diseño* es un proceso creativo de desarrollo o invención de algo que anteriormente no existía. El diseño es un proceso en el cual se aplica intensivamente el criterio en forma sistemática. El diseño es un proceso razonado que resuelve un problema específico mediante una solución novedosa.

Dentro del proceso de diseño se consideran principios, factores, técnicas y normas a partir de los cuales se justifica la realización de un diseño específico.

A.Principios del diseño.

Los *principios* de diseño son las recomendaciones básicas con las cuales se debe normar el criterio cuando se diseña. Estos principios de diseño son cuatro: modularidad, regularidad, localidad y conectividad.

1.Modularidad.

La *modularidad* se refiere a la partición del sistema en secciones operativas o módulos.

2.Regularidad.

La *regularidad* se refiere a que cada sección operativa o módulo dentro del sistema tenga una estructura semejante.

3.Localidad.

La *localidad* se considera en dos diferentes niveles: a nivel sistema y a nivel componente.

a.Localidad a nivel sistema.

La *localidad a nivel sistema* considera el lugar y el medio ambiente en donde opera el sistema.

b.Localidad a nivel componente.

La *localidad a nivel componente* considera la posición relativa de cada sección operativa o módulo dentro del sistema. Esto ayuda a definir la arquitectura del sistema.

4.Conectividad.

La *conectividad* se refiere a que las secciones operativas o módulos dentro del sistema se comuniquen adecuadamente entre sí.

B.Factores del diseño.

Los *factores* de diseño son aquellos aspectos que se deben cumplir cuando se diseña. Los principales factores que afectan el diseño con microprocesadores son: precisión, rango, resolución, medio ambiente, ergonomía, condiciones de falla y mantenimiento.

1.Precisión.

La *precisión* es la repetición en la operación del sistema. Un sistema que responde de manera similar ante un mismo estímulo es *confiable*. La precisión de un sistema depende de los algoritmos que se utilizan. Un *algoritmo* es un conjunto finito de acciones, sin ambigüedad, que resuelven correctamente un problema.

2.Rango.

El *rango* de las variables que intervienen en el sistema es la escala o intervalo de valores dentro del cual pueden fluctuar.

3.Resolución.

La *resolución* de las variables involucradas en el sistema es la fineza con la cual fluctúan los valores que pueden tomar. La resolución tiene relación directa con la longitud de palabra del microprocesador.

4.Medio ambiente.

El *medio ambiente* es el conjunto de condiciones externas que afectan el funcionamiento del sistema. Principalmente son la temperatura, la humedad y el ruido.

5.Ergonomía.

La *ergonomía* se refiere al conjunto de condiciones antropométricas que se deben tomar en cuenta para que la utilización del sistema sea cómoda.

6. Condiciones de falla.

Las *condiciones de falla* son la manera en que responde el sistema ante eventos imprevistos, como podrían ser la falta de suministro eléctrico o, los errores ocasionados por ruido. El sistema debe ser capaz de recuperarse por sí mismo.

7. Mantenimiento.

El *mantenimiento* es de dos tipos: mantenimiento correctivo y mantenimiento preventivo.

a. Mantenimiento correctivo.

El *mantenimiento correctivo* es el conjunto de acciones que se ejecutan para corregir una falla.

b. Mantenimiento preventivo.

El *mantenimiento preventivo* es el conjunto de acciones que se ejecutan para prevenir fallas.

El sistema debe contar con puntos de prueba y calibración accesibles, así como con rutinas que verifiquen su buen funcionamiento.

C. Técnica de diseño funcional descendente ("*Top-Down*").

Se entiende por *técnica* al conjunto de procedimientos que se ejecutan en cierto orden para lograr un fin específico.

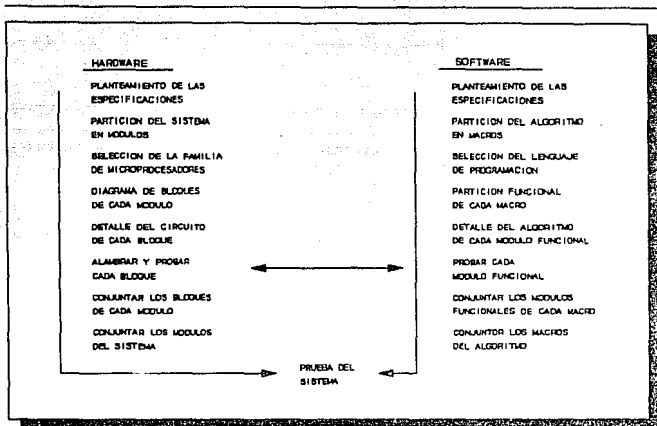


Figura 6.1: Técnica de diseño funcional descendente ("Top-Down").

La *técnica de diseño funcional descendente* es de propósito general y consiste en un conjunto de procedimientos para diseñar sistemas de aplicación específica, abordando el problema de lo general a lo particular. Es decir, la técnica de diseño funcional descendente separa un problema complejo en varios "subproblemas" con menor grado de complejidad, auxiliándose en los principios, factores y normas del diseño. (Véase Figura 6.1).

D. Normas comerciales.

Las *normas comerciales* marcan los lineamientos que se deben seguir para que un sistema sea compatible con otros sistemas del

mundo exterior. También, ciertas normas comerciales enuncian aspectos de seguridad que el sistema debe cumplir durante su operación.

III. Selección de una familia de microprocesadores.

La *selección de una familia de microprocesadores* incluye consideraciones sobre su utilización y una técnica para su selección.

A. Microprocesadores contra lógica combinacional y secuencial.

No siempre es necesario, y a veces hasta resulta inadecuado, la utilización de un microprocesador para implementar el diseño de un sistema digital. Para elegir entre la utilización de lógica combinacional y secuencial, o la utilización de microprocesadores dentro del diseño, se consideran los siguientes puntos:

1. Lógica combinacional y secuencial.

Se prefiere utilizar *lógica combinacional y secuencial* cuando:

- el sistema no es muy complejo.
- la velocidad de un microprocesador no es la suficiente.

2. Microprocesadores.

Se prefiere utilizar un *microprocesador* cuando:

- el sistema realiza un gran número de funciones distintas.
- el sistema maneja un gran número de entradas y/o salidas gestionadas en forma diferente.
- el sistema almacena en memoria un gran número de datos y/o estados.
- el sistema toma decisiones en función de su "historia" estadística (datos y/o estados almacenados).
- el sistema realiza numerosas operaciones lógicas y/o aritméticas.
- existe la necesidad de modificar la programación del sistema o sus características operativas externamente por medio de "software".
- el sistema debe ser flexible para adaptarse a distintos equipos.
- se requiere que el ciclo de desarrollo del sistema sea breve.

B. Técnica de selección de una familia de microprocesadores.

Se entiende por *técnica de selección* de una familia de microprocesadores la secuencia de pasos a seguir para elegir la familia de microprocesadores que será utilizada durante el diseño.

¹software: programación de un sistema.

Esta técnica de selección consiste en:

- Analizar detenidamente la aplicación que se pretende desarrollar, identificando claramente cuáles son las variables de interés.
- Determinar la resolución y la velocidad con la que se requiere procesar la información. Lo primero determina la utilización de microprocesadores de 4, 8, 16 o 32 bits; y lo segundo, el rango de frecuencias del reloj maestro del sistema.
- Determinar el tipo de interfaces que se necesita utilizar para seleccionar el tipo adecuado de coprocesadores (controladores y puertos de entrada y/o salida).
- Determinar el tipo de memorias (primaria, secundaria, caché), que se requiere en el sistema y su capacidad.
- Definir el grado de *paralelismo* a ser utilizado en el *esquema de procesamiento*.
- Definir la *arquitectura* a ser implantada.
- Seleccionar la familia de microprocesadores que se empleará apoyándose en la consulta de manuales.

C. Criterios adicionales para seleccionar una familia de microprocesadores.

Además de los puntos anteriores, es conveniente considerar otros criterios para seleccionar una familia de microprocesadores, como:

- 1. Disponibilidad:** que se encuentren disponibles comercialmente todos los elementos de "*hardware*" y "*software*" de la familia de microprocesadores.

- 2. Número de fuentes de poder:** que utilice una sola fuente de poder. Actualmente esto cobra una importancia relevante, ya que los dispositivos de las fuentes de poder frecuentemente son más caros que los dispositivos de la familia de microprocesadores.

- 3. Compatibilidad entre la familia de microprocesadores:** que los diversos dispositivos de la familia de microprocesadores puedan comunicarse entre sí directamente, o bien, con el mínimo empleo de lógica externa.

- 4. Soporte de "*software*":** que existan ensambladores y simuladores propios de la familia de microprocesadores; adicionalmente si se puede utilizar lenguajes de alto nivel y si existen los compiladores respectivos. La existencia de estos productos facilita el desarrollo de un proyecto.

- 5. Existencia de proveedores alternativos:** que dos o más proveedores comercialicen ampliamente la familia de microprocesadores. Esto garantiza su disponibilidad en el mercado.

- 6. Nuevos microprocesadores compatibles en "*software*" y "*hardware*":** que en un futuro haya la posibilidad de utilizar microprocesadores de la misma familia, más veloces y con un conjunto de instrucciones ampliado.

7. Empresas consultoras: que exista soporte técnico al menos durante las primeras etapas en el desarrollo del proyecto.

8. Equipos de desarrollo: que existan evaluadores en tiempo real para sistemas propios de la familia de microprocesadores.

IV. Limitaciones de diseño.

Las *limitaciones* son circunstancias que restringen el proceso de diseño.

Existen dos grandes tipos de limitaciones dentro del diseño en Ingeniería: las limitaciones de tipo tecnológico y las limitaciones de tipo económico.

Las *limitaciones tecnológicas* son aquellas circunstancias que restringen el proceso de diseño en cuanto a métodos, procedimientos y dispositivos existentes.

Las *limitaciones económicas* son aquellas circunstancias que restringen el proceso de diseño en cuanto a los recursos con que realmente se cuentan.

Además, existen otras limitaciones, como aquellas que restringen la creatividad en el proceso de diseño.

V. Planteamiento de especificaciones.

Como el planteamiento de las especificaciones del diseño,

se presenta lo siguiente:

La finalidad del presente diseño es realizar un sistema para sensar gases explosivos, que sea capaz de adquirir y almacenar datos en forma digital, y de comunicarse con una computadora personal para un análisis posterior de dichos datos.

Para lograr esto, se utiliza un elemento sensor catalítico y circuitería electrónica basada en un microprocesador para vigilar continuamente el ambiente, y detectar la presencia de gases o vapores combustibles.

Se plantea el uso de un sensor catalítico, ya que éste mide la cantidad de gas combustible presente en la atmósfera. La medición se presenta como un porcentaje del límite inferior de explosividad ("*Low Explosive Limit*", *LEL*) del gas combustible, el cual se refiere a la cantidad más pequeña de gas que soportaría por sí misma la propagación de su flama cuando se mezcla con aire (oxígeno) y se enciende.

A partir de la definición del problema, el sistema se conforma de los siguientes módulos operativos (Véase Figura 6.2):

- Sensado de gases combustibles.
- Almacenamiento de datos adquiridos.
- Comunicación con computadoras personales.

Este diseño se puede separar en dos grandes rubros: el "*hardware*" y el "*software*".

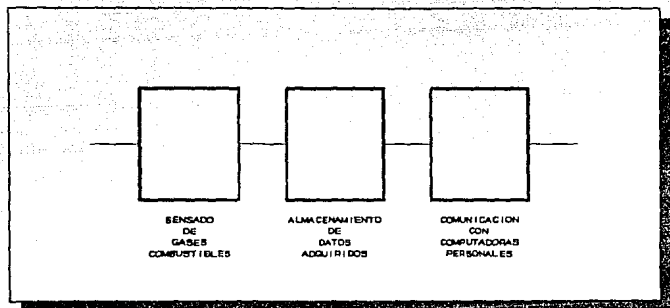


Figura 6.2: Diagrama de bloques del Sistema para Detectar Gases Combustibles.

VI. Diseño del "hardware".

El "hardware" se refiere al conjunto de elementos físicos (dispositivos) que constituyen a un sistema.

A. Partición del sistema en módulos.

El "hardware" del sistema se particiona en los siguientes módulos (Véase Figura 6.3):

- 1) Sensado de gases combustibles.
 - a) Transductor catalítico.
 - b) Conversión analógica-digital.
- 2) Almacenamiento de datos adquiridos.
 - a) Unidad de procesamiento.

- b) Unidad de almacenamiento.
- 3) Comunicación.
 - a) Comunicación con Computadoras Personales.
 - b) Comunicación con el usuario.

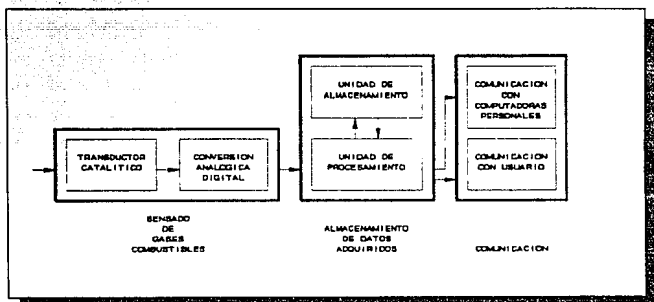


Figura 6.3: Diagrama de bloques del "hardware".

Este diagrama de bloques se apoya en el que se presentó en el capítulo V para un sistema de adquisición de datos.

B. Selección de la familia de microprocesadores.

Para implementar este diseño, se considera utilizar un microprocesador, ya que el sistema realiza diferentes funciones, maneja diversas entradas y salidas, almacena en memoria un gran número de datos, y realiza operaciones aritméticas y lógicas. Es decir, un microprocesador ofrece un fácil manejo de datos digitales y

,además, la posibilidad de cambiar su operación si se altera su programación.

Otro aspecto que se considera es la velocidad de un microprocesador, la cual es suficiente para el manejo de los datos adquiridos y, además, permite la administración de los dispositivos electrónicos en que se apoya.

También, una familia de microprocesadores cuenta con dispositivos compatibles especializados que pueden realizar diferentes funciones; tales como comunicación serial y comunicación en paralelo. Lo anterior facilita la comunicación con computadoras personales. Por otro lado, se considera utilizar memoria primaria para almacenar la programación del sistema, así como los datos adquiridos.

Debido a que se utiliza un microprocesador, es necesario definir un *esquema de procesamiento*, el cual se refiere a la manera de utilizar una arquitectura. Para este diseño se considera el *procesamiento a bloques*. Dicho esquema consiste en el procesamiento secuencial de las tareas del sistema, sin traslapar ninguna ejecución. Esto es, inicia con la primera tarea y termina con la última, y solamente cuando termina una tarea puede iniciar con la siguiente.

La *arquitectura* de un sistema se refiere a la disposición de los elementos que lo constituyen. Para este diseño se considera una arquitectura tipo *SISD* (*Single Instruction, Single Data*), en la cual un sólo procesador atiende un único proceso. (Véase Figura 6.4).

Se eligió el microprocesador MC68008 de Motorola, ya que éste ofrece las características del microprocesador MC68000; como son: un número elevado de registros internos, manejo de varios tipos de datos con diversas longitudes, soporte de los principales tipos de direccionamiento, incorporación de operaciones aritméticas como multiplicación y división, instrucciones potentes para el manejo individual de bits, varios registros que pueden actuar como punteros de pila (*stack pointer*), e instrucciones de control para ejecutar

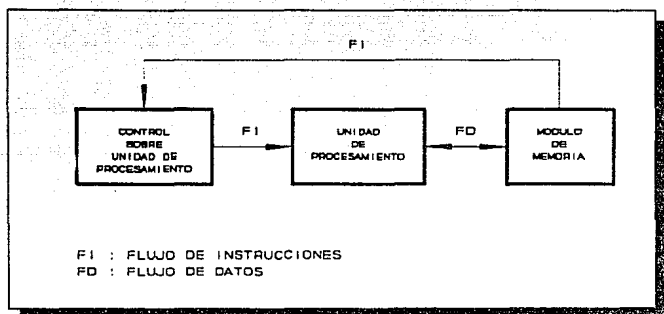


Figura 6.4: Diagrama de bloques de una arquitectura tipo SISD.

iteraciones; sin embargo solo utiliza un bus de datos de 8 bits. Esto último permite un fácil acceso a memorias y periféricos que poseen un bus de datos de 8 bits, simplificando conexiones.

Además, es importante considerar el conocimiento que se tiene sobre el microprocesador y su conjunto de instrucciones, así como la información disponible referente al tema: libros y manuales.

C. Selección de dispositivos para el sensado de gases combustibles.

Para el módulo de sensado de gases combustibles se considera seleccionar un transductor catalítico y un convertidor analógico-digital.

1. Transductor DET-TRONICS C7061C.

La elección de un transductor adecuado es una de las decisiones más críticas en el diseño de un sistema para la detección de gases combustibles. Lo anterior se debe a que, para la adecuada operación del sistema, se requiere de un sensor cuyo desempeño no se altere aún bajo las condiciones ambientales de operación más adversas.



Figura 6.5: Transductor Det-Tronics C7061C.

Es deseable utilizar un transductor que detecte cualquier gas combustible. Debido a esto último, se elige el transductor Det-Tronics C7061C (véase Figura 6.5), ya que puede detectar cualquier gas combustible bajo una misma configuración; aunque se especializa en la detección de gas metano. Esto simplifica considerablemente su instalación e incrementa la versatilidad del sistema.

El transductor Det-Tronics C7061C utiliza un elemento sensor catalítico para detectar la presencia de gases y vapores

combustibles en la atmósfera². El elemento sensor está montado dentro de una cubierta metálica a prueba de explosiones³ que evita la propagación del fuego, y que lo aísla térmicamente del medio ambiente.

Este transductor consiste esencialmente de dos resistencias del mismo valor. Una de ellas se recubre de un elemento catalítico para sensar gases y vapores combustibles. La otra es un elemento de referencia que compensa las condiciones ambientales. Este par de dispositivos opera a una temperatura *elevada*, mientras se expone al medio ambiente atmosférico. Los alambres resistivos están protegidos por una capa especial de óxido de aluminio (*alúmina*) de alta densidad. Lo anterior incrementa su capacidad para soportar esfuerzos mecánicos; evita su desintegración debida a la combustión que sobre éstos se efectúa; aproxima su respuesta a la linealidad; provoca una mayor estabilidad en su señal de salida; hace uniforme sus características; y prolonga su tiempo de vida. Adicionalmente, la forma cilíndrica del transductor Det-Tronics C7061C permite un calentamiento uniforme en él, y proporciona un corto tiempo de respuesta al contacto con aire en reposo. Finalmente, el pequeño tamaño este transductor hace que éste sea menos susceptible a daños mecánicos y a los efectos de la vibración prolongada.

El transductor Det-Tronics C7061C opera dentro del rango desde 0 % hasta 99 % del Límite Inferior de Explosividad de cualquier gas, con una linealidad en su respuesta de ± 5 %. Tiene un tiempo de respuesta menor a 20 segundos para alcanzar el 90 % de su escala completa si se expone a metano puro en reposo, y se recupera en un tiempo menor a los 30 segundos. Si se expone a cualquier gas combustible, el tiempo

²Refiérase al Capítulo IV, Transductores basados en el calor de combustión.

³Norma NEMA 7.

de respuesta es de 10 segundos para alcanzar el 50 % de su escala completa y de 30 segundos para el 90 %.

También, dicho transductor opera a cualquier temperatura dentro del rango de $-40\text{ }^{\circ}\text{C}$ a $75\text{ }^{\circ}\text{C}$, y tiene un período promedio de vida de 2 a 3 años.

2. Convertidor analógico digital ADC0804.

Un convertidor analógico digital es un dispositivo que recibe una señal de entrada, V_{ent} , y la transforma en una palabra digital con una resolución dada (cantidad de bits de la palabra), mediante una comparación con un voltaje de referencia, V_{ref} .

El circuito integrado ADC0804 es un convertidor analógico digital con una palabra de 8 bits, de aproximaciones sucesivas, fabricado con tecnología CMOS. Este reúne ambas características de rapidez y bajo costo.

Un convertidor analógico digital de aproximaciones sucesivas optimiza el tiempo de conversión, ejecutando un algoritmo de aproximación. Utiliza un comparador, un convertidor digital analógico y un registro de aproximaciones sucesivas. El algoritmo consiste en la aproximación de cada bit de la salida empezando por el bit más significativo ('MSB, "*Most Significant Bit*"), que corresponde a la mitad de la máxima excursión de voltaje: $\frac{1}{2}V_{\text{ref}}$. Este bit valdrá "1" ó "0" si el nivel de la señal de entrada está por encima o por debajo de este valor, respectivamente. En seguida se realiza la comparación del nivel de la señal de entrada con un voltaje que vale $\delta_1 \frac{1}{2}V_{\text{ref}} + \frac{1}{4}V_{\text{ref}}$, siendo δ_1 el valor binario obtenido en el primer bit calculado. Se designará a δ_2 el valor de "1" ó "0" con el mismo criterio anterior ($\delta_1 \frac{1}{2}V_{\text{ref}} + \delta_2 \frac{1}{4}V_{\text{ref}}$). Se sigue realizando el algoritmo de aproximación hasta llegar al bit menos significativo (LSB, "*Less Significant Bit*"). El número de iteraciones es igual al número de

bits de la palabra del convertidor. Este método es conceptualmente muy sencillo y permite realizar conversiones con gran rapidez, esto es, de unos pocos ciclos de reloj de un microprocesador. La implementación de este algoritmo en un circuito integrado no resulta muy costosa.

El convertidor analógico digital ADC0804 posee una resolución de 8 bits con un tiempo de conversión de 100 μ s; sus conversiones las realiza con un error total de $\pm 1/2$ LSB, $\pm 1/4$ LSB y ± 1 LSB. En general, su conexión con casi cualquier microprocesador es sencilla, y tiene un tiempo de acceso de 135 ns. Maneja entradas diferenciales de voltaje analógico que operan dentro del rango de 0 V a 5 V; requiere de una sola fuente de voltaje de 5 V; trabaja con un voltaje de referencia de 2.5 V; y no necesita circuitería externa para ajuste a cero. Los niveles lógicos de sus líneas de entrada y salida son compatibles con los niveles especificados para CMOS y TTL. Su frecuencia nominal de operación es de 640 kHz, y la circuitería adicional para el reloj que se requiere es mínima: solamente un capacitor y una resistencia.

D. Selección de dispositivos y estándares para comunicación.

Para el módulo de comunicación con computadoras personales se considera seleccionar el estándar de comunicación RS-232-C. Por otro lado, para la comunicación con el usuario, se considera efectuar mediante una pantalla digital de cristal líquido.

1. Estándar de comunicación RS-232-C.

La transferencia de información entre dos sistemas digitales se realiza generalmente carácter por carácter, utilizando códigos binarios; por ejemplo: ASCII, EBCDIC o BAUDOT. Cada carácter se suele formar de 5 a 8 bits. En el estándar de comunicación RS-232-C, la transmisión de los caracteres se realiza en forma serial.

La *transmisión serial* consiste en la transmisión secuencial en el tiempo de todos los bits que conforman un carácter, uno tras otro, a través de una sola línea de datos.

Quando se transmite serialmente datos a través de una sola línea, es indispensable utilizar una codificación que permita resolver el problema de sincronización. La codificación debe ser tal que permita al sistema reconocer la agrupación por caracteres de los datos transmitidos. En general, el número de bits por carácter se fija a 8.

Para obtener la sincronización en la comunicación de cada carácter se puede utilizar un *esquema de comunicación asíncrono*. En este esquema, cada carácter que se transmite va señalizado mediante dos bits: uno al principio llamado bit de inicio, y otro al final llamado bit de parada. Estos bits permiten reconocer las fronteras de los caracteres y los utiliza el receptor para sincronizar su reloj con el reloj del transmisor durante la comunicación de cada carácter.

La *norma RS 404 de EIA* ("Electronic Industries Association") define las reglas del método asíncrono de transmisión en serie. Estas reglas son:

- a) Cuando no se envían datos por la línea, ésta se mantiene en estado "1".

- b) Cuando se desea transmitir un carácter, primero se envía un bit de inicio que pone la línea a "0" durante el período de un bit.
- c) A continuación se envían todos los bits que conforman el carácter que se va a transmitir, temporizados por el período del reloj del transmisor.
- d) Finalmente, a continuación del último bit del carácter, se envía el bit de final que ocasiona que la línea se ponga a "1" por lo menos durante el período de un bit.

Los datos codificados según estas reglas pueden ser detectados fácilmente por el receptor si se siguen los siguientes pasos:

- a) Esperar una transición de "1" a "0" de la señal recibida.
- b) Activar un reloj con una frecuencia igual a la del reloj del transmisor.
- c) Muestrear la señal recibida a la frecuencia de dicho reloj para formar un carácter.
- d) Leer un bit más de la línea y comprobar si es un "1" para confirmar que no ha habido error de sincronización. El bit de final sirve también para dar tiempo al receptor de aceptar el dato recibido.

El esquema asíncrono de comunicación presenta las siguientes ventajas: permite enviar caracteres a distintas velocidades, ya que en cada carácter se incorpora la información para su sincronización; existen circuitos integrados de bajo costo que simplifican enormemente la realización de sistemas de entrada/salida con este esquema de comunicación; y, es un esquema de comunicación estándar.

La principal desventaja es su ineficiencia, ya que por cada carácter que se envía, se transmiten dos bits adicionales que no contienen información.

Los sistemas de comunicaciones en serie disponen de un conjunto de recomendaciones elaboradas por asociaciones e institutos de normalización (ISO, EIA, CCITT), que especifican con precisión todas las características del sistema de comunicación. Una de las normas más aceptadas es la *EIA RS-232-C* (o *CCITT V.24*), que define las características funcionales, eléctricas y mecánicas de la interfaz entre una terminal y un equipo de comunicaciones ("*modem*"). La norma RS-232-C también puede aplicarse a la conexión entre dos terminales, sin utilizar "*modems*".

Para la comunicación entre dos terminales se utiliza un

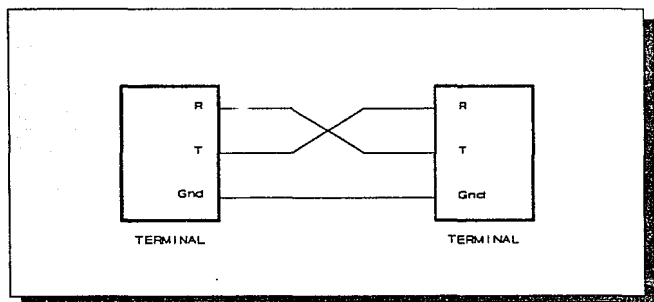


Figura 6.6: Conexión RS-232-C simplificada entre terminales.

subconjunto de 3 señales solamente. Véase Figura 6.4.

En cuanto a las especificaciones mecánicas, la norma RS-232-C establece un conector de 25 terminales, fijando sus dimensiones y la distribución de cada señal en las terminales del conector.

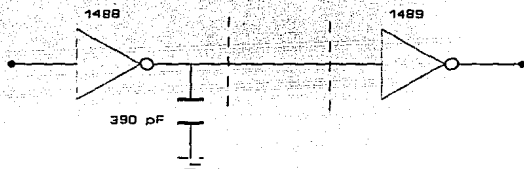


Figura 6.7: Circuito para una línea de transmisión con estándar RS-232-C.

Terminal 2:	Transmisión de datos.
Terminal 3:	Recepción de datos.
Terminal 7:	Tierra.

En cuanto a las especificaciones eléctricas, la norma RS-232-C utiliza emisores y receptores no diferenciales (véase Figura 6.5). La señal "1" se representa con un voltaje ≤ -3 V; y la señal "0", con un voltaje $\geq +3$ V. En la práctica, se utiliza normalmente una señal de +12 V y -12 V, ya que la norma permite un rango de ± 3 V a ± 25 V. La velocidad de subida de la señal se limita a 30 V/ μ s. La interfaz está especificada para una velocidad máxima de transmisión de 20 000 bits/s a una distancia máxima de 15 metros.

2. Pantalla digital de cristal líquido AND 671.

La pantalla de cristal líquido AND 671 es un módulo digital compacto que está integrado por un panel visual conformado por una matriz de puntos, un controlador interno y la circuitería de soporte necesaria para su operación. Este dispositivo puede mostrar 160 tipos de letras, números, símbolos y el alfabeto japonés "Katakana", así como hasta 8 caracteres definidos por el usuario.

La pantalla de cristal líquido AND 671 es compacta, muestra caracteres claros y definidos, y posee control para el contraste. La matriz de puntos que conforma cada carácter es de 5 x 7; y, además, puede mostrar un cursor. Se maneja por medio de 11 comandos de control y puede comunicarse directamente con microprocesadores de 4 u 8 bits. Tiene integrado un controlador tecnología LSI, memoria RAM para que el usuario defina sus propios caracteres, y memoria ROM donde almacena sus caracteres nativos. Utiliza solamente una fuente de poder de +5 V, consume poca potencia (5 mW), y opera dentro de un rango de temperaturas de 0 °C a 50 °C.

Como un dispositivo opcional, se le puede integrar a la pantalla de cristal líquido AND 671, una fuente luminosa para que la pantalla tenga fondo luminosos. De este modo, los caracteres que se muestran pueden ser observados aún en la oscuridad.

VII. Diseño del "software".

El "software" se refiere al conjunto de elementos lógicos (programación) que constituyen a un sistema.

A.Partición del algoritmo en macros.

El "software" del sistema se conforma de los siguientes macros:

- 1) Sensado de gases combustibles.
 - a) Adquisición de datos.
- 2) Almacenamiento de datos adquiridos.
 - a) Almacenamiento de datos.
- 3) Comunicación.
 - a) Comunicación serial con Computadoras Personales.
 - b) Comunicación con el usuario.
- 4) Otros.
 - a) Inicialización del sistema y selección de operación.
 - b) Pruebas para verificar el sistema.

B.Selección del lenguaje de programación.

Se considera utilizar directamente el lenguaje ensamblador de la familia de microprocesadores MC68000, ya que éste ofrece un conjunto de instrucciones completo para llevar a cabo las operaciones necesarias de almacenamiento, de comunicación, y de control sobre los dispositivos sensores. El alcance de sus instrucciones reduce inherentemente la programación; es decir, una sola instrucción se implantaría con varias en lenguaje ensamblador de otro microprocesador. Lo anterior da como resultado la generación de códigos compactos.

También, se considera el acceso a programas de apoyo como son compiladores y simuladores.

C.Partición funcional de cada macro.

Basándose en el diseño funcional descendente, a continuación se presenta la partición funcional de cada macro del algoritmo.

El *módulo de inicialización del sistema* consiste en aquellas rutinas que definen el modo de operación de cada uno de los dispositivos del sistema; mientras que el *módulo de selección de operación* se conforma de una rutina para elegir la modalidad de funcionamiento del sistema: adquisición de datos, comunicación serial con computadoras personales y pruebas al sistema.

El *módulo de adquisición de datos* consiste en una rutina que lee un dato del convertidor analógico digital.

El *módulo de almacenamiento de datos* consiste en rutinas que escriben datos en la memoria RAM. Además, verifican la cantidad de localidades de memoria RAM disponibles.

El *módulo de comunicación serial con computadoras personales* consiste en una rutina que escribe un dato en el puerto serial. Además, verifica que no haya un sobreflujo de datos por dicho puerto para evitar la pérdida de ellos.

El *módulo de comunicación con el usuario* consiste en rutinas que escriben datos en la pantalla de cristal líquido, y que leen datos del teclado.

El *módulo de pruebas para verificar el sistema* consiste en el conjunto de rutinas que certifican el adecuado funcionamiento del sistema; por ejemplo: verificar la memoria RAM, la pantalla de cristal líquido y señales luminosas, el puerto serial y el puerto paralelo (indirectamente, el teclado y el convertidor analógico digital).

D. Almacenamiento de información.

Un sistema de adquisición de datos debe muestrear una señal en todo momento; pero no toda la información adquirida es de interés. Consecuentemente, es recomendable almacenar sólo aquella información que se considere importante. La importancia de los datos depende de la aplicación.

Podemos establecer dos esquemas generales de almacenamiento de información: almacenamiento de información de todo evento, y almacenamiento de información de eventos específicos.

1. Almacenamiento de información de todo evento.

Consiste en almacenar toda la información adquirida. Este esquema es difícil de aplicar en un sistema digital debido a la limitación en la cantidad de memoria disponible.

2. Almacenamiento de información de eventos seleccionados.

Consiste en almacenar solamente aquella información adquirida que reúna ciertos requisitos. Para decidir cuál información es de interés, se considera la forma en que varía la señal muestreada; esto es, si se trata de una señal con variaciones frecuentes, o de una señal con variaciones esporádicas. La información de interés puede consistir en:

- a) variaciones de la señal dentro de un rango.
- b) variaciones fuera de un rango.

- c) duración de las variaciones.
- d) velocidad de las variación.
- e) momento cuando ocurrieron las variaciones.

E. Esquema de la programación.

A continuación se presenta el esquema general utilizado en la programación del Sistema para Detección de Gases Explosivos.

Configuración del sistema

- 1.- Configuración del puerto paralelo PIA 6821, puerto B.
- 2.- Configuración de la pantalla de cristal líquido AND 671.
- 3.- Prueba de señales luminosas y de la pantalla de cristal líquido AND 671.

La prueba consiste en encender las señales luminosas y activar todos los puntos de la pantalla de cristal líquido AND 671, durante un segundo como mínimo.

- 4.- Configuración del puerto paralelo PIA 6821, puerto A.
- 5.- Configuración del puerto serial ACIA 6850.
- 6.- Prueba de memoria de acceso aleatorio RAM 6264.

La prueba consiste en realizar en cada localidad de memoria RAM una escritura de un número binario y

posteriormente su lectura. Si al efectuar la comparación entre ellos resulta que son iguales, entonces la localidad de memoria RAM se encuentra en buen estado; si no, la memoria RAM podría estar en mal estado o no accesarse adecuadamente, por lo que se detiene la operación del sistema.

Definición del modo de operación

- 1.- Selección del modo de operación del sistema: *adquisición de datos, comunicación con computadoras personales, o pruebas al sistema.*

La acción consiste en seleccionar un modo de operación que sea válido y en ejecutar la sección del programa correspondiente.

Modo de adquisición de datos

- 1.- Se establece el número de muestras por segundo.

Primero se lee un número de muestras por segundo que sea válido: 1, 2, 4, 5, 10, 20, 25, 50 ó 100 (divisores de 100). Después se calcula, a partir del número de muestras por segundo, cada cuántas centésimas de segundo se debe tomar una muestra. (Se generan 100 interrupciones por segundo para la adquisición de datos).

$$\frac{0.01 \text{ segundos}}{\text{muestra}} = \frac{100}{(\text{muestras/segundo})}$$

- 2.- Se establece la frontera inferior para el registro de datos.

Primero se lee una frontera inferior para el registro de datos que sea válida: un número entero comprendido entre 0 y 99. Todos los datos muestreados que sean mayores o iguales a dicha frontera serán registrados, ya sea en memoria RAM o enviados por el puerto serial, si así se desea. Después se calcula, a partir de la frontera, una palabra de 8 bits proporcional.

$$\text{entero}\left(\frac{\text{frontera} + 255}{100}\right)$$

- 3.- Se define si los datos adquiridos se registran o no se registran.
- 4.- Sólo si se define que se registren los datos adquiridos, se especifica si se almacenan en memoria RAM, o se envían por el puerto serial.
- 5.- Se inicializa la hora del sistema.

La acción consiste en: primero leer las horas (0 a 23); luego, los minutos (0 a 59); y finalmente, los segundos (0 a 59).

- 6.- Rutina que maneja las interrupciones para la adquisición de datos.

Una vez que el microprocesador ha entrado en estado de paro, la acción consiste en esperar a que se genere una interrupción (cada 0.01 segundos), para ejecutar la sección del programa encargada de adquirir datos. De este modo, el microprocesador abandona el estado de paro al atender, por medio de la interrupción, el código relacionado con la adquisición de datos. Posteriormente, si no se verifica ninguna

condición de salida, como memoria RAM llena o petición del usuario, el microprocesador volverá a ponerse en estado de paro y esperar la siguiente interrupción.

algoritmo	s.	habilita interrupciones
<i>ciclo:</i>	s.	entra en estado de paro el microprocesador
	s.	si se verifica alguna condición de salida, entonces brinca a etiqueta <i>final</i> ;
	s.	brinca a etiqueta <i>ciclo</i> ;
<i>final:</i>	s.	deshabilita interrupciones
	s.	regresa al menú principal

Sección del programa encargada de adquirir datos.

- 1.- Actualiza la hora del sistema incrementando cada centésima de segundo. En su momento, actualiza el segundero, el minuterero y el horario.
- 2.- Actualiza el contador de centésimas de segundo dedicado a indicar el momento en que se toma una muestra.
- 3.- Si es el momento de tomar una muestra (según la cantidad de muestras por segundo requeridas), entonces muestrea un dato; si no, abandona esta sección.
- 4.- Si se deben registrar los datos muestreados, entonces continúa; si no, abandona esta sección.
- 5.- Si el dato muestreado es mayor que la frontera inferior para el registro de datos, entonces continúa; si no, abandona esta sección.
- 6.- Registra el dato muestreado, ya sea que lo almacene en la memoria RAM, o que lo envíe por el puerto serial hacia la computadora personal.

Rutina que muestrea un dato.

algoritmo

- s. Genera señal de inicio de conversión para el ADC
- s. Espera señal de final de conversión del ADC
- s. Muestrea un dato (valor binario de 8 bits)
- s. Transforma el dato muestreado a notación decimal y lo muestra en pantalla
- s. Enciende señales luminosas según el dato muestreado.

Modo de comunicación serial con computadoras personales.

- 1.- Verifica si existen datos en la memoria RAM para transmitirse. Si existen, entonces continúa; si no, abandona el modo de comunicación.
- 2.- Se espera que la computadora personal se encuentre preparada para la recepción de datos.
- 3.- Se transmiten caracteres de sincronización.
- 4.- Se transmite cada dato adquirido almacenado en la memoria RAM.

Modo de pruebas al sistema.

- 1.- Prueba a las señales luminosas y a la pantalla de cristal líquido.

La prueba consiste en encender las señales luminosas y activar todos los puntos de la pantalla de cristal líquido AND 671, durante dos segundos como mínimo.

2.- Prueba al teclado.

La prueba consiste en presionar cada uno de los dieciséis botones del teclado, y verificar que el dato binario de 4 bits que se introduce al sistema es el correcto.

3.- Prueba al convertidor analógico digital.

La prueba consiste en muestrear repetidamente un dato binario de 8 bits del convertidor analógico digital ADC 0804, y mostrarlo en notación decimal legible a través de la pantalla de cristal líquido AND 671.

algoritmo

- s. Genera señal de inicio de conversión para el ADC
- s. Espera señal de final de conversión del ADC
- s. Muestrea un dato (valor binario de 8 bits)
- s. Transforma el dato muestreado a notación decimal y lo muestra en pantalla

4.- Prueba de comunicación serial con computadoras personales.

La prueba consiste en transmitir por el puerto serial ACIA 6850 una cadena de caracteres conocida, un determinado número de veces. Previamente se transmiten caracteres de sincronización.

CAPÍTULO VII.

DISEÑO DEL SISTEMA PARA DETECCIÓN DE GASES COMBUSTIBLES.

I. Introducción.

El diseño del sistema para detectar gases combustibles se puede separar en dos grandes rubros: el *"hardware"* y el *"software"*. El *"hardware"* se refiere al conjunto de elementos físicos (dispositivos) que constituyen al sistema; en oposición al *"software"*, el cual se refiere al conjunto de elementos lógicos (programación) que constituyen al sistema.

II.El "hardware" del sistema.

El "hardware" del sistema se conforma de los siguientes módulos:

- 1) Sensado de gases combustibles.
 - a) Transductor catalítico.
 - b) Conversión analógica-digital.
- 2) Almacenamiento de datos adquiridos.
 - a) Unidad de procesamiento.
 - b) Unidad de almacenamiento.
- 3) Comunicación.
 - a) Comunicación con Computadoras Personales.
 - b) Comunicación con el usuario.

A.Transductor catalítico.

El *transductor catalítico Det-Tronics C7061C* es el encargado de entregar una señal de voltaje analógica proporcional a la cantidad de gas combustible presente en la atmósfera.

El transductor Det-Tronics tiene tres terminales: dos para alimentación eléctrica (V_{cc} y tierra), y una que entrega la señal de voltaje analógica proporcional a la cantidad de gas combustible que está presente en la atmósfera.

En la Figura 7.1 se muestra la conexión del transductor a través de un amplificador de instrumentación.

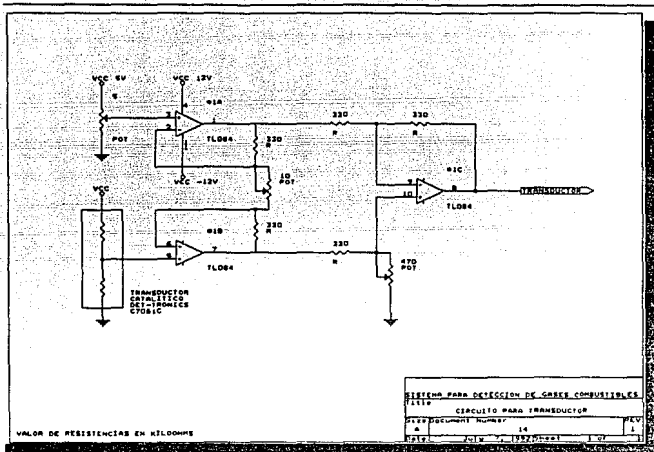


Figura 7.1: Conexión del transductor catalítico Det-Tronics C7061C.

B. Convertidor analógico-digital.

La principal función del *convertidor analógico-digital ADC0804* es proporcionar al sistema, una representación digital de los datos muestreados que provienen de la señal analógica de salida del transductor catalítico.

La temporización de la conversión analógico-digital que realiza el ADC0804 consiste: primero, se activan en bajo simultáneamente las terminales de entrada *CS* y *WR* para iniciar una conversión,

desactivándose en alto la terminal de salida *INTR*; luego, transcurren 100 μ s, que corresponden a la duración de la conversión; después, la terminal *INTR* se activa en bajo para indicar que la conversión ha terminado; finalmente, se activan en bajo simultáneamente las terminales *CS* y *RD* para leer el resultado binario de la conversión, desactivándose en alto la señal *INTR*.

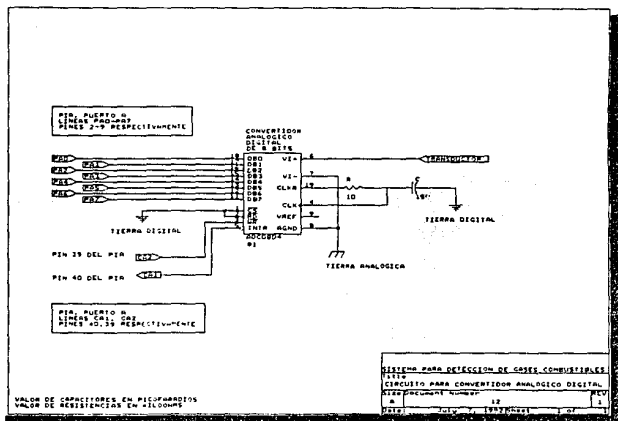


Figura 7.2: Circuito de conversión analógica-digital.

En la Figura 7.2 se muestra el circuito apropiado para conectar el convertidor ADC0804 con un puerto paralelo. La terminal *CS* se activa en bajo permanentemente para asegurar que acepte las peticiones de inicio de conversión, y proporciona las señales de final de conversión cuando éstas se generan. Consecuentemente, la

selección del convertidor por parte del microprocesador se efectúa accediendo al puerto A del PIA 6821.

También, la terminal *RD* se activa en bajo permanentemente para garantizar que el dato binario a la salida del convertidor ADC0804 esté siempre disponible. Lo anterior es posible debido a que el registro de salida del convertidor está aislado del bus de datos del microprocesador por medio del puerto paralelo.

La terminal *WR* del convertidor ADC0804 solamente se activa en bajo para indicar el inicio de conversión. Dicha señal se genera por medio de programación, a través de la terminal CA2 del puerto paralelo A del PIA 6821.

La terminal *INTR* del convertidor ADC0804 solamente se activa en bajo para indicar el final de conversión. La transición de nivel alto-bajo de dicha señal se monitorea por medio de programación, a través de la terminal CA1 del puerto paralelo A del PIA 6821.

Para finalizar, el convertidor ADC0804 tiene un tiempo de conversión de 100 μ s, y una frecuencia nominal de operación de 640 kHz.

C.Unidad de procesamiento.

La unidad de procesamiento consta del microprocesador MC68008 propiamente, de la memoria ROM 2764, del circuito de reloj, del circuito para inicializar el sistema, del circuito para decodificar el mapa de memoria, del circuito para habilitar operaciones de lectura y escritura, del circuito para generar la señal *DTACK*, del circuito para generar las señales *VPA* y *VMA*, y del circuito para generación de interrupciones.

1. Microprocesador MC68008.

La función principal del *microprocesador MC68008* es buscar, decodificar y ejecutar las instrucciones del programa que controla el sistema. A partir de ellas, genera las señales de temporización que gobiernan la operación de los módulos de memoria RAM, memoria ROM y dispositivos de Entrada/Salida.

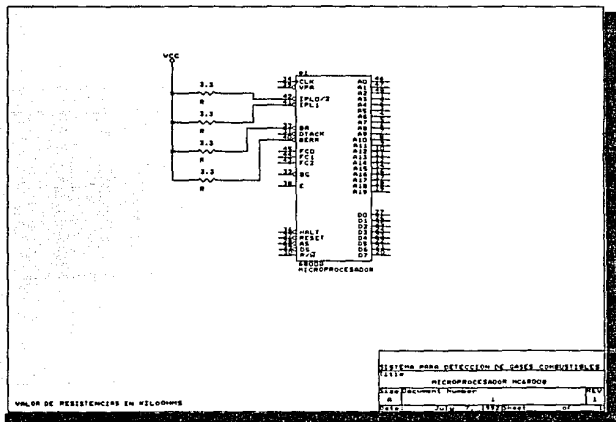


Figura 7.3: Terminales del microprocesador MC68008.

En la Figura 7.3 se muestran las terminales del microprocesador MC68008. Las terminales del bus de direcciones, A_0 a A_{19} , y del bus de datos, D_0 a D_7 , no necesitan

conectarse mediante circuitos que entreguen a su salida una mayor intensidad de corriente ("buffers"). La capacidad de corriente ("fan out") de dichas terminales puede activar las memorias y los dispositivos periféricos externos.

Sin embargo, si se desea incrementar la cantidad de memoria RAM del Sistema para Detección de Gases Combustibles, será necesario utilizar "buffers", y conectarlos en las terminales de los buses del microprocesador.

Como ningún otro dispositivo en el sistema puede llegar a tomar el control de los buses, la terminal de Petición de Bus, *BR* ("Bus Request"), se desactiva en alto por medio de una resistencia conectada a V_{cc} . Lo anterior incapacita al microprocesador de ceder el control de sus buses a otros dispositivos.

La terminal de Error de Bus, *BERR* ("Bus Error"), se desactiva en alto por medio de una resistencia conectada a V_{cc} . Lo anterior obedece a la falta de circuitería externa que indique al microprocesador cuando algún dispositivo no contesta durante una operación de lectura o escritura.

Finalmente, cada terminal de Petición de Interrupción, *IPL₂₀* e *IPL₁*, se conecta a V_{cc} por medio de una resistencia, ya que éstas se diseñaron para operar como compuertas *AND* alambradas. De este modo, se evitan falsas peticiones de interrupción.

2. Memoria ROM 2764.

La memoria ROM 2764 tiene la función de almacenar la programación del sistema, ya que es una memoria de sólo lectura ("Read Only Memory", *ROM*), y no es posible modificar los datos que contiene.

Debido a que la capacidad de almacenamiento de la memoria ROM 2764 es de 8 kilobytes (8192 bytes), el tamaño total en bytes de la programación del sistema debe ser, a lo sumo, igual a dicha capacidad.

Por último, se considera que la memoria ROM 2764 tiene un tiempo de acceso máximo de 300 ns. La operación de sus principales señales de control se muestra a continuación.

	CE	OE	Salidas
Lectura	nivel bajo	nivel bajo	D _{out}
Espera	nivel alto	*	Hi-Z
Salidas Hi-Z	*	nivel alto	Hi-Z

Tabla 7.1: Señales de control de la memoria ROM 2764.

3.Circuito de reloj.

El principal objetivo del *circuito de reloj*, es proporcionar al microprocesador y a los dispositivos periféricos, una señal de referencia estable, a una frecuencia determinada.

En la Figura 7.4 se muestra un oscilador de 8 MHz cuya señal de salida se alimenta a dos inversores 74LS04. Esto es para evitar que la carga demandada por el microprocesador y los dispositivos periféricos afecte la operación del oscilador. Así, la salida de uno de los inversores 74LS04 provee la señal maestra de reloj para el microprocesador: *CPU CLK*. Debido a que otros dispositivos en el sistema también requieren la misma señal maestra de reloj, se utiliza la salida del otro inversor 74LS04: *DEVICE CLK*. En consecuencia, el microprocesador no comparte con ningún dispositivo su señal maestra de reloj.

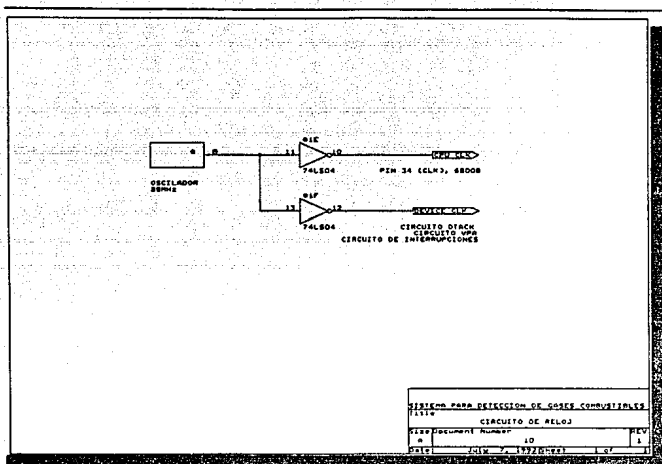


Figura 7.4: Circuito de reloj.

Es muy conveniente que las líneas de reloj sean independientes, para detectar fácilmente la ausencia de dicha señal en el microprocesador y en cualquier circuito o dispositivo periférico que la necesite.

Ya que la frecuencia nominal de operación del microprocesador MC68008 que se emplea es de 8 MHz, se utiliza una señal maestra de reloj proveniente de un oscilador de esa frecuencia.

4.Circuito para inicializar el sistema.

El principal objetivo del *circuito para inicializar el sistema*, es proporcionar al microprocesador una señal de inicialización inmediatamente después de encenderlo (señal de "Power-on RESET"). La inicialización del sistema asegura un arranque apropiado en la ejecución del programa que lo gobierna. Adicionalmente, durante la operación del sistema, este circuito puede proporcionar una señal de inicialización (*señal de "RESET"*), cuando el usuario presiona un botón designado para tal efecto.

Para inicializar el microprocesador MC68008, se requiere activar en bajo sus terminales *RESET* y *HALT* simultáneamente, al menos por 100 ms.

El circuito mostrado en la Figura 7.5 realiza la función descrita anteriormente. En éste, se utiliza un LM555 configurado para una operación monoestable o de un sólo disparo. El circuito RC¹ conectado a sus terminales 6 (*DIScharge*) y 7 (*THReshold*) fija una duración de al menos 100 ms al pulso que entrega como salida.

El monoestable LM555 se dispara aplicando un pulso activo bajo en su terminal 2 (*TRigger*). El circuito RC conectado a dicha terminal lo dispara inmediatamente después de encender el sistema. Lo anterior se debe a que inicialmente el capacitor conectado a la terminal 2 se encuentra descargado, fijándola momentáneamente a tierra. Además, es posible dispararlo al presionar el botón membretado como "RESET", ya que al hacerlo se aterriza la terminal 2.

$${}^1t = 1.1RC = (1.1)(1 \times 10^5)(0.47 \times 10^{-6}) = 517 \text{ ms}$$

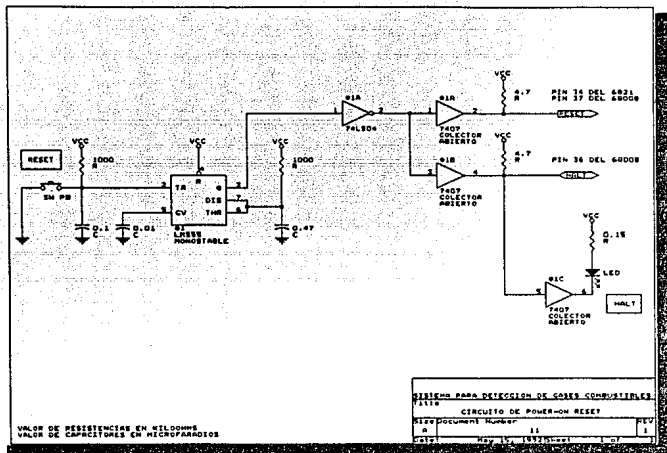


Figura 7.5: Circuito de "Power-on RESET".

Cuando se dispara el monoestable LM555, éste proporciona como salida en su terminal 3 (Q) un pulso activo alto, el cual se transforma a un pulso activo bajo con ayuda de un inversor 74LS04. Dicho pulso activo bajo se alimenta a dos "buffers" no inversores de colector abierto 7407, los cuales lo transfieren a las terminales *RESET* y *HALT* del microprocesador, inicializándolo. Se requiere utilizar "buffers" no inversores 7407 de colector abierto porque las terminales *RESET* y *HALT* del microprocesador son bidireccionales.

Cuando no está presente el pulso de inicialización, las resistencias de "pull-up" de los "buffers" no inversores 7407,

fijan las terminales *RESET* y *HALT* del microprocesador a un nivel alto.

Finalmente, para indicar visualmente que el microprocesador ha entrado en estado de paro ("*halt state*"), se conecta un "buffer" no inversor 7407 a la terminal *HALT* del microprocesador. Cuando dicha terminal se activa en bajo como consecuencia del estado de paro, se enciende un "LED" conectado a la salida de este "buffer" no inversor 7407.

5. Circuito para decodificar el mapa de memoria.

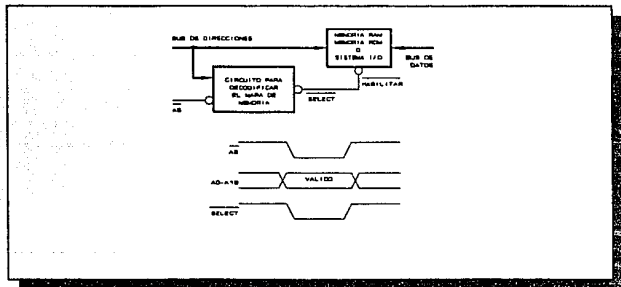


Figura 7.6: Diagrama de bloques y temporización de la lógica de decodificación de memoria.

La principal función del *circuito para decodificar el mapa de memoria*, es la generación de señales de habilitación para aquellos dispositivos que el microprocesador accesa, en función del estado de las líneas del bus de direcciones. Los dispositivos

accesados pueden pertenecer al módulo de memoria ROM, al módulo de memoria RAM o a algún módulo de E/S.

El estado del bus de direcciones lo determina la terminal AS ("Address Strobe") del microprocesador. Dicha terminal de salida, se activa en bajo para indicar que existe una dirección estable en el bus de direcciones.

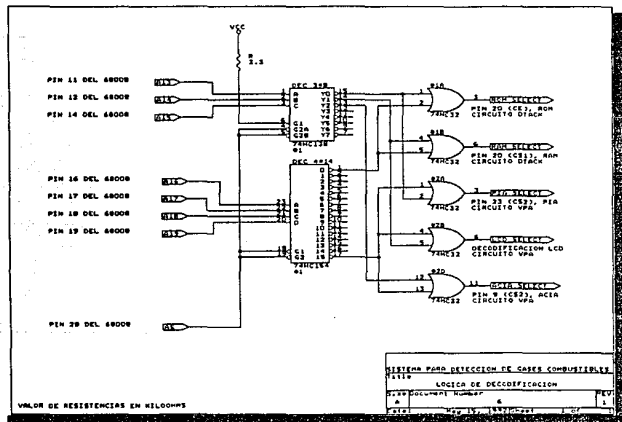


Figura 7.7: Lógica de decodificación de memoria.

La Figura 7.6 muestra un diagrama de bloques de la lógica de decodificación, junto con su temporización. Observe que la señal de habilitación *SELECT*, la cual representa una señal de habilitación genérica, se activa en bajo únicamente cuando

existe una dirección estable en el bus de direcciones y la terminal *AS* del microprocesador está activada en bajo. De otro modo, la señal *SELECT* permanece desactivada en alto. Sólo si se habilita el funcionamiento de un circuito integrado, éste es capaz de establecer una comunicación con el microprocesador a través del bus de datos. En oposición, un circuito integrado que no está habilitado, presenta una condición de alta impedancia en sus conexiones con el bus de datos.

La Figura 7.7 muestra el circuito realizado para decodificar el mapa de memoria. Las líneas del bus de direcciones que se utilizan para la lógica de decodificación son las terminales A_{13} , A_{14} , A_{15} , A_{16} , A_{17} , A_{18} y A_{19} del microprocesador. Las primeras tres se conectan a las terminales de selección *A*, *B* y *C* del decodificador 74HC138; mientras que las últimas cuatro, a las terminales de selección *A*, *B*, *C* y *D* del decodificador 74HC154. Observe que estos decodificadores sólo se habilitan cuando se activa en bajo la terminal *AS* del microprocesador. Dependiendo del estado de dichas líneas de direcciones, se activa en bajo sólo una terminal de salida (Y_n) de cada decodificador. Al final, por medio de una compuerta *OR*, 74HC32, se genera una señal de habilitación activo bajo, la cual se dedica a cierto dispositivo. Recuerde que, únicamente si sendas terminales de entrada a una compuerta *OR* se activan en bajo, su terminal de salida también se activa en bajo. En conclusión, el circuito para decodificar el mapa de memoria genera la señal *ROM SELECT* para habilitar la memoria ROM; la señal *RAM SELECT*, para la memoria RAM; la señal *PIA SELECT*, para el puerto paralelo; la señal *ACIA SELECT*, para el puerto serial; y la señal *LCD SELECT*, para la pantalla de cristal líquido. En la siguiente tabla, se muestra el mapa de memoria completo.

A19-A16	A15-A12	A11-A8	A7-A4	A3-A0	Dirección	Dispositivo
0000	000 0	0000	0000	0000	00000 ...	Memoria ROM ROM SELECT
0000	000 1	1111	1111	1111	01FFF	
0000	001 0	0000	0000	0000	02000 ...	Memoria RAM RAM SELECT
0000	001 1	1111	1111	1111	03FFF	
0000	010 0	0000	0000	0000	04000 ...	Ventana
1110	111 1	1111	1111	1111	EFFFF	
						PIA 6821 PIA SELECT repeticiones
1111	000 *	****	****	**00	F0000	Puerto A, dato
1111	000 *	****	****	**01	F0001	Puerto A, control
1111	000 *	****	****	**02	F0002	Puerto B, dato
1111	000 *	****	****	**03	F0003	Puerto B, control
						LCD AND 671 LCD SELECT repeticiones
1111	001 *	****	****	***0	F2000	Control
1111	001 *	****	****	***1	F2001	Dato
						ACIA 6850 ACIA SELECT repeticiones
1111	010 *	****	****	***0	F4000	Control
1111	010 *	****	****	***1	F4001	Dato
1111	011 0	0000	0000	0000	F6000 ...	Ventana
1111	111 1	1111	1111	1111	FFFFF	

Tabla 7.1: Mapa de memoria.

Debido a que sólo se utilizaron las terminales A_{13} a A_{19} del microprocesador para decodificar el mapa de memoria, éste se divide en 128 páginas (2^7 páginas), de 8 kB de longitud cada una (2^{13} bytes de longitud). Para direccionar cada byte que conforma una página, únicamente se emplean las líneas del bus de direcciones correspondientes a las terminales A_0 a A_{12} del microprocesador.

Se implantaron páginas de 8 kB de longitud, ya que los circuitos integrados de memoria ROM y memoria RAM utilizados son, precisamente, de dicha capacidad.

6. Circuito para habilitar operaciones de lectura o escritura.

Cualquier sistema basado en microprocesadores puede incluir módulos de memoria ROM, de memoria RAM, y dispositivos de E/S. El módulo de memoria ROM solamente se puede seleccionar para realizar operaciones de lectura. El módulo de memoria RAM y los dispositivos de E/S se pueden seleccionar para realizar operaciones de lectura o de escritura. La principal función del *circuito para habilitar operaciones de lectura o escritura*, es la generación de las señales que, dependiendo de la estructura de cada módulo, controlen dichas operaciones.

El estado del bus de datos lo determina la terminal *DS* ("*Data Strobe*") del microprocesador. Dicha terminal de salida, se activa en bajo para indicar que existe un dato estable en el bus de datos. Ahora bien, la terminal de salida *R/W* ("*Read/Write*") del microprocesador, se activa en alto para permitir una operación de lectura; o en bajo, para una operación de escritura.

Por una parte, la memoria ROM requiere de una señal activo bajo para permitir la lectura de un dato. Por otro lado, la

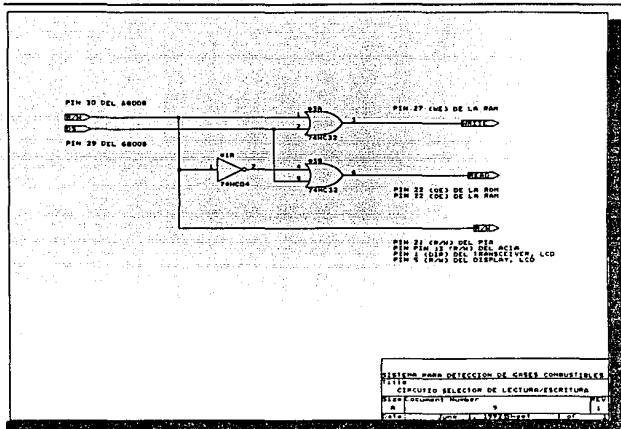


Figura 7.8: Circuito selector de lectura/escritura.

memoria RAM requiere de dos señales activo bajo: una para permitir la lectura de un dato, y otra diferente para permitir su escritura. Además, otros dispositivos como el puerto serial ACIA 6850, el puerto paralelo PIA 6821 y la pantalla de cristal líquido AND 671, requieren solamente una sola señal que se activa en bajo para realizar operaciones de escritura; o en alto, para realizar operaciones de lectura.

En la Figura 7.8 se muestra el circuito para habilitar operaciones de lectura o escritura. Se utiliza la terminal *RW* y la terminal *DS* del microprocesador para generar la señal activa bajo *READ*, asociada con operaciones de lectura; y la señal

activo bajo *WRITE*, asociada con operaciones de escritura. Con ayuda de compuertas OR, 74HC32, solamente se pueden activar en bajo alguna de las señales *READ* o *WRITE* cuando la terminal *DS* se activa en bajo.

En conclusión, se activa en bajo la señal *READ* si la señal *R/W* está activa en alto (obsérvese el inversor 74HC04). Ahora bien, se activa en bajo la señal *WRITE* si la señal *R/W* está activa en bajo.

7.Circuito para generar la señal DTACK.

El principal objetivo del *circuito para generar la señal DTACK*², es indicarle al microprocesador MC68008 el tiempo que debe esperar para que se habilite completamente cualquier dispositivo que lo intente acceder. Es decir, dicha señal indica al microprocesador que puede leer o escribir un dato válido de un dispositivo externo, sin importar la velocidad de operación de éste. Así, se realiza satisfactoriamente una transferencia asíncrona de datos entre el microprocesador y el dispositivo externo.

La terminal de entrada *DTACK* del microprocesador recibe una señal activo bajo, generada por lógica externa, la cual indica que un dispositivo externo ha recibido exitosamente un dato válido del bus de datos durante un proceso de escritura; o bien, que ha colocado un dato válido en el bus de datos durante un proceso de lectura.

La Figura 7.9 muestra un diagrama de bloques del circuito para generar la señal *DTACK*, junto con su temporización. Conceptualmente, la señal *SELECT* generada por la lógica de

² Acrónimo de "Data Transfer ACKnowledge", *DTACK*.

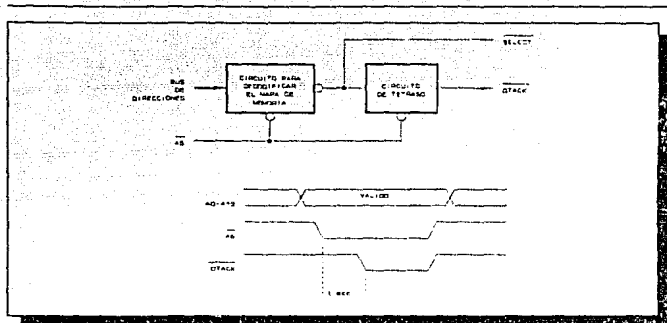


Figura 7.9: Diagrama de bloques y temporización del circuito de retardo para "DTACK".

decodificación, se introduce a un circuito de retraso, en cuya salida se obtiene la señal *DTACK*. El circuito de retraso demora la generación de la señal *DTACK*, el tiempo suficiente para permitir la habilitación del dispositivo externo accedido. En el diagrama de temporización, el intervalo de habilitación o de acceso, se representa con t_{acc} . También, se muestra claramente que *DTACK* se activa en bajo después que la terminal *AS* del microprocesador lo ha hecho; y que la señal *DTACK* se desactiva en alto cuando *AS* así lo hace. Es por esto que la señal *AS* se involucra en su generación.

La Figura 7.10 muestra el circuito para generar la señal *DTACK*. Esencialmente consiste de un registro de corrimiento de 2 bits, el cual demora las señales de habilitación *ROM SELECT* y *RAM SELECT* (generadas por la lógica de decodificación), el equivalente al tiempo de acceso de los circuitos integrados de memoria involucrados. La terminal *AS* del microprocesador se conecta a la terminal *PRReset* de ambos "Flip-Flop" D, 74HC74, por medio de un inversor. De este modo, la señal *AS* pone en

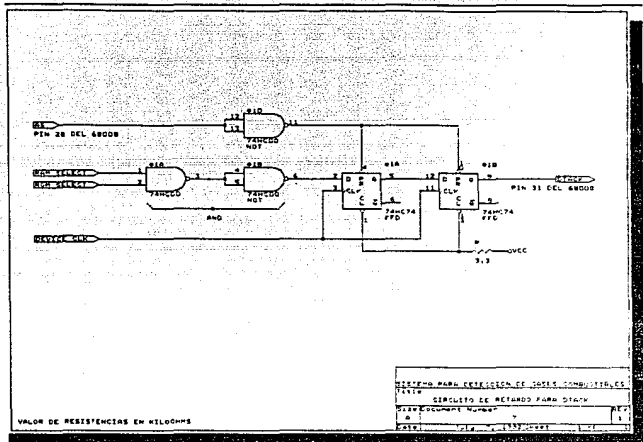


Figura 7.10: Circuito de retardo para "DTACK".

alto las terminales de salida de los "Flip-Flop" D cuando se desactiva en alto; y demora las señales de habilitación cuando se activa en bajo. El tiempo de retraso que introduce este circuito es de 250 nanosegundos.

8.Circuito para generar las señales VPA y VMA.

El principal objetivo del circuito para generar las señales VPA ("Valid Port Address"), y VMA ("Valid Memory Address"),

es indicarle al microprocesador MC68008 que el dispositivo que accesa transfiere datos en forma síncrona.

Inicialmente, el microprocesador inicia un ciclo asíncrono de lectura o escritura. Posteriormente, circuitería externa activa en bajo la terminal *VPA* ("Valid Peripheral Address") del microprocesador, para indicarle que se realizará una transferencia síncrona. Entre tanto, dicha circuitería externa espera a que la terminal *E* ("Enable") del microprocesador presente un nivel bajo para activar en bajo la señal *VMA*, la cual habilita el periférico. Esto es, la señal *VMA* se sincroniza con el flanco de bajada de la señal *E*.

La señal *E* tiene una frecuencia de la décima parte de la señal maestra de reloj.

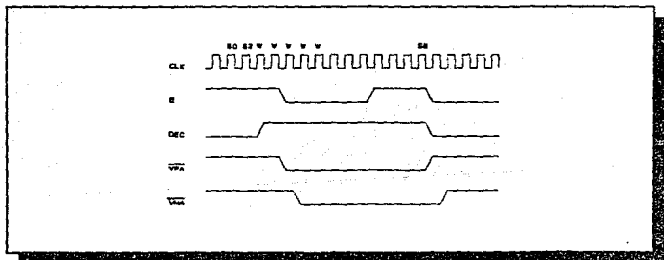


Figura 7.12: Temporización del circuito generador de *VMA* y *VPA*.

El circuito de la Figura 7.11 realiza la operación anteriormente mencionada. La lógica de decodificación del mapa de memoria activa en alto una señal cuando se accesa un dispositivo periférico síncrono, la cual se introduce al primer "Flip-Flop" por la terminal *J*. Entonces, la salida *Q* se pone a "1"

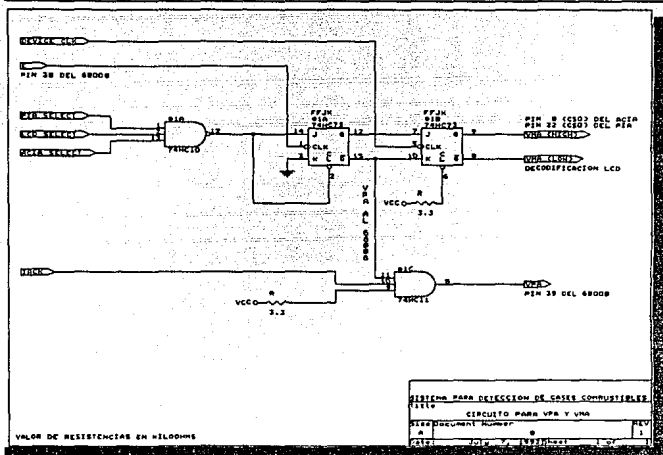


Figura 7.11: Circuito generador de VMA y VPA.

en el siguiente flanco de bajada de la señal E ; mientras que la salida Q negada activa en bajo la señal VPA que se introduce al microprocesador. El segundo "Flip-Flop" toma las señales del primero y las retrasa durante un ciclo del reloj maestro del sistema (obsérvese que las terminales de entrada J y K son complementarias), activando en bajo VMA en su salida Q negada. La temporización del circuito se muestra en la Figura 7.12.

9. Circuito para generación de interrupciones.

El principal objetivo del *circuito para generación de interrupciones*, es establecer una base de tiempo para administrar la adquisición de datos.

Se obtiene una señal de reloj de 100 Hz a partir de la división en frecuencia de una señal de 200 Hz, la cual proviene del generador de frecuencias MC14411. Esta, mediante un "Flip-Flop" D, genera una petición de interrupción al microprocesador cada centésima de segundo, activando en bajo la terminal *IPL1*.

El circuito mostrado en la Figura 7.13 genera las peticiones de interrupción. Inicialmente, la salida *Q* del "Flip-Flop" D se encuentra en "1"; pero cuando ocurre el flanco de subida de su señal de reloj de 100 Hz, cambia a "0" (la entrada *D* se encuentra a tierra). Lo anterior constituye la petición de interrupción, ya que ésta se conecta a la terminal *IPL₁*. A continuación, el microprocesador reconoce ésta petición, activando en alto sus terminales *FC0*, *FC1* y *FC2*. Introduciendo estas señales junto con *AS* negada a una compuerta *NAND*, se genera una señal de reconocimiento de interrupción: *IACK*. La señal *IACK* se introduce en la terminal *VPA* del microprocesador para que, mediante la generación de un autovector, ejecute la rutina de interrupción correspondiente. También, la señal *IACK* se retrasa un ciclo del reloj maestro mediante otro "Flip-Flop" D, para regresar el primer "Flip-Flop" D a sus condiciones iniciales, ya que se introduce por su terminal *PRReset* (salida *Q* en "1").

D. Unidad de almacenamiento.

La *memoria RAM 6264* tiene la función de almacenar los datos adquiridos por el sistema, ya que es una memoria de acceso aleatorio

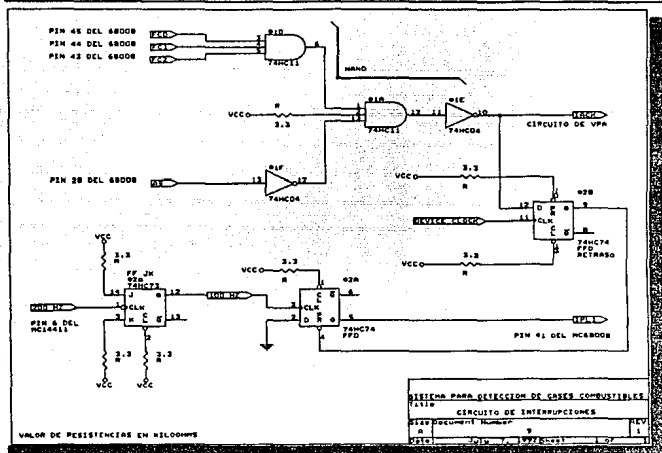


Figura 7.13: Circuito para generación de interrupciones.

("Random Access Memory", RAM), en la que se realizan operaciones de lectura y escritura.

Debido a que la capacidad de almacenamiento de la memoria RAM 6264 es de 8 kilobytes (8192 bytes), la cantidad de datos registrados por el sistema debe ser menor a dicha capacidad. Considérese que la pila de almacenamiento de datos que el sistema utiliza para resguardar la información de sus registros internos de trabajo, cuando ejecuta subrutinas o atiende excepciones, también reside en memoria RAM.

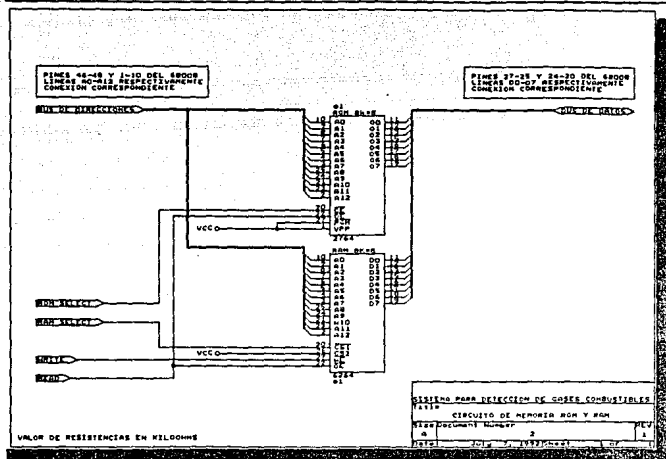


Figura 7.14: Circuitos de memoria RAM y ROM.

Por último, se considera que la memoria RAM 6264 tiene un tiempo de acceso máximo de 150 ns. La operación de sus principales señales de control se muestra a continuación. En la Figura 7.14 se muestra la conexión de la memoria RAM y de la memoria ROM.

	CS1	CS2	WE	OE	Salidas
Lectura	nivel bajo	nivel alto	nivel alto	nivel bajo	D _{out}
Escritura	nivel bajo	nivel alto	nivel bajo	nivel alto	Hi-Z
Espera	nivel alto	*	*	*	Hi-Z

Espera	*	nivel bajo	*	*	Hi-Z
Salidas Hi-Z	*	*	*	nivel alto	Hi-Z

Tabla 7.2: Señales de control de la memoria RAM 6264.

E. Comunicación serial con Computadoras Personales.

El módulo de comunicación serial con computadoras personales lo conforman el ACIA 6850 (puerto serial), el MC14411 (generador de frecuencias), y el 1488 (interfaz RS-232).

1. Puerto serial ACIA 6850.

El módulo de comunicación serial consiste de la circuitería necesaria para comunicarse con el mundo exterior por medio de una terminal compatible con el estándar EIA RS-232. El circuito que se encarga de la transmisión y recepción serial de datos es el ACIA 6850. Sólomente requiere de circuitería externa para generar una señal de reloj con la frecuencia seleccionada para la recepción y la transmisión de datos. En la Figura 7.15 se muestra la conexión de este circuito.

2. Generador de frecuencias MC14411.

Es el encargado de generar una señal de reloj de 9600 Hz que se utiliza para las operaciones de transmisión de datos por parte del sistema. Para su operación, se requiere de un cristal de 1.8432 MHz y de una resistencia de 15 M Ω .

Además, genera una señal de reloj de 200 Hz que se utiliza para la producción de interrupciones.

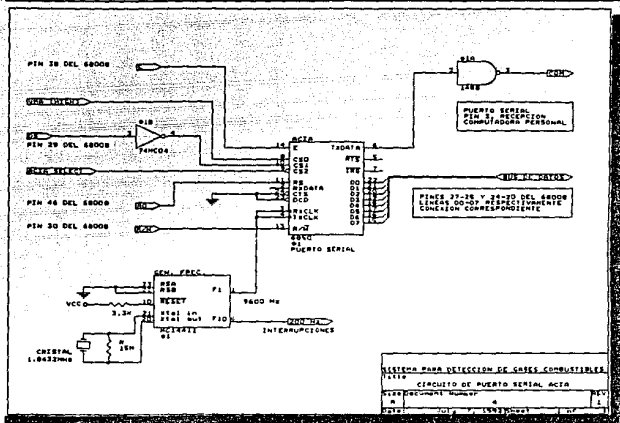


Figura 7.15: Circuito del puerto serial ACIA 6850.

3. Interfaz RS-232-C.

El circuito 1488 convierte una señal con niveles TTL a niveles RS-232. Requiere dos fuentes de voltaje: una positiva de +12 V, y otra negativa de -12 V.

F. Comunicación con el usuario.

El módulo de comunicación con el usuario comprende el puerto paralelo PIA 6821, el teclado y la pantalla de cristal líquido AND 671, así como señales indicadoras luminosas.

1. Puerto paralelo PIA 6821.

A través del *puerto paralelo PIA 6821*, el microprocesador MC68008 se comunica con el convertidor analógico-digital ADC0804, con el teclado y con las señales luminosas.

El puerto A del PIA se conecta al convertidor analógico-digital ADC0804; mientras que en el puerto B, los primeros cuatro bits se comunican con el teclado, y los segundos con las señales luminosas. Véase Figura 7.16.

La línea CA1 y CA2 del puerto A se programan, una de entrada y la otra de salida respectivamente. La línea CB1 del puerto B se programa como entrada.

2. Teclado.

El principal objetivo del *teclado* es introducir al sistema aquellos datos que el usuario tiene que proporcionar para configurar el sistema, o bien, para definir algún parámetro o elegir alguna opción de funcionamiento de interés.

El teclado está formado por lógica combinatorial de codificación, un "Flip-Flop" y un registro. La lógica combinatorial codifica 16 líneas de entrada, que se activan excluyentemente,

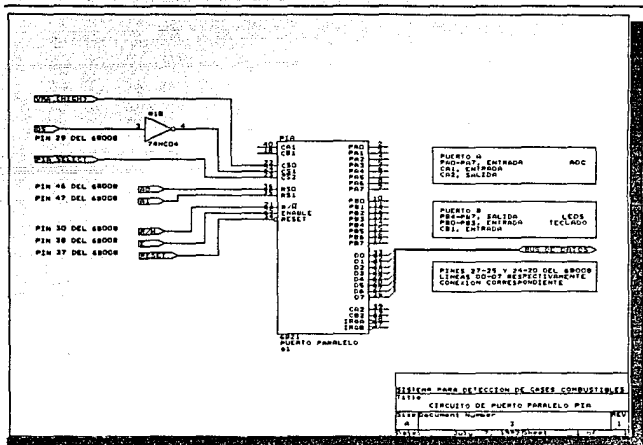


Figura 7.16: Circuito del puerto paralelo PIA 6821.

en 4 líneas de salida. La salida del codificador corresponde al valor binario representado en cuatro bits, de la posición de la tecla oprimida. Véase Figura 7.17 y Figura 7.18.

Cuando se oprime una tecla, la señal *TECLA* (tecla oprimida), se habilita en bajo para permitir la carga del dato binario de 4 bits en el registro del teclado. La señal *DATO LISTO*, generada a partir del retraso de la señal *TECLA*, indica al sistema que un dato binario válido se encuentra en el registro del teclado.

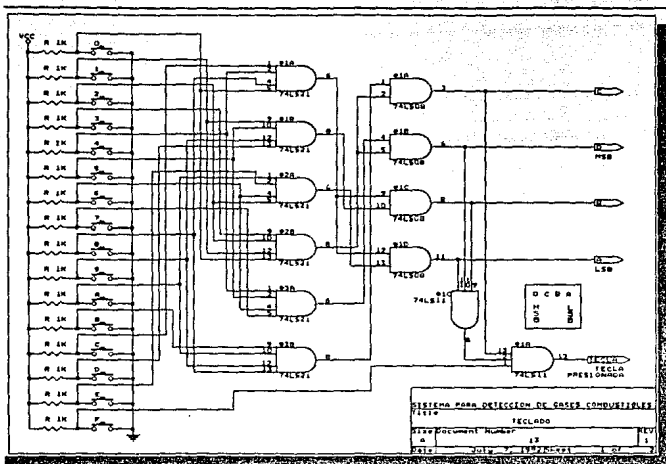


Figura 7.17: Teclado

Los "Flip-Flops" D elimina los rebotes de las señales que se generan al presionar un botón.

3. Señalización luminosa.

La principal función de la *señalización luminosa* es, proporcionar al usuario una indicación visual de la concentración de gas combustible presente en la atmósfera.

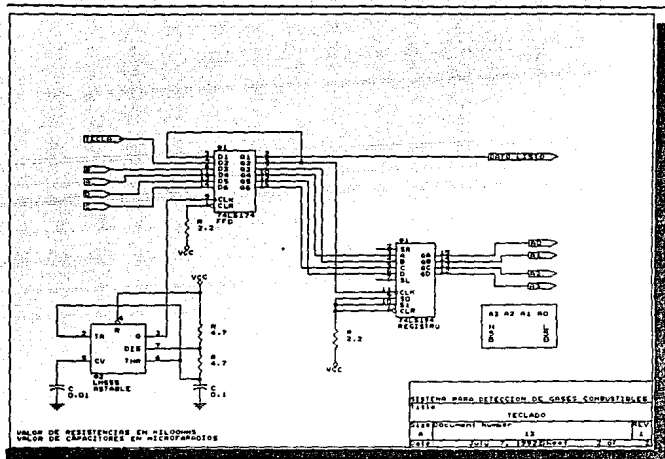


Figura 7.18: Teclado (continuación)

De este modo, el "LED" de color verde enciende si la concentración del gas combustible en la atmósfera va del 0% al 25% del LEL; el "LED" de color amarillo, si la concentración va del 25% al 50%; y el "LED" rojo, si la concentración es mayor al 50%. Adicionalmente, un "LED" de color blanco enciende si se registra un dato en memoria. Todo lo anterior está definido en la programación del sistema.

4. Pantalla y su interfaz.

El principal objetivo de la *pantalla de cristal líquido* AND 671, es proporcionar al usuario indicaciones legibles del estado del sistema, para que éste proporcione algún dato o ejecute acciones preparativas. También, muestra una representación de los datos muestreados por el sistema inmediatamente.

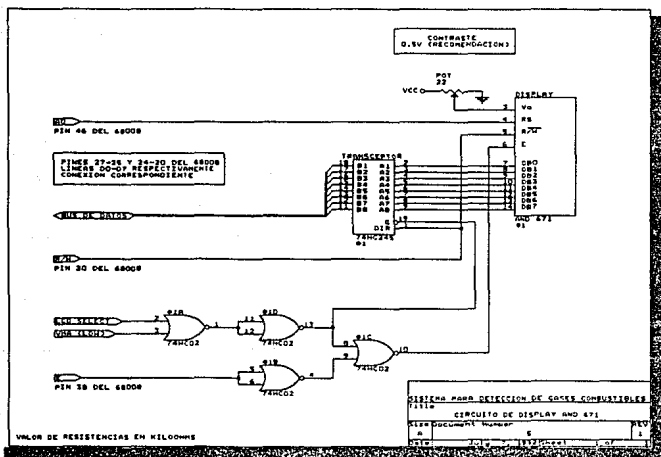


Figura 7.19: Pantalla de cristal líquido AND 671.

En la Figura 7.19 se muestra este circuito. La pantalla se conecta al bus de datos del microprocesador a través de un

registro bidireccional de 8 bits de tres estados ("*Transceptor*"), que se comunica con el microprocesador al efectuarse un ciclo síncrono.

El *transceptor* utilizado fue el 74HC245; este circuito cuenta con dos señales de control: la señal de habilitación (*G*), y la señal de dirección de datos (*DIR*). La señal *G* se activa en bajo cuando las señales *VMA* y *LCD SELECT* se activan en bajo. La señal *DIR* se conecta a la terminal *RAW* del microprocesador.

Finalmente, para acceder de una forma síncrona a la pantalla de cristal líquido AND 671, se utilizan las señales *LCD SELECT*, *VMA* y *E*, para generar un único pulso que le permite ejecutar comandos. Sólomente cuando las señales *LCD SELECT* y *VMA* están ambas activadas en bajo, se permite el paso de la señal *E* a la pantalla de cristal líquido.

III. Software del sistema.

El "*software*" del sistema se conforma de los siguientes macros:

- 1) Sensado de gases combustibles.
 - a) Adquisición de datos.
- 2) Almacenamiento de datos adquiridos.
 - a) Almacenamiento de datos.
- 3) Comunicación
 - a) Comunicación serial con Computadoras Personales.
 - b) Comunicación con el usuario.
- 4) Otros
 - a) Inicialización del sistema y selección de operación.
 - b) Pruebas para verificar el sistema.

A continuación se presenta el código fuente completo de la programación del Sistema para Detección de Gases Combustibles. El programa fue realizado en lenguaje ensamblador del microprocesador MC68008 y compilado con la utilería A68K.EXE de *Quele*. Cada línea del programa se acompaña de un comentario que clarifica su función.

Localidades de memoria 00000h a 00003h : 00003FFEh
 Inicialización, apuntador a pila de interrupciones inicial
 Localidades de memoria 00004h a 00009h : 00000600h
 Inicialización, contador de programa inicial, comienzo de programa
 Localidades de memoria 00068h a 00071h : 00001000h
 Autovector, nivel 2 de interrupciones, comienzo de subrutina

```

org $600 ; dirección inicial del programa
; .....
; CONFIGURACION DEL PIA 6821 PUERTO B
; .....
move.b #$3a,$f0003 ; interrupciones deshabilitadas
; IRQB1 monitorea transición bajo-alto de línea CB1
; línea CB2 configurada como salida
; control inmediato sobre línea CB2
; selecciona registro de dirección de datos
move.b #$f0,$f0002 ; configura como salida los bits PB4-PB7
; configura como entrada los bits PB0-PB3
move.b #$3e,$f0003 ; selecciona registro de salida
; .....
; CONFIGURACION DEL DISPLAY AND671
; .....
bsr delay ; retardo de 0.1 segundos, "power-on" AND671
move.b #$38,$f2000 ; comando "function set" del AND671
; 8 bits para longitud de datos: DL=1
; configuración AND671: M=1, F=0
bsr lcdw ; verifica el estado del display
move.b #$0c,$f2000 ; comando "display ON/OFF control" del AND671
; display encendido: D=1
; cursor apagado: C=0
; carácter sin parpadeo: B=0
bsr lcdclr ; borra display
; .....
; PRUEBA DE SEÑALES LUMINOSAS Y DISPLAY
; UTILIZA REGISTROS
; A0 : pase de parámetros a rutina LCDSTR
; .....
move.l #$00000400,a0 ; dir. de inicio de la cadena
bsr lcdstr ; coloca cadena en display

```

```

move.b #$00,$f0002 ; prende señales luminosas (PIA pto. B)
bsr delay1          ; retardo de 1 segundo
move.b #$f0,$f0002 ; apaga señales luminosas (PIA pto. B)
; ..
move.l #$00000410,a0 ; dir. de inicio de la cadena "CONFIGURANDOSE"
bsr lcdstr          ; coloca cadena en display
bsr delay           ; retardo de 0.1 segundos
; ..
; CONFIGURACION DEL PIA 6821 PUERTO A
; ..
move.b #$3B,$f0001 ; interrupciones deshabilitadas
; IRQA1 monitorea transición alto-bajo de línea CA1
; línea CA2 configurada como salida
; control inmediato sobre línea CA2
; selecciona registro de dirección de datos
move.b #$00,$f0000 ; configura como entrada el puerto
move.b #$3c,$f0001 ; selecciona registro de salida
; ..
; CONFIGURACION DEL ACIA 6850
; ..
move.b #$03,$f4000 ; comando "master reset" del ACIA 6850
move.b #$14,$f4000 ; interrupciones deshabilitadas
; 8 bits de datos + 1 bit de parada (no paridad)
; NOTA: opera a 9600 bauds CONF. POR HARDWARE
; ..
; PRUEBA DE MEMORIA RAM
; UTILIZA REGISTROS
; ..
D0 ; contador de 1FFFh hasta 0h
D1 ; dato leído de memoria RAM
A0 ; pase de parámetros a rutina LCDSTR
A1 ; dirección de memoria RAM en prueba
; ..
clr.l d0 ; borra registro D0
clr.l d1 ; borra registro D1
move.l #$00000420,a0 ; dir. de inicio de la cadena "VERIFICANDO RAM"
bsr lcdstr          ; coloca cadena en display
bsr delay           ; retardo de 0.1 segundos
move.l #$00002000,a1 ; coloca 2000h en A1, (dir. inicial de RAM)
move.w #$1fff,d0    ; para D0 = 1FFFh hasta 0h decrementando en 1h
ramtst:move.b #$55,(a1) ; coloca 55h en (A1), escribe un dato en RAM
move.b (a1),d1      ; coloca (A1) en D1, lee el dato de RAM
cmpi.b #$55,d1      ; compara 55h con D1
bne ramerr          ; si son diferentes, brinca a ramerr:
move.b #$aa,(a1)    ; coloca AAh en (A1), escribe un dato en RAM
move.b (a1),d1      ; coloca (A1) en D1, lee el dato de RAM
movc.b (a1)+,d1     ; incrementa A0 (siguiente dir. de RAM)
; compara AAh con D1
cmpi.b #$aa,d1      ; compara AAh con D1
bne ramerr          ; si son diferentes, brinca a ramerr:
dbf d0,ramtst       ; siguiente D0
bra.s ramend        ; brinca a ramend: (prueba exitosa)
; ..
ramerr:move.l #$00000430,a0 ; ** dir. de inicio de la cadena "ERROR EN RAM" **
bsr lcdstr          ; ** coloca cadena en display **
stop #$2000         ; ** detiene operación debido a ERROR EN RAM **
; ..
ramend:nop          ; no operación
; ..
; MENU

```

```

; SELECCIÓN DE MODO DE OPERACION
; UTILIZA REGISTROS
;   D4 : pase de parámetros a rutina TCDR
;   A0 : pase de parámetros a rutina LCDSTR
;   A5 : dirección de almacenamiento de datos adquiridos
; *****
menu: move.l #$00002000,a5 ; no hay datos almacenados en RAM, pila vacía
      move.l #$00000440,a0 ; dir. de inicio de la cadena "OPCION (EDC)"
      bsr lcdstr          ; coloca cadena en display
      bsr lcdw           ; verifica el estado del display
      move.b #$c7,$f2000 ; posiciona cursor en 47
      bsr lcdcon         ; prende cursor del display
      bsr delay          ; retardo de 0.1 segundos
      bsr tcdr           ; lee carácter del teclado
      bsr lcdcof         ; apaga cursor del display
                        ; según D4, elige una opción
      cmpi.b #$0c,d4     ; opción 0ch
      beq menu0         ; entonces brinca a menu0:
                        ; *** MODO DE ADQUISICION DE DATOS ***
      cmpi.b #$0d,d4     ; opción 0dh
      beq menu1         ; entonces brinca a menu1:
                        ; *** MODO DE COMUNICACION CON PC ***
      cmpi.b #$0e,d4     ; opción 0eh
      beq menu2         ; entonces brinca a menu2:
                        ; *** MODO DE PRUEBAS AL SISTEMA ***
      cmpi.b #$0f,d4     ; opción 0fh
      beq menu          ; entonces brinca a menu: (escape)
                        ; cualquier otra opción
      move.l #$00000450,a0 ; dir. de inicio de la cadena "OPCION INVALIDA"
      bsr lcdstr          ; coloca cadena en display
      bsr delay          ; retardo de 0.1 segundos
      bra menu           ; brinca a menu:
                        ; fin de elige una opción
; *****
; MENU0
; MODO DE ADQUISICION DE DATOS
; *****
; UTILIZA REGISTROS
;   A0 : pase de parámetros a rutina LCDSTR
menu0: move.l #$00000540,a0 ; dir. de inicio de la cadena "MODO DE MUESTREO"
      bsr lcdstr          ; coloca cadena en display
      bsr delay1         ; retardo de 1 segundo
; *****
; ESTABLECE MUESTRAS POR SEGUNDO
; 1, 2, 4, 5, 10, 20, 25, 50 ó 100 MUESTRAS/SEGUNDO
; UTILIZA REGISTROS
;   D0 : apuntador a tabla de divisores de 100
;   D1 : divisor de 100 elegido
;       resultado intermedio para display
;   D2 : vector de configuración (0.01s/muestra)
;   D4 : pase de parámetros a rutina TCDR
;   D7 : pase de parámetros a rutina LCDDD
;   A0 : pase de parámetros a rutina LCDSTR
;   A1 : dir. de inicio de tabla de div. de 100
; *****
pmues: clr.l d0          ; borra registro D0

```



```

clr.l d1 ; borra registro D1
clr.l d2 ; borra registro D2
move.l #$00000460,a1 ; dir. de inicio de la tabla de divisores de 100
move.l #$000004f0,a0 ; dir. de inicio de la cadena "001?MUESTRAS/S"
pmop: bsr lcdstr ; coloca cadena en display
bsr lcdw ; verifica estado del display
move.b #$83,$f2000 ; posiciona cursor en 03
bsr lcdcon ; prende cursor del display
bsr delay ; retardo de 0.1 segundos
bsr tcdr ; lee carácter del teclado
bsr lcdcof ; apaga cursor del display
; según D4, elige una opción
cmpi.b #$0f,d4 ; opción 0Fh
beq pmues ; entonces brinca a pmues: (escape)
cmpi.b #$0b,d4 ; opción 0Bh
beq pmsig ; entonces brinca a pmsig: (otro div. 100)
cmpi.b #$0a,d4 ; opción 0Ah
beq pmend ; entonces brinca a pmend: (acepta div. 100)
; cualquier otra opción
bra pmop ; brinca a pmop:
; fin de elige una opción
pmsig: addi.b #$1,d0 ; suma 1h en D0, apunta a siguiente div. 100
cmpi.b #$09,d0 ; compara 9h con D0
bne pmsigl ; si son diferentes, brinca a pmsigl:
clr.l d0 ; borra registro D0 (recorrido circular de tabla)
pmsigl:clr.l d1 ; borra registro D1
bsr lcdw ; verifica estado del display
move.b #$80,$f2000 ; posiciona cursor en 00
move.b 00(a1,d0),d1 ; coloca siguiente div. 100 en D1
divu.w #$0064,d1 ; divide D1 = D1/100 (residuo.w:cociente.w)
move.b d1,d7 ; coloca D1 en D7
bsr lcddd ; coloca número binario <10 en display (centenas)
swap d1 ; intercambia palabras en D2 (cociente.w:residuo.w)
andi.l #$0000ffff,d1 ; filtra palabra baja en D1
divu.w #$000a,d1 ; divide D1 = D1/.0 (residuo.w:cociente.w)
move.b d1,d7 ; coloca D1,D7
bsr lcddd ; coloca número binario <10 en display (decenas)
swap d1 ; intercambia palabras en D2 (cociente.w:residuo.w)
move.b d1,d7 ; coloca D1 en D7
bsr lcddd ; coloca número binario <10 en display (unidades)
bra pmop ; brinca a pmop:
pmend: move.b 00(a1,d0),d2 ; coloca siguiente div. 100 en D2
move.l #$00000064,d1 ; coloca 64h en D1, (100 decimal)
divu.w d2,d1 ; divide D1=D1/D2 (CENTESIMAS DE SEGUNDO/MUESTRA)
move.w d1,d2 ; coloca D1 en D2
swap d2 ; intercambia palabras en D2
bsr delay ; retardo de 0.1 segundos
; *****
; ESTABLECE FRONTERA (REGISTRO Y ALARMA)
; UTILIZA REGISTROS
; D0 : resultado intermedio para D2
; D2 : vector de configuración (frontera)
; D4 : pase de parámetros a rutina TCDR
; D7 : pase de parámetros a rutina LCDDD
; A0 : pase de parámetros a rutina LCDSTR
; *****
pfron: clr.l d0 ; borra registro D0
clr.w d2 ; borra registro D2 (únicamente WORD)

```

```

move.l #$000004e0,a0 ; dir. de inicio de la cadena "001?FRONTERA"
bsr lcdstr           ; coloca cadena en display
                    ; PRIMER DIGITO (0 a 9)
bsr lcdcon          ; prende cursor del display
bsr lcdw            ; verifica el estado del display
move.b #$80,$f2000 ; posiciona cursor en 00
pfd1: bsr tcdr      ; lee carácter del teclado
cmpi.b #$0f,d4     ; compara 0Fh con D4 (escape)
beq pfron          ; si son iguales, brinca a pfron:
cmpi.b #$09,d4     ; compara 09h con D4
bgt pfd1           ; si d4>9h, brinca a pfd1:
move.b d4,d0       ; coloca D4 en D0
mulu.w #$000a,d0   ; multiplica D0=D0*10 (decenas)
move.b d4,d7       ; coloca D4 en D7
bsr lcdddd         ; coloca número binario <10 en display
                    ; SEGUNDO DIGITO (0 a 9)
pfd2: bsr tcdr      ; lee carácter del teclado
cmpi.b #$0f,d4     ; compara Fh con D4 (escape)
beq pfron          ; si son iguales, brinca a pfron:
cmpi.b #$09,d4     ; compara 9h con D4
bgt pfd2           ; si d4>9h, brinca a pfd2:
add.b d4,d0        ; suma D4 en D0 (decenas + unidades)
mulu.w #$00ff,d0   ; multiplica D0=D0*255 ** PALABRA DE 8 BITS
divu.w #$0064,d0   ; divide D0=D0/100 ** (multiplica por 2.55)
move.b d0,d2       ; coloca D0 en D2 (vector de configuración)
swap d2            ; intercambia palabras en D2
move.b d4,d7       ; coloca D4 en D7
bsr lcdddd         ; coloca número binario <10 en display
bsr lcdcof         ; apaga cursor del display
bsr delay          ; retardo de 0.1 segundos
; *****
; ¿REGISTRA INFORMACION?
; UTILIZA REGISTROS
; D4 : pase de parámetros a rutina TCDR
; A0 : pase de parámetros a rutina LCDSTR
; A3 : vector de configuración (¿reg. inf.?)
; A4 : vector de configuración (borrado)
; *****
preg: move.l #$00000500,a0 ; dir. de inicio de la cadena "SI?REGISTRAR"
move.l #$00000510,a3     ; dir. de inicio de la cadena "NO?REGISTRAR"
prop: bsr lcdstr         ; coloca cadena en display
bsr lcdw                ; verifica estado del display
move.b #$82,$f2000     ; posiciona cursor en 02
bsr lcdcon              ; prende cursor del display
bsr delay                ; retardo de 0.1 segundos
bsr tcdr                ; lee carácter del teclado
bsr lcdcof              ; apaga cursor del display
cmpi.b #$0f,d4          ; compara Fh con D4 (escape)
beq preg                ; si son iguales, brinca a preg:
cmpi.b #$0b,d4          ; compara Bh con D4 (otra opción)
beq prsig               ; si son iguales, brinca a prsig:
cmpi.b #$0a,d4          ; compara Ah con D4 (elegir opción)
beq prend               ; si son iguales, brinca a prend:
bra prop                ; brinca a prop:
prsig: exg.l a0,a3      ; intercambia A0 con A3, opciones "SI" y "NO"
bsr lcdstr              ; coloca cadena en display
bra prop                ; brinca a prop:
prend: exg.l a0,a3     ; intercambia A0 con A3, opciones "SI" y "NO"

```

```

bsr delay ; retardo de 0.1 segundos
move.l #$00000000,a4 ; borra registro A4 "clr.l a4"
cmpa.w #$0500,a3 ; compara 500h con A3
bne phora ; si son diferentes, brinca phora:
; ** se salta la siguiente rutina de selección **
; *****
; ALMACENA INFORMACION EN MEMORIA O
; LA ENVIA POR PUERTO SERIAL
; UTILIZA REGISTROS
; D4 : pase de parámetros a rutina TCDR
; A0 : pase de parámetros a rutina LCDSTR
; A4 : vector de configuración
; *****
palm: move.l #$00000520,a0 ; dir. de inicio de la cadena "MEM?ALMACENA"
move.l #$00000530,a4 ; dir. de inicio de la cadena "COM?ENVIA"
bsr lcdstr ; coloca cadena en display
paop: bsr lcdw ; verifica estado del display
move.b #$83,$f2000 ; coloca cursor en 03
bsr lcdcon ; prende cursor del display
bsr delay ; retardo de 0.1 segundos
bsr tcdr ; lee carácter del teclado
bsr lcdcof ; apaga cursor del display
cmpi.b #$0f,d4 ; compara Fh con D4 (escape)
beq palm ; si son iguales, brinca a palm:
cmpi.b #$0b,d4 ; compara Bh con D4 (otra opción)
beq pasig ; si son iguales, brinca a pasig:
cmpi.b #$0a,d4 ; compara Ah con D4 (elegir opción)
beq paend ; si son iguales, brinca a paend:
bra paop ; brinca a paop:
pasig: exg.l a0,a4 ; intercambia A0 con A4, opciones "MEM" y "COM"
bsr lcdstr ; coloca cadena en display
bra paop ; brinca a paop:
paend: exg.l a0,a4 ; intercambia A0 con A4, opciones "SI" y "NO"
cmpa.l #$00000530,a4 ; compara 540h con A4, transmite a PC
bne panpc ; si son diferentes, brinca a panpc:
move.l #$000005c0,a0 ; dir. inicial de cadena "PREPARE PC"
bsr lcdstr ; coloca cadena en display
bsr lcdw ; verifica estado del display
move.b #$c7,$f2000 ; posiciona cursor en 47
bsr lcdcon ; prende cursor del display
bsr tcdr ; espera a que se presione una tecla
bsr lcdcof ; apaga cursor del display
panpc: bsr delay ; retardo de 0.1 segundos
; *****
; INICIALIZA LA HORA DEL SISTEMA
; UTILIZA REGISTROS
; D0 : resultado intermedio para D5
; D4 : pase de parámetros a rutina TCDR
; D5 : HORA:MINUTO:SEGUNDO:CENTESIMAS
; D7 : pase de parámetros a rutina LCDDD
; *****
phora: clr.l d0 ; borra registro D0
clr.l d5 ; borra registro D5
move.l #$00000490,a0 ; dir. de inicio de la cadena "00:00:00 HORA"
bsr lcdstr ; coloca cadena en display
; PRIMER DIGITO (0 a 2) HORAS
bsr lcdcon ; prende cursor del display
bsr lcdw ; verifica el estado del display

```

```

phd1:  move.b #\$80,$f2000 ; posiciona cursor en 00
        bsr tcdr          ; lee carácter del teclado
        cmpi.b #\$0f,d4   ; compara Fh con D4 (escape)
        beq phora        ; si son iguales, brinca a phora:
        cmpi.b #\$02,d4   ; compara 2h con D4
        bgt phd1         ; si d4>2h, brinco a phd1:
        move.b d4,d0      ; coloca D4 en D0
        mulu.w #\$0a,d0   ; multiplica D0=DO*10 (horas : 00, 10 ó 20)
        move.b d4,d7     ; coloca D4 en D7
        bsr lcdddd       ; coloca número binario <10 en display
                          ; SEGUNDO DIGITO (0 a 9 SI EL PRIMERO ES 0 ó 1)
                          ; (0 a 3 SI EL PRIMERO ES 2)

        bsr lcdw         ; verifica el estado del display
        move.b #\$81,$f2000 ; posiciona cursor en 01
phd2:  bsr tcdr          ; lee carácter del teclado
        cmpi.b #\$0f,d4   ; compara Fh con D4 (escape)
        beq phora        ; si son iguales, brinca a phora:
        cmpi.b #\$14,d0   ; compara 14h con D0 (20 horas)
        beq phd2e        ; si son iguales, brinco a phd2e:
        cmpi.b #\$09,d4   ; compara 9h con D4
        bgt phd2         ; si D4>9h, brinca phd2:
        bra phd2c         ; brinca a phd2c:
phd2e: cmpi.b #\$03,d4   ; compara 3h con D4
        bgt phd2         ; si D4>3h, brinca phd2:
phd2c: add.b d4,d0       ; suma D4 en D0 (horas totales)
        move.b d0,d5     ; coloca D0 en D5
        rol.l #\$8,d5    ; 8 corrimientos a la izquierda en D5 (encapsula)
        move.b d4,d7     ; coloca D4 en D7
        bsr lcdddd       ; coloca número binario <10 en display
                          ; TERCER DIGITO (0 a 5) MINUTOS
                          ; borra registro D0
        clr.l d0         ; borra registro D0
        bsr lcdw         ; verifica el estado del display
        move.b #\$83,$f2000 ; posiciona cursor en 03
phd3:  bsr tcdr          ; lee carácter del teclado
        cmpi.b #\$0f,d4   ; compara Fh con D4 (escape)
        beq phora        ; si son iguales, brinca a phora:
        cmpi.b #\$05,d4   ; compara 5h con D4
        bgt phd3         ; si D4>5h, brinco a phd3:
        move.b d4,d0      ; coloca D4 en D0
        mulu.w #\$000a,d0 ; multiplica D0=DO*10 (minutos 0,10,20,30,40 ó 50)
        move.b d4,d7     ; coloca D4 en D7
        bsr lcdddd       ; coloca número binario <10 en display
                          ; CUARTO DIGITO (0 a 9)
                          ; verifica estado del display
        bsr lcdw         ; posiciona cursor en 04
phd4:  move.b #\$84,$f2000 ; posiciona cursor en 04
        bsr tcdr          ; lee carácter del teclado
        cmpi.b #\$0f,d4   ; compara Fh con D4 (escape)
        beq phora        ; si son iguales, brinca a phora:
        cmpi.b #\$09,d4   ; compara 9h con D4
        bgt phd4         ; si D4>9h, brinco a phd4:
        add.b d4,d0       ; suma D4 en D0 (minutos totales)
        move.b d0,d5     ; coloca D0 en D5
        rol.l #\$8,d5    ; 8 corrimientos a la izquierdo en D5 (encapsula)
        move.b d4,d7     ; coloca D4 en D7
        bsr lcdddd       ; coloca número binario <10 en display
                          ; QUINTO DIGITO (0 a 5) SEGUNDOS
                          ; borra registro D0
        clr.l d0         ; borra registro D0
        bsr lcdw         ; verifica estado del display

```

```

phd5:  move.b #$86,$f2000 ; posiciona cursor en 06
      bsr tcdr           ; lee carácter del teclado
      cmpi.b #$0f,d4    ; compara Fh con D4 (escape)
      beq phora        ; si son iguales, brinca a phora:
      cmpi.b #$05,d4    ; compara 5h con D4
      bgt phd5         ; si D4>5h, brinco a phd5:
      move.b d4,d0      ; coloca D4 en D0
      mulu.w #$000a,d0  ; multiplica D0=D0*10 (seg. 0,10,20,30,40 ó 50)
      move.b d4,d7      ; coloca D4 en D7
      bsr lcddd        ; coloca número binario <10 en display
                        ; SEXTO DIGITO, (0 a 9)
      bsr lcdw         ; verifica estado del display
phd6:  move.b #$87,$f2000 ; posiciona cursor en 07
      bsr tcdr           ; lee carácter del teclado
      cmpi.b #$0f,d4    ; compara Fh con D4 (escape)
      beq phora        ; si son iguales, brinca a phora:
      cmpi.b #$09,d4    ; compara 9h con D4
      bgt phd6         ; si D4>9h, brinco a phd6:
      add.b d4,d0       ; suma D4 en D0 (segundos totales)
      move.b d0,d5      ; coloca D0 en D5
      rcl.l #$8,d5      ; 8 corrimientos a la izquierda en D5 (encapsula)
      move.b d4,d7      ; coloca D4 en D7
      bsr lcddd        ; coloca número binario <10 en display
      bsr lcdcof       ; apaga cursor del display
      ; *****
      ; DESPLIEGUE DE CARACTERES "00:00:00"
      ; UTILIZA REGISTROS
      ; D0 : contador de 5h hasta 0h
      ; *****
      clr.l d0          ; borra registro D0
      bsr lcdw         ; verifica estado del display
      move.b #$c0,$f2000 ; posiciona cursor en 40
      bsr lcdw         ; verifica estado del display
      move.b #$7e,$f2001 ; transfiere carácter ASCII "flecha derecha"
      move.b #$05,d0    ; para D0 = 5h hasta 0h decrementando en 1h
dcar:  bsr lcdw         ; verifica estado del display
      move.b #$50,$f2001 ; transfiere carácter ASCII "espacio"
      dbf d0,dcar       ; siguiente D0
      bsr lcdw         ; verifica estado del display
      move.b #$25,$f2001 ; transfiere carácter ASCII "porcentaje"
      ; *****
      ; MANEJO DE INTERRUPCIONES
      ; UTILIZA REGISTROS
      ; i : pase de parámetros a rutina TCDR
      ; D6 : contador de muestras por segundo
      ; A0 : pase de parámetros a rutina LCDSTR
      ; A5 : dirección de almacenamiento de datos
      ; adquiridos
      ; *****
      clr.l d6         ; limpia registro D6
      move.l #$00002000,a5 ; coloca 2000h en A5
      bsr aciaw       ; verifica estado del ACIA
      move.b #$a5,$f4001 ; transmite carácter de sincronización
      bsr aciaw       ; verifica estado del ACIA
      move.b #$a5,$f4001 ; transmite carácter de sincronización
      bsr aciaw       ; verifica estado del ACIA
      move.b #$aa,$f4001 ; transmite carácter de sincronización
      andi.w #$f8ff,ar ; habilita interrupciones (nivel 0)

```

```

m0int: nop ; no operación
        stop #S2000 ; estado de paro (EN ESPERA DE INTERRUPCION)
        cmpa.l #S00003c20,a5 ; compara 3C20 con D5 (tope de pila)
        beq m0m1l ; si son iguales, brinca a m0m1l: (salida)
        btst.b #S7,$f0003 ; prueba la bandera IRQB1 (tecla oprimida)
        beq m0int ; si IRQB1 flag = 0, brinca a m0int:
                ; si IRQB1 flag = 1, continúa
        move.b $f0002,d4 ; coloca ($f0002) en D4, lee carácter del teclado
        andi.b #S0f,d4 ; filtra los 4 bits menos significativos de D4
        cmpi.b #S0f,d4 ; compara Fh con D4 (escape)
        bne m0int ; si son diferentes, brinco a m0int:
        move.w #S2700,sr ; ** deshabilita interrupciones, modo supervisor
        move.b #Sf0,$f0002 ; ** apaga señales luminosas
        bra menu ; ** brinca a menu:
m0m1l: move.w #S2700,sr ; ** deshabilita interrupciones, modo supervisor
        move.b #Sf0,$f0002 ; ** apaga señales luminosas
        move.l #S00000580,a0 ; ** dir. inicial de cadena "MEMORIA LLENA"
        bsr lcdstr ; ** coloca cadena en display
        bsr lcdw ; ** verifica estado del display
        move.b #Sc7,$f2000 ; ** posiciona cursor en 47
        bsr lcdcon ; ** prende cursor del display
        bsr tcdr ; ** espera a que se presione una tecla
        bsr lcdcof ; ** apaga cursor del display
        bra menu ; ** brinca a menu:
; .....
; MENU1
; MODO DE COMUNICACION SERIAL CON COMPUTADORAS PERSONALES
; UTILIZA REGISTROS
; A0 : pase de parámetros a rutina LCDSTR
; A2 : dir. de memoria RAM que se transmite
; : contador de 2000h hasta registro A5
; A5 : última dir. de memoria RAM que se transmite
; .....
menu1: move.l #S00000470,a0 ; dir. inicial de cadena "MODO DE COM:"
        bsr lcdstr ; coloca cadena en display
        bsr delay1 ; retardo de 1 segundo
        cmpa.l #S00002000,a5 ; compara 2000h con A5
        beq mind ; si son iguales, brinca a mind:
        move.l #S000005c0,a0 ; dir. inicial de cadena "PREPARE PC"
        bsr lcdstr ; coloca cadena en display
        bsr lcdw ; verifica estado del display
        move.b #Sc7,$f2000 ; posiciona cursor en 47
        bsr lcdcon ; prende cursor del display
        bsr tcdr ; espera a que se presione una tecla
        bsr lcdcof ; apaga cursor del display
        move.l #S000005B0,a0 ; dir. inicial de cadena "TRANSMITIENDO"
        bsr lcdstr ; coloca cadena en display
        bsr aciaw ; verifica estado del ACIA
        move.b #Sa5,$f4001 ; transmite carácter de sincronización
        bsr aciaw ; verifica estado del ACIA
        move.b #Sa5,$f4001 ; transmite carácter de sincronización
        bsr aciaw ; verifica estado del ACIA
        move.b #Saa,$f4001 ; transmite carácter de sincronización
        move.l #S00002000,a2 ; coloca 2000h en A2, dir. inicial de memoria RAM
        bsr aciaw ; verifica estado del ACIA
        move.b (a2)+,$f4001 ; transmite dato
        cmpa.l a2,a5 ; compara A2 con A5
        bne misc ; si son diferentes, brinca a misc:

```

```

move.l #$000005a0,a0 ; dir. inicial de cadena "FIN DE COM:"
bsr lcdstr           ; coloca cadena en display
bsr lcdw            ; verifica estado del display
move.b #$c7,$f2000 ; posiciona cursor en 47
bsr lcdcon         ; prende cursor del display
bsr tcdr           ; espera a que se presione una tecla
bsr lcdcof         ; apaga cursor del display
bra menu          ; brinca a menu:
m1nd: move.l #$000005d0,a0 ; dir. inicial de cadena "NO HAY DATOS"
bsr lcdstr           ; coloca cadena en display
bsr lcdw            ; verifica estado del display
move.b #$c7,$f2000 ; posiciona cursor en 47
bsr lcdcon         ; prende cursor del display
bsr tcdr           ; espera a que se presione una tecla
bsr lcdcof         ; apaga cursor del display
bra menu          ; brinca a menu:
; .....
; MENU2
; SERVICIO DE PRUEBAS AL SISTEMA
; .....
menu2: move.l #$00000480,a0 ; dirección inicial de cadena "MODO DE PRUEBAS"
bsr lcdstr           ; coloca cadena en display
bsr delay1          ; retardo de 1 segundo
; .....
; PRUEBA AL DISPLAY Y SEÑALES LUMINOSAS
; UTILIZA REGISTROS
; A0 : pase de parámetros a rutina LCDSTR
; .....
move.l #$00000400,a0 ; dirección inicial de cadena "
bsr lcdstr           ; coloca cadena en display
move.b #$00,$f0002 ; prende señales luminosas
bsr delay1          ; retardo de 1 segundo
bsr delay1          ; retardo de 1 segundo
move.b #$f0,$f0002 ; apaga señales luminosas
; .....
; PRUEBA AL TECLADO
; UTILIZA REGISTROS
; D0 : contador de Fh hasta 0 (cada tecla)
; D4 : pase de parámetros a rutina TCDR
; D7 : pase de parámetros a rutina LCDCHR
; A0 : pase de parámetros a rutina LCDSTR
; .....
clr.l d0            ; limpia registro D0
bsr lcdcon         ; prende cursor del display
move.w #$000f,d0   ; para D0 = Fh hasta 0h decrecentando en 1h
tsttcd:move.l #$000004a0,a0 ; dir. inicial de cadena "PR" "ONE"
bsr lcdstr           ; coloca cadena en display
bsr lcdw            ; verifica el estado del display
move.b #$c1,$f2000 ; posiciona cursor en 41
move.b d0,d7       ; coloca D0 en D7
cmpi.b #$a,d7      ; compara Ah con D7
blt tstts          ; si D7 < Ah, brinca a tstts:
addi.b #$07,d7     ; suma 07h a D7 (convierte a ASCII)
tstts: addi.b #$30,d7 ; suma 30h a D7 (convierte a ASCII)
bsr lcdchr         ; coloca carácter ASCII en display
bsr lcdw            ; verifica estado del display
move.b #$c1,$f2000 ; posiciona cursor en 41
bsr tcdr           ; lee carácter del teclado

```

```

cmp.b d0,d4 ; compara D0 con D4
beq tsttb ; si son iguales, brinca a tsttb:
bsr lcdcof ; apaga cursor del display
move.l #$00004b0,a0 ; dir. inicial de cadena "EN MAL ESTADO"
bsr lcdstr ; coloca cadena en display
bsr delay1 ; retardo de 1 segundo
bsr lcdcon ; prende cursor del display
tsttb: dbf d0,tsttc ; siguiente D0
bsr lcdcof ; apaga el cursor del display
;
; *****
; PRUEBA AL ADC
; UTILIZA REGISTROS
; D0 : dato leído del ADC
; D1 : contador de 4h hasta 0h
; D4 : pase de parámetros a rutina TCDR
; D7 : pase de parámetros a rutina LCDCHR
; A0 : pase de parámetros a rutina LCDSTR
; *****

clr.l d1 ; limpia registro D1
move.l #$00004c0,a0 ; dir. inicial de la cadena "00.000% FIN=ESC"
bsr lcdstr ; coloca cadena en display
move.b $f0002,d4 ; limpia bandera IRQB1 (tecla oprimida)
tstadc:nop ; GENERA SEÑAL DE INICIO DE CONVERSION ADC
move.b #$34,$f0001 ; línea CA2 puesta a "0"
move.b #$3c,$f0001 ; línea CA2 puesta a "-1"
tstcnv:nop ; ESPERA SEÑAL DE FINAL DE CONVERSION ADC
bstb.b #$7,$f0001 ; prueba bandera IRQA1 (final de conversión)
beq tstcnv ; si IRQA1 flag = 0, brinca a tstcnv:
; si IRQA1 flag = 1, continúa

clr.l d0 ; limpia registro D0
move.b $f0000,d0 ; coloca (F0000h) en D0, LEER DATO DEL ADC
asl.w #$1,d0 ; l corrimiento aritmético a la izquierda
; (multiplica por 2)

bsr lcdw ; verifica el estado del display
move.b #$80,$f2000 ; posición cursor en 00
move.w #$0004,d1 ; para D1 = 4h hasta 0h decrementando en 1h
tstcd: cmpi.w #$0002,d1 ; compara 2h con D1
bne tstcp ; si son diferentes, brinca a tstcp:
move.b #$2e,d7 ; coloca 2Eh en D7, carácter ASCII "punto"
bsr lcdchr ; coloca carácter en display
tstcp: divu.w #$0033,d0 ; divide D0=D0/51
move.b d0,d7 ; coloca D0 en D7, COCIENTE
bsr lcddd ; coloca número binario <10 en display
swap d0 ; intercambia palabras en D0
andi.l #$0000ffff,d0 ; filtra palabra baja en D0, RESIDUO
mulu.w #$000a,d0 ; multiplica D0=D0*10
dbf d1,tstcd ; siguiente D1
bstb.b #$7,$f0003 ; prueba bandera IRQB1 (tecla oprimida)
beq tstadc ; si IRQB1 flag = 0, brinca a tstadc:
; si IRQB1 flag = 1, continúa
move.b $f0002,d4 ; limpia bandera IRQB1 (tecla oprimida)
; *****
; PRUEBA DE COMUNICACION SERIAL CON PC
; UTILIZA REGISTROS
; D0 : contador de 2Dh hasta 0h
; D1 : contador de 13h hasta 0h
; D4 : pase de parámetros a rutina TCDR

```



```

; A0 : pase de parámetros a rutina LCDSTR
; dir. inicial de la cadena a transmitir
; dir. de cada carácter transmitido
; .....
clr.l d0 ; borra registro D0
clr.l d1 ; borra registro D1
move.l #$000004d0,a0 ; dir. inicial de la cadena "PRUEBA COM:"
bsr lcdstr ; coloca cadena en display
bsr delayl ; retardo de 1 segundo
move.l #$000005c0,a0 ; dir. inicial de la cadena "PREPARE PC"
bsr lcdstr ; coloca cadena en display
bsr lcdw ; verifica estado del display
move.b #$c7,$f2000 ; posiciona cursor en 47
bsr lcdcon ; prende cursor del display
bsr tcdr ; espera a que se presione una tecla
bsr lcdcof ; apaga cursor del display
move.l #$000005b0,a0 ; dir. inicial de la cadena "TRANSMITIENDO"
bsr lcdstr ; coloca cadena en display
bsr aciaw ; verifica estado del ACIA
move.b #$a5,$f4001 ; transmite carácter de sincronización
bsr aciaw ; verifica estado del ACIA
move.b #$a5,$f4001 ; transmite carácter de sincronización
bsr aciaw ; verifica estado del ACIA
move.b #$aa,$f4001 ; transmite carácter de sincronización
move.w #$0013,d1 ; para D1 = 13h hasta 0h decrementando en 1h
tstcom:move.l #$00000550,a0 ; dir. inicial de la cadena
; "SISTEMA PARA DETECCION DE GASES COMBUSTIBLES\n"
move.w #$002d,d0 ; para D0 = 2Dh hasta 0h decrementando en 1h
tstcw: bsr aciaw ; verifica estado del ACIA
move.b (a0)+,$f4001 ; coloca (A0) en (F4001h)
; incrementa A0, dirección del siguiente carácter
; siguiente D0
dbf d0,tstcw ; siguiente D1
dbf d1,tstcom ; retardo de 1 segundo
bsr delayl ; dir. inicial de la cadena "FIN DE COM:"
move.l #$000005a0,a0 ; coloca cadena en display
bsr lcdstr ; verifica estado del display
bsr lcdw ; posiciona cursor en 47
move.b #$c7,$f2000 ; prende cursor de display
bsr lcdcon ; espera a que se presione una tecla
bsr tcdr ; apaga cursor del display
bsr lcdcof ; brinca a menu
bra menu

```

```

org $1000 ; dirección inicial de la rutina de interrupción;
; .....
; RUTINA PARA ACTUALIZAR EL RELOJ DEL SISTEMA
; UTILIZA REGISTROS
; D2 : vector de configuración
; D5 : HORAS:MINUTOS:SEGUNDOS:CENTESIMAS
; D6 : contador de muestras por segundo
; D7 : pase de parámetros con la rutina LCDBYT
; .....
addq.b #$1,d5 ; incrementa 1/100 segundo (CONF. POR HARDWARE)
cmp.l.b #$64,d5 ; si (d5 <> 1 segundo) entonces
bne sigcs ; brinca a sigcs:
move.b #$00,d5 ; inicia cuenta de 1/100 segundo
ror.l #$8,d5 ; 8 corrimientos a la derecha en D5 (desencapsula)

```

```

addq.b #\$1,d5 ; incrementa 1 segundo
cmpi.b #\$3c,d5 ; si (d5 <> 1 minuto) entonces
bne sigs ;
move.b #\$00,d5 ; inicia cuenta de segundos brinca a sigs:
ror.l #\$8,d5 ; 8 corrimientos a la derecha en D5 (desencapsula)
addq.b #\$1,d5 ; incrementa 1 minuto
cmpi.b #\$3c,d5 ; si (d5 <> 1 hora) entonces
bne sigm ; brinca a sigm:
move.b #\$00,d5 ; inicia cuenta de minutos
ror.l #\$8,d5 ; 8 corrimientos a la derecha en D5 (desencapsula)
addq.b #\$1,d5 ; incrementa 1 hora
cmpi.b #\$18,d5 ; si (d5 <> 1 día) entonces
bne sigh ; brinca a sigh:
move.b #\$00,d5 ; inicia cuenta de horas
nop ;
sigh: ; SIGUIENTE HORA
brr lcdw ; verifica el estado del display
move.b #\$80,$f2000 ; posiciona cursor en 00
move.b d5,d7 ; coloca D5 en D7
brr lcdbyt ; coloca un número binario <100 en el display
rol.l #\$8,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
sigm: ; SIGUIENTE MINUTO
nop ;
brr lcdw ; verifica el estado del display
move.b #\$83,$f2000 ; posiciona cursor en 03
move.b d5,d7 ; coloca D5 en D7
brr lcdbyt ; coloca un número binario <100 en el display
rol.l #\$8,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
sigs: ; SIGUIENTE SEGUNDO
nop ;
brr lcdw ; verifica el estado del display
move.b #\$86,$f2000 ; posiciona cursor en 06
move.b d5,d7 ; coloca D5 en D7
brr lcdbyt ; coloca un número binario <100 en el display
rol.l #\$8,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
sigcs: ; no operación
nop ;
; *****
; LLAMA A RUTINA DE MUESTREO DE DATOS
; *****
addq.b #\$1,d6 ; incrementa 1/100 segundo (CONF. POR HARDWARE)
cmp.b d2,d6 ; compara D2 con D6
bne nomues ; si son diferentes, brinca a nomues:
clr.l d6 ; limpia registro D6
brr muestra ; subrutina muestra
; *****
; ALMACENA INFORMACION
; *****
swap d2 ; intercambia palabras en D2
cmpa.l #\$00000500,a3 ; ¿SE REGISTRA DATO?
bne noalma ; compara 500h con A3
; si son diferentes, brinca a noalma:
; 500h = si almacena, 510h= no almacena
; ¿EL DATO ES MAYOR QUE LA FRONTERA?
cmp.w d2,d3 ; compara D2 con D3
blt noalma ; si D3<D2, brinca a noalma:
cmpa.l #\$00000530,a4 ; compara 530h con A4
bne almem ; si son diferentes, brinca a almem:
; 520h = memoria, 530h puerto serial
brr aciaw ; ENVIA DATOS POR EL PUERTO SERIAL
move.b #\$aa,$f4001 ; verifica estado del ACIA
; transmite carácter de sincronización

```

```

rol.l #58,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
bsr aciaw ; verifica estado del ACIA
move.b d5,$f4001 ; transmite horas
rol.l #58,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
bsr aciaw ; verifica estado del ACIA
move.b d5,$f4001 ; transmite minutos
rol.l #58,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
bsr aciaw ; verifica estado del ACIA
move.b d5,$f4001 ; transmite segundos
rol.l #58,d5 ; 8 corrimientos a la izquierda en D5 (encapsula)
bsr aciaw ; verifica estado del ACIA
move.b d5,$f4001 ; transmite centésimas de segundo
bsr aciaw ; verifica estado del ACIA
move.b d3,$f4001 ; transmite dato muestreado del ADC
bra noalma ; brinca a noalma:
almem: nop ; ALMACENA DATOS EN MEMORIA
move.l d5,(a5)+ ; coloca D5 en (A5) (hora)
; incrementa A5
move.w d3,(a5)+ ; coloca D3 en (a5) (dato muestreado)
; incrementa A5
noalma: swap d2 ; intercambia palabras en D2
; .....,.....
nomues: rte ; retorno de excepción
; .....,.....
; RUTINA DE MUESTREO DE DATOS
; UTILIZA REGISTROS
; D0 ; dato leído del ADC, lo convierte en ASCII
; D1 ; contador de 4h hasta 0h
; D2 ; vector de configuración
; D3 ; dato leído del ADC
; D7 ; pase de parámetros a rutina LCDCHR
; ; pase de parámetros a rutina LCDDD
; .....,.....
muestra: clr.l d3 ; borra registro D3
nop ; GENERA SENAL DE INICIO DE CONVERSION
move.b #$34,$f0001 ; línea CA2 puesta a "0"
move.b #$3c,$f0001 ; línea CA2 puesta a "1"
m0cnv: nop ; ESPERA SENAL DE FINAL DE CONVERSION
btst.b #$7,$f0001 ; prueba el bit más significativo (IRQA1 flag)
beq m0cnv ; F0001h registro de control del display
; si IRQA1 flag=0, brinca a m0cnv:
; si IRQA1 flag=1, continua (final de conversión)
move.b $f0000,d3 ; coloca (F0000h) en D3, LEZ DATO DEL ADC
; .....,.....
; PRENDE SEÑALES LUMINOSAS
; .....,.....
clr.l d0 ; borra registro D0
swap d2 ; intercambia palabras en D2
cmp.w d2,d3 ; compara D2 con D3
blt m0aal ; si D3<D2, brinca a m0aal:
move.b #$e0,d0 ; coloca E0h en (f0002)
; (prende s. de frontera y apaga las demás)
; brinca a m0fal:
bra m0fal ; coloca F0h en (f0002) (apaga señales)
m0aal: move.b #$f0,d0 ; intercambia palabras en D2
m0fal: swap d2 ; compara 40h con D3 (0% a 25%)
cmpi.w #$0040,d3 ; si es menor, brinca a m0ver
blt m0ver ; compara 80h con D3 (25% a 50%)

```

```

        blt m0ama                ; si es menor, brinca a m0ama
                                ; (50% a 100%)
        andi.b #$d0,d0           ; prende señal roja
        bra m0eal                ; brinca a m0eal:
m0ama:  andi.b #$b0,d0           ; prende señal amarilla
        bra m0eal                ; brinca a m0eal:
m0ver:  andi.b #$70,d0           ; prende señal verde
m0eal:  move.b d0,$f0002        ; coloca D2 en ($f0002)
                                ; .....
                                ; COLOCA MUESTRA EN DISPLAY
                                ; .....
        clr.l d0                 ; limpia registro D0
        clr.l d1                 ; limpia registro D1
        move.b d3,d0             ; coloca D3 en D0
        asl.w #$1,d0             ; 1 corrimiento aritmético a la izquierda en D0
                                ; (multiplicar por: 2)
        bsr lcdw                 ; verifica el estado del display
        move.b #$c1,$f2000       ; posición cursor en 4l
                                ; para D1 = 4h hasta 0h decrementando en 1h
m0digit: move.b #$0004,d1        ; compara 2h con D1
        bne m0pto                ; si son diferentes, brinca a m0pto:
        move.b #$2e,d7           ; coloca 2Eh en D7, carácter "punto decimal"
        bsr lcdchr              ; coloca carácter en display
m0pto:  divu.w #$0033,d0         ; divide D0=D0/51
        move.b d0,d7             ; coloca D0 en D7, COCIENTE
        bsr lcdddd              ; coloca número binario <10 en display
        swap d0                  ; intercambia palabras en D0
        andi.l #$0000ffff,d0     ; filtra palabra baja en D0, RESIDUO
        mulu.w #$000a,d0         ; multiplica D0=D0*10
        dbf d1,m0digit          ; siguiente D1
        rts                      ; retorno de subrutina

```

```

        org $1c00                ; dirección inicial de rutinas de servicio
; .....
; RUTINAS PARA EL MANEJO DEL TECLADO
; .....
; .....
; RUTINA QUE LEE UN CARACTER DEL TECLADO
; ESPERA A QUE SE PRESIONE UNA TECLA
; RUTINA TCDR (TeCLado_Read)
; REGISTRO D4, CARACTER LEIDO
; .....
tcdr:  move.b $f0002,d4          ; limpia bandera IRQB1 (tecla oprimida)
        clr.l d4                 ; limpia registro D4
tcdw:  btst.b #$7,$f0003        ; prueba el bit más significativo (IRQB1 flag)
        beq tcdw                 ; si IRQB1 flag = 0, brinca a tcdw:
        move.b $f0002,d4         ; lee carácter del teclado
        andi.b #$0f,d4           ; filtra los cuatro bits menos significativos de D4
        rts                      ; retorno de subrutina
; .....
; RUTINAS PARA EL MANEJO DEL DISPLAY
; .....
; .....
; RUTINA QUE VERIFICA EL ESTADO DEL DISPLAY
; RUTINA LCDW (LCD_Wait)
; .....

```

```

lcdw:  btst.b #$7,$f2000 ; prueba el bit más significativo (busy flag)
      bne lcdw           ; si busy flag=1, brinca a lcdw: (display ocupado)
      rts               ; si busy flag=0, continúa (display libre)
      ; retorno de subrutina
      ; *****
      ; RUTINA QUE ENCIENDE EL CURSOR
      ; RUTINA LCD_Cursor_ON
      ; *****

lcdcon: bsr lcdw         ; verifica el estado del display
      move.b #$0d,$f2000 ; comando "display ON-OFF control" AND671 (blink)
      ; display encendido: D=1
      ; cursor apagado: C=0
      ; carácter con parpadeo: B=1
      ; retorno de subrutina
      rts               ; *****
      ; RUTINA QUE APAGA EL CURSOR
      ; RUTINA LCD_Cursor_OFF
      ; *****

lcdcof: bsr lcdw         ; verifica el estado del display
      move.b #$0c,$f2000 ; comando "Display ON-OFF control" del AND671
      ; display encendido: D=1
      ; cursor apagado: C=0
      ; carácter sin parpadeo: B=0
      ; retorno de subrutina
      rts               ; *****
      ; RUTINA QUE BORRA EL DISPLAY
      ; RUTINA LCDCLR (LCD_CLeAr)
      ; *****

lcdclr: bsr lcdw         ; verifica el estado del display
      move.b #$01,$f2000 ; comando "clear display" del AND671
      ; retorno de subrutina
      rts               ; *****
      ; RUTINA QUE TRANSFIERE UN CARACTER AL DISPLAY
      ; RUTINA LCDCHR (LCD_CHaR)
      ; REGISTRO D7, CARACTER ASCII
      ; *****

lcdchr: bsr lcdw         ; verifica el estado del display
      move.b d7,$f2001   ; transfiere el carácter ASCII contenido
      ; en el registro D7 al display
      ; retorno de subrutina
      rts               ; *****
      ; RUTINA QUE TRANSFIERE UN NUMERO <10 AL DISPLAY
      ; RUTINA LCDDD (LCD_Decimal_Digit)
      ; REGISTRO D7, NUMERO BINARIO MENOR QUE 10
      ; *****

lcdddd: move.l d7,-(a7)  ; push D7
      addi.b #$30,d7     ; convierte el número en carácter ASCII
      bsr lcdchr         ; coloca carácter ASCII en display
      move.l (a7)+,d7    ; pop D7
      rts               ; retorno de subrutina
      ; *****
      ; RUTINA QUE TRANSFIERE UN NUMERO <100 AL DISPLAY
      ; RUTINA LCDBYT (LCD_BYTe)
      ; REGISTRO D7, NUMERO BINARIO MENOR QUE 100
      ; *****

lcdbyt: move.l d7,-(a7)  ; push D7
      andi.l #$000000ff,d7 ; filtra el número binario
      divu.w #$000a,d7   ; divide entre 10 (word residuo:word cociente)

```

```

    addi.w #\$30,d7      ; convierte el cociente en carácter ASCII
    bar lcdchr          ; coloca carácter ASCII en display
    swap d7            ; intercambia cociente y residuo
    addi.w #\$30,d7      ; convierte el residuo en carácter ASCII
    bar lcdchr          ; coloca carácter ASCII en display
    move.l (a7)+,d7     ; pop D7
    rts                ; retorno de subrutina
; .....
```

```

; RUTINA QUE TRANSFIERE UNA CADENA AL DISPLAY
; RUTINA LCDSTR (LCD_String)
; REGISTRO A0, DIRECCION DE INICIO DE LA CADENA
; 00h, CARACTER DE FINAL DE CADENA
; .....
```

```

lcdstr:move.l d0,-(a7) ; push D0
        move.l a0,-(a7) ; push A0
        bar lcdclr      ; borra display, inicio del primer renglón
        move.w #\$0007,d0 ; para D0 = 7h hasta 0h decrementando en lh
lcd11:  tst.b (a0)        ; verifica final de cadena
        beq lcdest      ; si encuentra final de cadena brinca a lcdest;
        bar lcdw         ; verifica el estado del display
        move.b (a0)+,$f2001 ; transfiere el carácter ASCII almacenado
; en la dirección A0 al display,
; y luego A0 apunta al siguiente carácter
        dbf d0,lcd11     ; siguiente D0
        bar lcdw         ; verifica el estado del display
        move.b #\$c0,$f2000 ; comando "set DDRAM address" del AND671
; inicio del segundo renglón
        move.w #\$0007,d0 ; para D0 = 7h hasta 0h decrementando en lh
lcd12:  tst.b (a0)        ; verifica final de cadena
        beq lcdest      ; si encuentra final de cadena brinca a lcdest;
        bar lcdw         ; verifica el estado del display
        move.b (a0)+,$f2001 ; transfiere el carácter ASCII almacenado
; en la dirección A0 al display,
; y luego A0 apunta al siguiente carácter
        dbf d0,lcd12     ; siguiente D0
lcdest:move.l (a7)+,a0   ; pop A0
        move.l (a7)+,d0   ; pop D0
        rts                ; retorno de subrutina
; .....
```

```

; RUTINAS PARA EL MANEJO DEL ACIA 6850
; .....
```

```

; RUTINA QUE VERIFICA EL ESTADO DEL REGISTRO
; DE TRANSMISION DE DATOS DEL ACIA
; RUTINA ACIAW (ACIA_Wait)
; .....
```

```

aciaw: btst.b #\$1,$f4000 ; prueba el bit 2 del registro de estado del ACIA
; (Transmit Data Register Empty)
        beq aciaw        ; si TRDE=0, brinca a aciaw;
; si TRDE=1, continúa (final de transmisión)
        rts                ; retorno de subrutina
; .....
```

```

; RUTINAS DE RETARDO
; .....
```

```

; RUTINA DE RETARDO -0.1 SEGUNDOS
; RUTINA DELAY
; .....
```

```

delay: move.l d0,-(a7)      ; push D0
      move.w #$ad9c,d0     ; para D0 = AD9Ch hasta 0h decrementando en 1h
delw:  nop                 ;
      dbf d0,delw          ; siguiente D0, 4 ciclos de reloj
      move.l (a7)+,d0      ; pop D0
      rts                 ; retorno de subrutina
      ; *****
      ; RUTINA DE RETARDO -1 SEGUNDO
      ; RUTINA DELAY1
      ; *****
delay1: move.l d0,-(a7)    ; push D0
      move.l d1,-(a7)     ; push D1
      move.w #$0009,d1     ; para D1 = 9h hasta 0h decrementando en 1h
dellw2: move.w #$ad9c,d0  ; para D0 = AD9C hasta 0h decrementando en 1h
dellw1: nop                ;
      dbf d0,dellw1       ; siguiente D0, 4 ciclos de reloj
      dbf d1,dellw2       ; siguiente D1
      move.l (a7)+,d1      ; pop D1
      move.l (a7)+,d0      ; pop D0
      rts                 ; retorno de subrutina

```

CAPÍTULO VIII.

PRUEBAS AL SISTEMA PARA DETECCIÓN DE GASES COMBUSTIBLES.

I. Introducción.

El *Sistema para Detección de Gases Combustibles* fue objeto de diversas pruebas a lo largo de su diseño, a fin de verificar su correcto funcionamiento. A continuación se presenta un resumen de las pruebas más importantes.

II. Cuidados en la realización del "hardware".

Para minimizar la posibilidad de fallas durante el alambrado de cualquier sistema que involucre circuitos integrados, se deben tomar en cuenta ciertos cuidados.

Mantenga todos los alambres lo más corto posible (los alambres largos capturan ruido). Conecte en todos los circuitos integrados, capacitores de "bypass" de $0.1\mu\text{F}$, de V_{cc} a tierra. Elimine los componentes innecesarios, para evitar cortos circuitos. Conecte V_{cc} y tierra a todos los circuitos integrados, antes de realizar cualquier otra conexión. Fije a V_{cc} todas las entradas que no utilice de los circuitos integrados TTL, a través de resistencias de "pull up" de $4.7\text{ k}\Omega$ ó $3.3\text{ k}\Omega$. Haga una copia esquemática de todas las conexiones, cuando las esté efectuando. Utilice un ohmmetro para verificar la continuidad en cada conexión hecha. Primero alambre el reloj de los circuitos integrados; si el reloj no trabaja, el resto del sistema tampoco lo hará. Verifique que todos los circuitos integrados tengan una polarización adecuada antes de empezar a resolver cualquier problema; luego, verifique las señales de reloj.

III. Comunicación con el puerto paralelo PIA 6821.

El siguiente programa verifica la comunicación entre el microprocesador MC68008 y el puerto paralelo PIA 6821. El programa lee un dato del puerto A del PIA y lo transfiere sin cambios al puerto B. De este modo, se prueba sencillamente que la lógica de decodificación habilita oportunamente al PIA cuando el microprocesador intenta accederlo; que el PIA responde apropiadamente ante la petición de operaciones de lectura/escritura por parte del microprocesador; y que su conexión con el bus de datos del microprocesador opera correctamente. Para esta prueba se considera que los puertos del PIA 6821 no se encuentran conectados a ningún otro dispositivo.

```

; . . . . .
; PROGRAMA QUE LEE UN DATO DEL PUERTO A Y LO ESCRIBE EN EL PUERTO B,
; AMBOS DEL PIA 6821
; . . . . .
;
; DIRECCIONES:
; PUERTO A DEL PIA, REGISTRO DE DATOS                F0000h
; PUERTO A DEL PIA, REGISTRO DE CONTROL             F0001h
; PUERTO B DEL PIA, REGISTRO DE DATOS                F0002h
; PUERTO B DEL PIA, REGISTRO DE CONTROL             F0003h
;
; REGISTROS UTILIZADOS:
;   D0                : dato leído del puerto A del PIA
;
; . . . . .
;   org $800          ; dirección de inicio del programa
;   move.b #$00,$f0001 ; selecciona registro de dirección de datos del pto. A
;   move.b #$00,$f0003 ; selecciona registro de dirección de datos del pto. B
;   move.b #$00,$f0000 ; configura puerto A como entrada
;   move.b #$ff,$f0002 ; configura puerto B como salida
;   move.b #$04,$f0001 ; selecciona registro de salida del puerto A
;   move.b #$04,$f0003 ; selecciona registro de salida del puerto B
loop:  move.b $f0000,d0 ; lee dato del puerto A
;   move.b d0,$f0002   ; escribe dato en el puerto B
;   bra loop          ; brinca a etiqueta loop:

```

IV. Comunicación con la memoria RAM 6264.

El siguiente programa verifica la comunicación entre el microprocesador MC68008 y la memoria RAM 6264, así como las operaciones de lectura y escritura sobre ésta última. El programa escribe un dato de 8 bits en cada localidad de la memoria RAM, y luego lo lee, haciendo una comparación entre ellos. Si ambos datos son iguales, la localidad de memoria RAM se encuentra en buen estado, y se accesa apropiadamente; de otro modo, el circuito integrado de memoria RAM podría estar defectuoso, o el microprocesador no lo logra acceder. Además, se prueba que la lógica de decodificación habilita oportunamente a la memoria RAM cuando el microprocesador intenta accederla; que la memoria RAM responde apropiadamente ante la petición de operaciones de

lectura/escritura por parte del microprocesador; y que su conexión con el bus de datos del microprocesador opera correctamente. Para esta prueba se considera que el puerto B del PIA 6821 no se encuentra conectado a ningún otro dispositivo.

```

;
;
; PROGRAMAS QUE LEE Y ESCRIBE UN DATO EN CADA LOCALIDAD DE LA MEMORIA RAM
; PRIMERO EL DATO 55h (01010101b) Y LUEGO EL DATO AAh (10101010b)
; SI LA PRUEBA TIENE EXITO DESPLIEGA 11h EN EL PUERTO B DEL PIA
; SI LA PRUEBA NO TIENE EXITO DESPLIEGA 22h EN EL PUERTO B DEL PIA
;
;
; .....
; DIRECCIONES:
; RAM ESTÁTICA                                2000h - 3FFFh
; PUERTO B DEL PIA, REGISTRO DE DATOS          F0002h
; PUERTO B DEL PIA, REGISTRO DE CONTROL       F0003h
;
; REGISTROS UTILIZADOS:
; D0      : contador de 1FFFh a 0h decremantando en 1h
; D1      : dato leído de la memoria RAM
; A0      : dirección de la memoria RAM que se verifica
;          : contador de 2000h a 3FFFh incrementando en 1h
;
; .....
; org $800 ; dirección de inicio del programa
; move.b #$00,$f0003 ; selecciona registro de dirección de datos pto. B
; move.b #$ff,$f0002 ; configura al puerto B como salida
; move.b #$04,$f0003 ; selecciona registro de salida pto. B
; move.l #$00002000,a0 ; coloca 2000h en A0
;          ; dirección inicial de la memoria RAM
loop: move.w #$ffff,d0 ; para D0 = 1FFFh hasta 0h decremantando en 1h
; move.b #$55,(a0) ; coloca 55h en (A0)
; move.b (a0),d1 ; coloca (A0) en D1
; cmpi.b #$55,d1 ; compara 55h con D1
; bne error ; si son diferentes, brinca a la etiqueta error:
; move.b #$aa,(a0) ; coloca AAh en (A0)
; move.b (a0)+,d1 ; coloca (A0) en D1, postincrementa A0
; cmpi.b #$aa,d1 ; compara AAh con D1
; bne error ; si son diferentes, brinca a la etiqueta error:
; dbf d0,loop ; siguiente D0
; move.b #$11,$f0002 ; coloca 11h en (F0002h), EXITO EN LA PRUEBA
; stop #$2000 ; detener operación
error: move.b #$22,$f0002 ; coloca 22h en (F0002h), ERROR EN LA PRUEBA
; stop #$2000 ; detener operación

```

V. Comunicación con el puerto serial ACIA 6850.

El siguiente programa verifica la comunicación entre el microprocesador MC68008 y el puerto serial ACIA 6850. El programa transmite el carácter ASCII de la letra "A" por el puerto serial. Este programa se complementa con otro que recibe dicho carácter por el puerto serial de una computadora personal, mostrándolo en su pantalla. Esta acción se realiza hasta que dicho carácter aparece íntegro en la pantalla de la computadora personal. De este modo, se prueba que la lógica de decodificación habilita oportunamente al ACIA cuando el microprocesador intenta accederlo; que el ACIA responde apropiadamente ante la petición de operaciones de lectura/escritura por parte del microprocesador; y que su conexión con el bus de datos del microprocesador opera correctamente. Además, se demuestra que la comunicación entre el Sistema para Detección de Gases Combustibles y la computadora personal se realiza satisfactoriamente; esto es, que los datos transmitidos hacia la computadora personal no se alteran ni se pierden.

```

; .....
; PROGRAMA QUE TRANSMITE EL CARACTER ASCII DE LA LETRA "A"
; POR EL ACIA 6850
; .....
;
; DIRECCIONES:
; ACIA, REGISTRO DE CONTROL/ESTADO          F4000h
; ACIA, REGISTRO DE TRANSMISION Y RECEPCION DE DATOS  F4001h
;
; REGISTROS UTILIZADOS
; DO                      : estado del ACIA
; .....
org $800                ; dirección de inicio del programa
move.b #$03,$f4000      ; comando "master reset" del ACIA 6850
move.b #$14,$f4000      ; configuración del ACIA 6850
; CR7=0 receive interrupt enable bit (disable)
; CR6=0, CR5=0 transmitter control bits
; (RTS = low; transmitting interrupt disable)
; CR4=1, CR3=0, CR2=1 word select bits
; (8 bits + 1 stop bit)
; CR1=0, CR0=0 counter divide select bits (/1)

```

```

loop:  nop
wait:  move.b $f4000,d0      ; coloca (F4000h) en D0 (estado del ACIA)
      andi.b #$02,d0      ; prueba el bit 2 (Transmit Data Register Empty, TDRE)
      beq wait            ; si TDRE=0, brinca a etiqueta wait:
      ; si TDRE=1, continúa (final de transmisión)
      move.b #$41,$f4001  ; transmite el carácter ASCII de la letra "A"
      bra loop           ; brinca a etiqueta loop:

```

VI. Comunicación con el convertidor analógico-digital ADC0804.

El siguiente programa verifica la conexión propuesta entre el puerto A del PIA 6821 y el convertidor analógico-digital ADC0804, así como las señales para el control de éste último. En el programa, primero se genera la señal de inicio de conversión para el ADC0804; una vez que éste responde con la señal de final de conversión, significa que un dato binario se encuentra disponible en su salida; luego, el programa toma dicho dato del puerto A del PIA y, finalmente, lo envía al puerto B. De esta manera se prueba la conexión del ADC0804 con el puerto A del PIA y la correcta aceptación de los datos de entrada al sistema, ya que estos se muestran inmediatamente en el puerto B del PIA. Además, esta prueba permite revisar los tiempos de espera que el microprocesador debe insertar a fin de permitir la conversión analógica-digital. Para esta prueba se considera que el puerto B del PIA 6821 no se encuentra conectado a ningún otro dispositivo.

```

; . . . . .
;
; PROGRAMA QUE LEE UN DATO DEL CONVERTIDOR ANALOGICO DIGITAL
; CONECTADO AL PUERTO A DEL PIA, Y LO TRANFIERE AL PUERTO B DEL PIA
;
; . . . . .
;
; DIRECCIONES:
; PUERTO A DEL PIA, REGISTRO DE DATOS           F0000h (CONVERTIDOR A/D)
; PUERTO A DEL PIA, REGISTRO DE CONTROL        F0001h (CONVERTIDOR A/D)
; PUERTO B DEL PIA, REGISTRO DE DATOS           F0002h

```

```

; PUERTO B DEL PIA, REGISTRO DE CONTROL          F0003h
;
; PROGRAMACION INICIAL DEL PUERTO A, (38h)
; b7 = 0, bandera de interrupciones IRQA1 (monitorea línea CA1)
; b6 = 0, (obligado)
; b5 = 1, configura línea CA2 como salida
; b4 = 1, control inmediato sobre línea CA2
; b3 = 1, línea CA2 puesta a "1"
; b2 = 0, selecciona registro de dirección de datos
; b1 = 0, bandera de interrupciones IRQA1 puesta a "1"
; por la transición de "1" a "0" en la línea CA1
; b0 = 0, deshabilita la línea de interrupción IRQA
;
; OTROS COMANDOS PARA EL PUERTO A
; 3ch      b2 = 1, selecciona registro de salida
;          b3 = 1, línea CA2 puesta a "1"
; 34h      b2 = 1, selecciona registro de salida
;          b3 = 0, línea CA2 puesta a "0"
;
; REGISTROS UTILIZADOS:
; D0      : estado del puerto A del pia
; D1      : dato leído del puerto A del PIA
; D2      : contador desde 6h hasta 0h decrecentando en 1h
;
; . . . . .
; org $800 ; dirección de inicio del programa
; move.b #$38,$f0001 ; selecciona registro de dirección de datos, pto. A
; move.b #$00,$f0003 ; selecciona registro de dirección de datos, pto. B
; move.b #$00,$f0000 ; configura al puerto A como entrada
; move.b #$ff,$f0002 ; configura al puerto B como salida
; move.b #$1c,$f0001 ; selecciona registro de salida, puerto A
; move.b #$04,$f0003 ; selecciona registro de salida, puerto B
loop:   nop ; SEÑAL DE INICIO DE CONVERSION
; move.b #$34,$f0001 ; línea CA2 puesta a "0"
; move.b #$3c,$f0001 ; línea CA2 puesta a "1"
wait:  nop ; ESPERA SEÑAL DE FINAL DE CONVERSION
; move.b $f0001,d0 ; coloca (F0001h) en D0
; andi.b #$80,d0 ; prueba el bit más significativo (IRQA1 flag)
; beq wait ; si IRQA1 flag = 0, brinca a wait;
; nop ; si IRQA1 flag = 1, continua (final de conversión);
; LEER DE PUERTO A, ESCRIBIR EN PUERTO B
; move.b $f0000,d1 ; coloca (F0000h) en D1
; move.b d1,$f0002 ; coloca D1 en (F0002h)
; RETARDO DE 12.5 microsegundos
; move.w #$0006,d2 ; coloca 6h en D2
; delay: nop ; 4 ciclos de reloj
; dbf d2,delay ; 14 ciclos de reloj
; bra loop ; brinca a loop;

```

VII. Comunicación con la pantalla de cristal líquido AND 671.

El siguiente programa verifica la conexión entre el microprocesador MC68008 y la pantalla de cristal líquido AND 671. El programa proporciona las instrucciones pertinentes al circuito controlador de la pantalla e inserta los tiempos de espera requeridos por éste para realizar correctamente las operaciones de lectura y escritura. Primero, se configura la pantalla. Después se revisa el estado de ésta, a fin de verificar que se encuentre desocupada, para que pueda aceptar los datos que se le envían. Una vez que la pantalla está desocupada, se envían los caracteres ASCII para escribir dos letras "A": una en la primera y otra en la novena posición. De este modo, se prueba que la lógica de decodificación habilita oportunamente al AND 671 cuando el microprocesador intenta accederla; que el AND 671 responde apropiadamente ante la petición de operaciones de lectura/escritura por parte del microprocesador; y que su conexión con el bus de datos del microprocesador opera correctamente. Además, esta prueba permite revisar los tiempos de espera que el microprocesador debe insertar a fin de permitir el procesamiento que el controlador de la pantalla debe hacer sobre los datos, a fin de exhibirlos correctamente.

```

;.....
;
; PROGRAMA QUE DESPLIEGA DOS LETRAS "A",
; UNA EN LA PRIMERA POSICION DEL DISPLAY
; Y OTRA A LA NOVENA POSICION DEL DISPLAY
;
;.....
;
; DISPLAY, REGISTRO DE COMANDOS          F2000h
; DISPLAY, REGISTRO DE DATOS           F2001h
;
; D0  CONTADOR DESDE 1A0Bh HASTA 0h DECREMENTANDO EN 1h
; D1  ESTADO DEL DISPLAY (LIBRE U OCUPADO)
;
;.....
org $800          ; dirección de inicio del programa
                 ; RETARDO DE 15ms (power up del display)
move.w #$1a0b,d0 ; coloca 1A0Bh en D0

```

```

delay: nop                ; 4 ciclos de reloj
      dbf d0,delay        ; 14 ciclos de reloj
      move.b #538,$f2000 ; comando "function set" del AND671
                          ; 8 bits para longitud de datos: DL=1
                          ; configuración AND671: N=1, F=0
      bsr wait           ; verifica estado del display
      move.b #50c,$f2000 ; comando "display ON/OFF control" del AND671
                          ; display encendido: D=1
                          ; cursor apagado: C=0
                          ; caracter sin parpadeo: B=0
      bsr wait           ; verifica estado del display
      move.b #501,$f2000 ; comando "clear display" del AND671
                          ; borra pantalla
      bsr wait           ; verifica estado del display
      move.b #580,$f2000 ; comando "set DD RAM address" del AND671
                          ; se posiciona en la primera posición
      bsr wait           ; verifica estado del display
      move.b #541,$f2001 ; letra A, primera posición
      bsr wait           ; verifica estado del display
      move.b #5c0,$f2000 ; comando "set DD RAM address" del AND671
                          ; se posiciona en la novena posición
      bsr wait           ; verifica estado del display
      move.b #541,$f2001 ; letra A, novena posición
      stop #52000        ; detener
                          ;
wait:  nop                ;
                          ; RUTINA QUE ESPERA A QUE EL DISPLAY TERMINE
                          ; LA EJECUCION DE CUALQUIER COMANDO
                          ;
      move.b $f2000,d1    ; coloca (F2000h) en D1 (estado del display)
      andi.b #580,d1     ; prueba el bit más significativo (busy flag)
      bne wait           ; si busy flag=1, brinca a wait: (display ocupado)
                          ; si busy flag=0, continúa (display libre)
      rts                ; retorno de subrutina

```

VIII. Comunicación con el teclado.

El siguiente programa verifica la conexión propuesta entre el puerto B del PIA 6821 y el teclado, así como las señales de estado de éste último. En el programa, primero verifica la existencia de la señal de estado "tecla oprimida"; esto significa que un dato binario se encuentra disponible en su salida. Luego, el programa toma dicho dato de los bits 0 a 3 del puerto B del PIA y, finalmente, lo envía a los bits 4 a 7 del mismo puerto. De esta manera se prueba la conexión del teclado con el puerto B del PIA, y la correcta aceptación de los datos de entrada al sistema, ya que estos se muestran inmediatamente en el mismo puerto.


```

;*****
;
; PROGRAMA QUE LEE UN DATO DEL TECLADO CONECTADO EN PB0-PB3 DEL PUERTO A
; DEL PIA, Y LO ESCRIBE EN PB4-PB7 DEL PUERTO B DEL PIA
;*****
;
; PUERTO B DEL PIA, REGISTRO DE DATOS      F0002h
; PUERTO B DEL PIA, REGISTRO DE CONTROL  F0003h
;
; PROGRAMACION INICIAL DEL PUERTO B, (3Ah)
; b7 = 0, bandera de interrupciones IRQB1 (monitorea línea CB1)
; b6 = 0, (obligado)
; b5 = 1, configura línea CB2 como salida
; b4 = 1, control inmediato sobre línea CB2
; b3 = 1, línea CB2 puesta a "1"
; b2 = 0, selecciona registro de dirección de datos
; b1 = 1, bandera de interrupciones IRQB1 puesta a "1"
;      por la transición de "0" a "1" en la línea CB1
; b0 = 0, deshabilita la línea de interrupción IRQB
;
; OTROS COMANDOS PARA EL PUERTO B
; 3Eh      b2 = 1, selecciona registro de salida
;          b3 = 1, línea CA2 puesta a "1"
; 36h      b2 = 1, selecciona registro de salida
;          b3 = 0, línea CA2 puesta a "0"
;
;*****
org $800 ; dirección de inicio del programa
move.b #$3a,$f0003 ; selecciona registro de dirección de datos, pto. B
move.b #$f0,$f0002 ; configura bits PB4 a PB7 del pto.B como salida
; configura bits PB0 a PB3 del pto.B como entrada
move.b #$3e,$f0003 ; selecciona registro de salida B
;
loop: nop ; LEE DE PB0-PB3 Y ESCRIBE EN PB4-PB7 DEL PTO.B
move.b $f0002,d0 ; coloca (f0002) en D0
andi.b #$0f,d0 ; los 4 bits menos significativos son de interés
lsl.b #$04,d0 ; 4 corrimientos a la izquierda en D0
move.b d0,$f0002 ; coloca D0 en (f0002)
bra loop ; brinca a loop:

```

CONCLUSIONES Y COMENTARIOS.

El objetivo del presente diseño es la realización de un Sistema Electrónico para Detección de Gases Combustibles capaz de adquirir y almacenar datos en forma digital, y de comunicarse con una computadora personal para un análisis posterior de dichos datos.

El objetivo anterior es posible debido a que los gases combustibles se comportan de una manera semejante por debajo de su límite inferior de explosividad (*LEL*). De tal forma que el transductor catalítico detecta directamente la concentración de un gas combustible en función de este parámetro. Esto permite generalizar el sentido de gases combustibles, es decir, detectar la existencia de un gas combustible sin determinar su identidad.

Por otro lado, el prototipo del sistema de adquisición de datos realizado, se puede considerar como de propósito general; ya que intercambiando el transductor y la programación, puede adquirir datos asociados a otras variables físicas.

En lo que se refiere a las capacidades del prototipo, es posible incrementarlas mediante:

1. Aumento en la capacidad de memoria RAM. El límite de la memoria lo define la capacidad de direccionamiento del microprocesador.
2. Incremento en el número de transductores involucrados. Se pueden multiplexar las señales de salida de varios transductores por un solo puerto, y/o incrementar la cantidad de puertos dedicados a transductores.
3. Disminuir la complejidad del circuito utilizando arreglos lógicos programables (*PAL*), en lugar de lógica combinacional implantada con circuitos discretos.

Ahora bien, el límite de la velocidad de operación del sistema se impone, en primer lugar, por la velocidad de la transmisión serial de datos y, en segundo lugar, por la velocidad del convertidor analógico-digital. Para

incrementar la velocidad del sistema, se podría considerar la transmisión paralela de datos en cortas distancias.

Puesto que la acumulación de gases combustibles constituye una de las principales causas de incendios y explosiones, un sistema integral de protección contra estos incidentes, que detecte su presencia, tiene gran importancia.

Generalmente, un aviso originado con anticipación por un sistema para la detección de gases, proporciona suficiente tiempo para ejecutar las acciones de seguridad necesarias antes de que el peligro aumente, minimizando la posibilidad de una catástrofe. Por tanto, en un sistema para la detección de gases, el énfasis se centra en la prevención.

Finalmente, se recomienda utilizar un sistema para la detección de gases bajo las siguientes circunstancias:

1. Cuando se manejan gases altamente inflamables o explosivos.
2. Cuando se requiere de una respuesta inmediata durante una situación de emergencia.
3. Cuando se requiere de un sistema de protección continuo y automático.
4. Cuando se debe proteger una gran inversión de capital.
5. Cuando la vida humana o su integridad está en riesgo.
6. Cuando un gas puede acumularse en un área inaccesible.
7. Cuando un gas puede acumularse en un área que represente un riesgo para la seguridad.

Un sistema para detectar gases incrementa la capacidad y versatilidad de un sistema de protección contra incendios, ya que, además de la protección correctiva que éste último ofrece, también se puede contar con una protección preventiva, sobre todo si se trata de una zona de alto riesgo de acumulación de gases combustibles.

APÉNDICE A.

ECUACIÓN GENERAL DE LOS GASES. EJEMPLO DE APLICACIÓN.

Considere la *ecuación general de los gases*:

$$V = \frac{\left(\frac{m}{M}\right)RT}{p}$$

1.- **datos:**

$$p = 101\,000 \text{ Pa}$$

$$T = 273 + 21 \text{ °C} = 294 \text{ K}$$

$$m = 9 \text{ kg}$$

$$M = 32 \text{ kg/kmol}$$

$$R = 8\,314 \text{ J/(kmol K)}$$

Condiciones normales de presión.

Condiciones normales de temperatura.

Masa del O_2 .

Peso molecular del O_2 .

Constante universal de los gases.

resultado:

$$V = 6.81 \text{ m}^3$$

Volumen que ocupa el O_2 .

2.- datos:

$$p = 101\,000 \text{ Pa}$$

$$T = 273 + 21 \text{ }^\circ\text{C} = 294 \text{ K}$$

$$m = 53 \text{ kg}$$

$$M = 32 \text{ kg/kmol}$$

$$R = 8\,314 \text{ J/(kmol K)}$$

Condiciones normales de presión.

Condiciones normales de temperatura.

Masa del O_2 .Peso molecular del O_2 .

Constante universal de los gases.

resultado:

$$V = 40.1 \text{ m}^3$$

Volumen que ocupa el O_2 .

APÉNDICE B.

REGISTROS DEL MICROPROCESADOR Y CADENAS DE CARACTERES.

I. Cadenas de caracteres.

A continuación se muestran las direcciones y cadenas de caracteres definidas en la programación del Sistema para Detección de Gases Explosivos.

400h	[REDACTED]
410h	CONFIGURANDOSE
420h	VERIFICANDO RAM
430h	ERROR EN RAM
440h	OPCION (EDC)
450h	OPCION INVALIDA
460h	\01h\02h\04h\05h\0Ah\14h\19h\32h\64h. ¹
470h	MODO DE COM:
480h	MODO DE PRUEBAS
490h	00:00:00?HORA
4A0h	PRESIONE
4B0h	EN MAL ESTADO
4C0h	00.000% FIN=ESC
4D0h	PRUEBA COM:
4E0h	00%?FRONTERA
4F0h	001?MUESTRAS/SEG
500h	SI?REGISTRAR
510h	NO?REGISTRAR
520h	MEM?ALMACENA

530h	COM?ENVIA
540h	MODULO DE MUESTREO
550h	SISTEMA PARA DET
560h	ECCION DE GASES
570h	COMBUSTIBLES\10h\13h ¹
580h	MEMORIA LLENA
590h	ERROR EN SISTEMA
5A0h	FIN DE COM:
5B0h	TRANSMITIENDO
5C0h	PREPARE PC
5D0h	NO HAY DATOS

¹ La notación \0xh, corresponde a un dato hexadecimal.

II.Registros de Datos (D0-D7).

- D0** registro de trabajo
- D1** registro de trabajo
- D2** vector de configuración
(frontera : word, 0.01 segundo/muestra : word)
- D3** dato muestreado
- D4** pase de parámetros a rutinas del teclado
- D5** hora del sistema
(horas : byte, minutos : byte, segundos : byte, centésimas : byte)
- D6** contador de centésimas de segundo por muestra
- D7** pase de parámetros con rutinas de la pantalla de cristal líquido

III.Registros de direcciones (A0-A7).

- A0** pase de parámetros con rutina LCDSTR
(coloca cadena de caracteres en la pantalla de cristal líquido)

-
- A1** dirección de inicio de la tabla de divisores de 100
 - A2** dirección de memoria RAM que se transmite por el puerto serial
 - A3** vector de configuración
(500h = sí registra, 510h = no registra)
 - A4** vector de configuración
(520h = almacena en memoria RAM, 530h = envía por puerto serial)
 - A5** dirección de memoria RAM, apuntador al tope de la pila de datos muestreados que se almacenan
 - A6** registro de trabajo
 - A7** apuntador a la pila del sistema

APÉNDICE C.

ESTÁNDAR DE COMUNICACIÓN RS-232.

Niveles de voltaje:

+5 a +15 V

Dato:

"0"

Línea de control:

"verdadera" o "activa".

-5 a -15 V

Dato:

"1"

Línea de control:

"falsa" o "inactiva".

Puerto serial, configuración 25 terminales.

Terminal	Abreviatura	Dirección	Designación
1	GND	ambas	Frame Ground
2	TxD	a modem	Transmitted Data
3	RxD	a terminal	Received Data
4	RTS	a modem	Request to Send
5	CTS	a terminal	Clear to Send
6	DSR	a terminal	Data Set Ready
7	SG	ambas	Signal Ground
8	CD	a terminal	Carrier Detect
20	DTR	a modem	Data Terminal Ready
22	RI	a terminal	Ring Indicator

Designación	Traducción	Función
Frame Ground	Tierra Física	Se conecta al marco metálico de la terminal o del "modem"
Transmitted Data	Transmisión de Datos	Transmisión de Datos
Received Data	Recepción de Datos	Recepción de Datos
Request to Send	Petición de Envío	Se activa cuando la terminal está encendida y es capaz de comunicarse
Clear to Send	Limpia para Enviar	Se activa cuando la computadora del otro extremo es capaz de recibir
Data Set Ready	Conjunto de Datos Listo	Se activa cuando el "modem" está encendido y funcionando
Signal Ground	Tierra Eléctrica	Punto de referencia para todas las señales de voltaje
Carrier Detect	Detección de Portadora	Se activa cuando se establece comunicación con otra computadora
Data Terminal Ready	Terminal de Datos Lista	Se activa cuando la terminal está encendida y funcionando
Ring Indicator	Indicador de Campana	Se activa cuando el teléfono está llamando

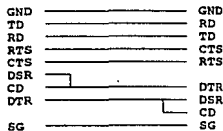
Puerto serial, configuración 9 terminales.

- 1 Carrier Detect
- 2 Receive Data
- 3 Transmit Data
- 4 Data Terminal Ready
- 5 Signal Ground
- 6 Data Set Ready
- 7 Request to Send
- 8 Clear to Send
- 9 Ring Indicator

Conexión simple para eliminar modem, sacrificando algunas líneas de control.



Conexión completa para eliminar modem.



APÉNDICE D.

MANUAL DE OPERACIÓN.

Operación del Sistema para Detección de Gases Combustibles.

Al encender el sistema, éste realizará:

1. Prueba de pantalla de cristal líquido y señales luminosas.
(Enciende todos los puntos de la pantalla y las señales luminosas)
2. Aparece letrero "**CONFIGURANDOSE**"
3. Aparece letrero "**VERIFICANDO RAM**"
4. Aparece letrero "**OPCION (EDC)**". Elija una de las siguientes opciones del menú principal:

Modo de Pruebas:	Presione la tecla E (paso 17).
Modo de Comunicación:	Presione la tecla D (paso 13).
Modo de Muestreo:	Presione la tecla C (paso 5).

Si presiona la tecla F, será ignorada. Si presiona cualquier otra tecla, aparecerá el letrero "**OPCION INVALIDA**" momentáneamente.

Modo de Muestreo.

5. Aparece letrero **"MODO DE MUESTREO"** momentáneamente.
6. Aparece letrero **"001?MUESTRAS/SEG"**. Elija el número de muestras por segundo: 1, 2, 4, 5, 10, 20, 25, 50 ó 100. Las opciones se muestran presionando la tecla B, y se selecciona presionando la tecla A.

Si presiona la tecla F, aparecerá el letrero inicial. Si presiona cualquier otra tecla, será ignorada.

7. Aparece letrero **"00%?FRONTERA LEL"**. Introduzca el valor inferior a partir del cual se registrarán los datos adquiridos: 00 al 99 por ciento. Utilice las teclas del 0 al 9.

Si presiona la tecla F, aparecerá el letrero inicial (borra). Si presiona cualquier otra tecla, será ignorada.

8. Aparece letrero **"SI?REGISTRAR"**. Elija si desea registrar los datos adquiridos o no. Las opciones **"SI REGISTRAR"** y **"NO REGISTRAR"** se muestran presionando la tecla B, y se selecciona presionando la tecla A.

Si presiona la tecla F, aparecerá el letrero inicial. Si presiona cualquier otra tecla, será ignorada.

9. Si elige **"NO?REGISTRAR"**, diríjase al paso 11.

Sólo si elige **"SI?REGISTRAR"**, aparece letrero **"MEM?ALMACENAR"**. Elija si desea almacenar los datos adquiridos en la memoria RAM local (**"MEM ALMACENAR"**), o si desea enviarlos por el puerto serial directamente hacia una computadora personal (**"COM ENVIAR"**). Las opciones se muestran presionando la tecla B, y se selecciona presionando la tecla A.

Si presiona la tecla F, aparecerá el letrero inicial. Si presiona cualquier otra tecla, será ignorada.

10. Si elige **"MEM ALMACENAR"**, dirijase al paso 11.

Solo si elige **"COM?ENVIAR"**, aparece letrero **"PREPARE PC"**. Cuando la computadora personal esté habilitada para recibir los datos adquiridos por el puerto serial, presione cualquier tecla del sistema para continuar. Los datos serán exhibidos simultáneamente en la pantalla de cristal líquido del sistema y en la computadora personal.

11. Aparece letrero **"00:00:00?HORA"**. Introduzca la hora, considerando: horas de 00 a 23, minutos de 00 a 59 y segundos de 00 a 59. Utilice las teclas del 0 al 9.

Si presiona la tecla F, aparecerá el letrero inicial (borra). Si presiona cualquier otra tecla, será ignorada.

12. Aparece letrero **"HH:MM:SS→XX.XXX%"**. Exhibe el dato adquirido (XX.XXX), y la hora cuando se adquirió (HH:MM:SS).

Si presiona la tecla F, abandonará el Modo de Muestreo y regresa al menú principal (paso 4). Si presiona cualquier otra tecla, será ignorada.

Si se encuentra almacenando los datos adquiridos en la memoria RAM local, cuando ésta se ocupe completamente, aparecerá el letrero **"MEMORIA LLENA"**. Entonces presione cualquier tecla para continuar y regresa al menú principal (paso 4).

Modo de Comunicación.

13. Aparece letrero **"MODO DE COM:"** momentáneamente.

Si no existen datos previamente adquiridos en la memoria RAM local, aparecerá letrero **"NO HAY DATOS"**. Entonces, presione cualquier tecla para continuar y regresa al menú principal (paso 4).

14. Aparece letrero **"PREPARE PC"**. Cuando la computadora personal esté habilitada para recibir los datos adquiridos por el puerto serial, presione cualquier tecla del sistema para continuar.
15. Aparece letrero **"TRANSMITIENDO"** momentáneamente. Esto significa que los datos adquiridos se están transmitiendo por el puerto serial a la computadora personal.
16. Aparece letrero **"FIN DE COM:"**. Esto significa que los datos adquiridos ya fueron enviados por el puerto serial a la computadora personal. Presione cualquier tecla para continuar y regresa al menú principal (paso 4).

Modo de pruebas.

17. Aparece letrero "**MODO DE PRUEBAS**" momentáneamente.
18. Prueba de pantalla de cristal líquido y señales luminosas. Enciende todos los puntos de la pantalla y las señales luminosas por un instante.
19. Prueba de teclado. Aparece letrero "**PRESIONE F**". Entonces presione la tecla F. Si presiona cualquier otra tecla, aparece letrero "**EN MAL ESTADO**" momentáneamente. Esta acción se repite para cada tecla: F, E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1 y 0.
20. Prueba de convertidor analógico-digital. Aparece letrero "**XX.XXX% FIN=ESC**", donde XX.XXX representa el dato adquirido. Presione cualquier tecla para finalizar la prueba.
21. Prueba de comunicación. Aparece letrero "**PRUEBA DE COM:**" momentáneamente.
22. Aparece letrero "**PREPARE PC**". Cuando la computadora personal esté habilitada para recibir los datos adquiridos por el puerto serial, presione cualquier tecla del sistema para continuar.
23. Aparece letrero "**TRANSMITIENDO**" momentáneamente. Esto significa que una cadena de caracteres se está transmitiendo por el puerto serial a la computadora personal.

Aparece en la computadora personal la cadena de caracteres "**SISTEMA PARA DETECCION DE GASES COMBUSTIBLES**" repetidamente.

24. Aparece letrero "**FIN DE COM:**". Esto significa que la cadena de caracteres ya fue enviada por el puerto serial a la computadora personal. Presione cualquier tecla para continuar y regresa al menú principal (paso 4).

Señales luminosas.

El Sistema para Detección de Gases Combustibles cuenta con cuatro señales luminosas relacionadas con la adquisición de datos. Dicha relación se presenta en la siguiente tabla:

Señal luminosa	Significado
LED blanco	dato mayor que frontera
LED verde	00% a 25% de LEL
LED amarillo	25% a 50% de LEL
LED rojo	50% a 99% de LEL

APÉNDICE E.

PROGRAMACIÓN DE LA COMPUTADORA PERSONAL.

A continuación se presentan los programas fuentes utilizados para recibir los datos adquiridos por el Sistema para Detección de Gases Combustibles. Los programas son: **COM.C**, **SERIAL.H**, **SERIAL.C**, **DISPLAY.H** y **DISPLAY.C**. Los programas se realizaron en lenguaje C y se compilaron con el paquete Turbo C++.

```
/* com.c */
#include <stdio.h>
#include <conio.h>
#include <alloc.h>
#include <dos.h>
#include "serial.h"
#include "display.h"

/* prototipo de las funciones */
void prueba(void);
void recepcion_continua(void);
void recepcion_bloque(void);

struct elemento{
    unsigned char dato[5];
    struct elemento *liga;
};

typedef struct elemento ELEMENTO;
typedef ELEMENTO *ENLACE;

main(){
    char opcion;
    inicializa(1,CONF);
    while(1){
        clrscr();
        printf("Elige opción:\n");
        printf("    C : recepción de datos continua\n");
```

```

printf(" D : recepción de datos por bloque\n");
printf(" E : prueba de comunicación\n");
printf(" F : salir\n");
printf(">");
scanf("%c",&opcion);
switch (opcion){
    case 'C':
    case 'c':
        recepcion_continua();
        break;
    case 'D':
    case 'd':
        recepcion_bloque();
        break;
    case 'E':
    case 'e':
        prueba();
        break;
    case 'F':
    case 'f':
        exit(0);
}
}
}

```

```

#define REN 10
#define COL 10

```

```

void recepcion_continua(void){
    scr far *pantalla;
    unsigned char caracter;
    unsigned int buffer,video;
    int i=0;
    ENLACE cabeza,p,q;

    video=video_segment();
    pantalla=MK_FP(video,0);
    (*pantalla)[REN][COL+ 2].ch=': ';
    (*pantalla)[REN][COL+ 5].ch=': ';
    (*pantalla)[REN][COL+ 8].ch=': ';
    (*pantalla)[REN][COL+11].ch='>';
    (*pantalla)[REN][COL+14].ch='.';
    (*pantalla)[REN][COL+18].ch='&';

    cabeza=(ENLACE)malloc(sizeof(ELEMENTO));
    p=cabeza;
    p->liga=NULL;

    do{/* caracteres de sincronización */
        while(!(estado(1)&0x0100));
        recibe(1,&caracter);
    }while(caracter!=0xaa);

    while(!kbhit()){
        do{/* carácter de sincronización */
            while(!(estado(1)&0x0100));
            recibe(1,&caracter);
        }
    }
}

```

```

)while(caracter!=0xAA);

q=(ENLACE)malloc(sizeof(ELEMENTO));
q->liga=NULL;
p->liga=q;
p=q;

while(!(estado(1)&0x0100));
recibe(1,&p->dato[0]);
(*pantalla){REN}[COL+ 0].ch=p->dato[0]/10+48;
(*pantalla){REN}[COL+ 1].ch=p->dato[0]%10+48;
while(!(estado(1)&0x0100));
recibe(1,&p->dato[1]);
(*pantalla){REN}[COL+ 3].ch=p->dato[1]/10+48;
(*pantalla){REN}[COL+ 4].ch=p->dato[1]%10+48;
while(!(estado(1)&0x0100));
recibe(1,&p->dato[2]);
(*pantalla){REN}[COL+ 6].ch=p->dato[2]/10+48;
(*pantalla){REN}[COL+ 7].ch=p->dato[2]%10+48;
while(!(estado(1)&0x0100));
recibe(1,&p->dato[3]);
(*pantalla){REN}[COL+ 9].ch=p->dato[3]/10+48;
(*pantalla){REN}[COL+10].ch=p->dato[3]%10+48;
while(!(estado(1)&0x0100));
recibe(1,&p->dato[4]);
buffer=(unsigned int)p->dato[4];
buffer*=2;
(*pantalla){REN}[COL+12].ch=buffer/51+48;
buffer=(buffer%51)*10;
(*pantalla){REN}[COL+13].ch=buffer/51+48;
buffer=(buffer%51)*10;
(*pantalla){REN}[COL+15].ch=buffer/51+48;
buffer=(buffer%51)*10;
(*pantalla){REN}[COL+16].ch=buffer/51+48;
buffer=(buffer%51)*10;
(*pantalla){REN}[COL+17].ch=buffer/51+48;

}

/* muestra datos en pantalla */
clrscr();
getch();
p=cabeza->liga;
while(p){
    printf("%5d%5d%5d%5d%5d\n",p->dato[0],p->dato[1],p->dato[2],
        p->dato[3],p->dato[4]);
    if(++i%20){
        while(!kbhit());
        getch();
        clrscr();
        i=0;
    }
    p=p->liga;
}
while(!kbhit());
}

```

```

void recepcion_bloque(void){
    unsigned char caracter[7200];
    int i;

    clrscr();
    do{/* caracteres de sincronización */
        while(!((estado(1)&0x0100)));
        recibe(1,&caracter[0]);
    }while(caracter[0]!=0xaa);

    for(i=0;i<7200;i++){/* recibe 7200 caracteres */
        while(!((estado(1)&0x0100)));
        recibe(1,&caracter[i]);
    }

    /* muestra datos en pantalla */
    for(i=0;i<1200;i++){
        printf("%5d%5d%5d%5d\n",caracter[6*i],caracter[6*i+1],
            caracter[6*i+2],caracter[6*i+3],caracter[6*i+5]);
        if(!((i+1)%20)){
            while(!kbhit());
            getch();
            clrscr();
        }
    }

    while(!kbhit());
}

void prueba(void){
    unsigned char caracter;
    int i;

    clrscr();
    do{/* caracteres de sincronización */
        while(!((estado(1)&0x0100)));
        recibe(1,&caracter);
    }while(caracter!=0xaa);

    for(i=0;i<920;i++){/* recibe 920 caracteres */
        while(!((estado(1)&0x0100)));
        recibe(1,&caracter);
        putchar(caracter);
    }

    while(!kbhit());
}

```

```

/* serial.h */
#define CONF 0xE3
/* b7=1, b6=1, b5=1    9600 baud      */
/* b4=0, b3=0          sin paridad    */
/* b2=0                1 bits de parada */

```

```

/* b1=1, b0=1      8 bits de datos */

/*      prototipos de la funciones      */
/* void inicializa(unsigned int puerto, unsigned int configuracion) */
void inicializa(unsigned int, unsigned char);

/* unsigned char transmite(unsigned int puerto, unsigned char caracter) */
/* regresa el valor de 1 si la transmisión fue exitosa */
/* regresa el valor de 0 si la transmisión no fue exitosa */
unsigned char transmite(unsigned int, unsigned char);

/* unsigned char recibe(unsigned int puerto, unsigned char *ptr_caracter) */
/* regresa el valor de 1 si la recepción fue exitosa */
/* regresa el valor de 0 si la recepción no fue exitosa */
unsigned char recibe(unsigned int, unsigned char *);

/* unsigned int estado(unsigned int puerto) */
/* regresa 2 bytes de estado asociados al puerto de comunicaciones */
unsigned int estado(unsigned int);

```

```

/* serial.c */
#include <dos.h>

/*      inicialización del puerto de comunicacion      */
void inicializa(unsigned int puerto, unsigned char codigo){
    union REGS r;
    r.h.ah = 0x00;      /* funcion 00h, inicializacion */
    r.x.dx = puerto;    /* numero del puerto de comunicacion */
    r.h.al = codigo;    /* parametros de configuracion */
    int86(0x14, &r, &r);
}

/*      transmision de un caracter por el puerto de comunicacion      */
unsigned char transmite(unsigned int puerto, unsigned char caracter){
    union REGS r;
    r.h.ah = 0x01;      /* funcion 01h, transmision */
    r.x.dx = puerto;    /* numero del puerto de comunicacion */
    r.h.al = caracter;  /* caracter a transmitir */
    int86(0x14, &r, &r);
    if(r.h.ah & 0x80)    /* prueba bit 7 */
        return 0;      /* bit 7 = 1, error en transmision */
    else
        return 1;      /* bit 7 = 0, transmision exitosa */
}

/*      recepcion de un caracter por el puerto de comunicacion      */
unsigned char recibe(unsigned int puerto, unsigned char *caracter){
    union REGS r;
    r.h.ah = 0x02;      /* funcion 02h, recepcion */
    r.x.dx = puerto;    /* numero del puerto de comunicacion */
    int86(0x14, &r, &r);
    if(r.h.ah & 0x80){  /* prueba bit 7 */
        *caracter = 0x00;
        return 0;      /* bit 7 = 1, error en recepcion */
    }
}

```



```
}
else{
    *caracter = r.h.al; /* caracter leído */
    return 1; /* bit 7 = 0, recepcion exitosa */
}
}

/* estado del puerto de comunicacion */
unsigned int estado(unsigned int puerto){
    union REGS r;
    r.h.ah = 0x03; /* funcio 03h, estado */
    r.x.dx = puerto; /* numero del puerto de comunicacion */
    int86(0x14, &r, &r);
    return r.x.ax; /* estado del puerto */
}
```

```
/* display.h */
typedef struct{
    unsigned char ch;
    unsigned char attr;
}character;

typedef character scr[25][80];

/* prototipo de las funciones */
/* regresa el segmento de memoria que utiliza el display */
unsigned int video_segment(void);
```

```
/* display.c */
/* segmento de memoria que utiliza el display */
unsigned int video_segment(void){
    if(*(char far *)0x00400049==7)
        return 0xb000;
    else
        return 0xb800;
}
```

APÉNDICE F.

HOJAS DE ESPECIFICACIONES.

Se incluyen las hojas de especificaciones de los siguientes dispositivos:

Transductor Catalítico C7061C Det-Tronics	209
Microprocesador MC68000	211
Puerto Paralelo PIA 6821	220
Puerto Serial ACIA 6850	230
Generador de Frecuencias MC14411	239
Pantalla de Cristal Líquido AND671	242
Convertidor Analógico-Digital ADC0804	253

DET TRONICS

SPECIFICATION DATA Combustible Gas Sensor

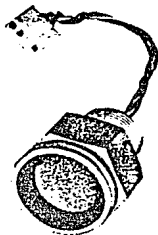
DESCRIPTION

The sensor is a very critical component in a combustible gas detection system. The reliability of the system requires consistent and dependable performance of the sensor under extremes of operating and environmental conditions.

The sensor is essentially two matched reactors, a catalytic element to sense combustible gases and a reference element to compensate for environmental conditions. These elements are electronic components in the same sense as the other electronic parts in the detection systems circuitry. However, unlike the other components, the element pair must operate at a much higher temperature while exposed to the ambient atmosphere. For this reason, Det-Tronics has developed its "Hot-Seal™" special high density alumina sheath to protect the resistance wire of each element. This results in greater mechanical strength, less drift, greater signal stability, more uniform characteristics, and increased sensor life. The unique tubular shape of the Det-Tronics sensor element provides other significant benefits. Because its mass is reduced, the Det-Tronics sensor is less susceptible to mechanical shock and long term vibration effects. In addition, the cylindrical shape provides more uniform heating characteristics and fast stable response time.

Electronic parts produced by component manufacturers have a known reliability as a result of their extensive testing and characterization of the devices. At Det-Tronics, sensor elements are manufactured with the same built-in reliability as the other more common electronic components of the system. Det-Tronics sensor elements are burned in and extensively tested at a number of voltages and under a variety of test gas conditions. The data of these tests are stored in a computer. Devices failing to meet our stringent requirements are rejected. The acceptable active and reference elements are then computer matched as closely as possible in electrical and thermal characteristics. The purpose of this test is to match the active and reference elements to criteria that minimize drift and provide better long term stability over the wide range of temperatures encountered by operational sensors.

Det-Tronics sensors are universal in operation and configuration. They detect all combustible gases and are supplied in a single configuration that is used with all Det-Tronics gas detection systems. This considerably simplifies installation and spares provisioning since only one type of sensor needs to be stocked.



FEATURES

- Computer matched active and reference elements minimize zero and span drift.
- Exclusive Metrosal ceramic coating and tubular shape provide lower vibration sensitivity, more uniform heating, greater stabilization, and fast stable response time.
- Thermal isolation within the sensor housing reduces thermal interaction and assures equal exposure to ambient conditions for both sensor elements.
- Gold plated corrosion inhibiting contacts and keyed connectors simplify maintenance and installation.
- Optional clip-on dust covers are easily installed for protection in dusty or windy environments.
- Full one year warranty on both sensors and electronics in all Det-Tronics gas detection systems.
- Single sensor for all hydrocarbons simplifying installation and replacement.
- Sintered stainless steel flame arrester.

SPECIFICATIONS

Storage Temperature -67°F to +257°F (-55°C to +125°C)
 Optional High temperature model -67°F to +400°F
 (-55°C to +205°C)
 Ambient Operating Temperature -40°F to +185°F
 (-40°C to +85°C)
 Optional high temperature model -40°F to +400°F
 (-40°C to +205°C)
 Response Time* Less than 20 seconds to reach 90%
 of full scale reading with methane (in still air)
 Recovery Time† Less than 30 seconds after exposure
 to pure methane

Sensor Life 2 to 3 years expected
 Calibration Cycle 80 to 90 days
 Linearity* ±5% linearity for all combustible gases
 *For additional information contact the factory for
 technical notes, article reprints and tests on specific
 gases.

ORDERING INFORMATION

Sensor—Aluminum Housing	225006-02
High Temperature Model	226031-01
Poison Resistant Model	226338-01
Sensor—Stainless Steel	225057-01
High Temperature Model	226031-02
Poison Resistant Model	226031-01
Junction Box	226590
Junction Box with Connector Board	226565



Approved

Class I, Division 1, Groups C & D

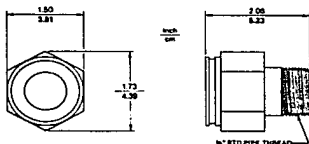


Figure 1—Combustible Gas Sensor Dimensions

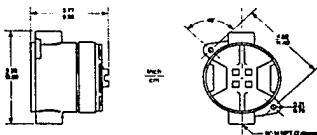


Figure 2—Junction Box Dimensions (not included as part of the sensor, order separately)

Detector Electronics Corporation
 9601 West 110th Street • Minneapolis, Minnesota 55438 USA • Telephone (612) 841-8806
 Facsimile (612) 829-8750 • Telex: 98790-43 DETEL UW • Cable DETRONICS
 Printed in USA

MC68000

Technical Summary

16-/32-Bit Microprocessor

This document contains both a summary of the MC68000 as well as a detailed set of parameters. The purpose is twofold: to provide an introduction to the MC68000 and support for the sophisticated user. For detailed information on the MC68000, refer to the *MC68000 16-/32-Bit Microprocessor Advance Information Data Sheet*.

The MC68000 is the first implementation of the M68000 16/32 microprocessor architecture. The MC68000 has a 16-bit data bus and 24-bit address bus while the full architecture provides for 32-bit address and data buses. It is completely code compatible with the MC68008 8-bit data bus implementation of the M68000 and is upward code compatible to the MC68010, MC68012 virtual extensions and the MC68020 32-bit implementation of the architecture. Any user mode programs written using the MC68000 instruction set will run unchanged on the MC68008, MC68010, MC68020. This is possible because the user programming model is identical for all five processors and the instruction sets are proper sub-sets of the complete architecture. Resources available to the MC68000 user consists of the following:

- 17 32 Bit Data and Address Registers
- 16 Megabyte Direct Addressing Range
- 56 Powerful Instruction Types
- Operations on Five Main Data Types
- Memory Mapped I/O
- 14 Addressing Modes

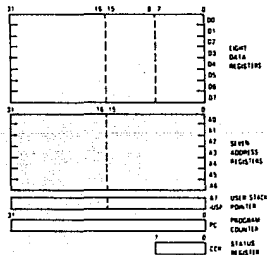


Figure 1 User Programming Model

This document contains information on a new product. Specifications and information herein are subject to change without notice.



MOTOROLA

INTRODUCTION

As shown in the user programming model (Figure 1), the MC68000 offers 18 32 bit registers and a 32 bit program counter. The first eight registers (D0-D7) are used as data registers for byte (8 bit), word (16 bit), and long word (32 bit) operations. The second set of seven registers (A0-A6) and the user stack pointer (USP) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. Any of the 18 registers may be used as index registers.

In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) are also available to the programmer. These registers are shown in Figure 2.

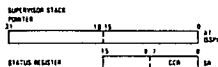


Figure 2 Supervisor Programming Model Supplement

The status register (Figure 3) contains the interrupt mask (eight levels available) as well as the condition codes extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in a trace (T) mode and in a supervisor (S) or user state.

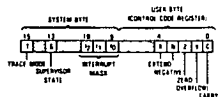


Figure 3 Status Register

DATA TYPES AND ADDRESSING MODES

Five basic data types are supported. These data types are:

- Bits
- BCD Digits (4 Bits)
- Bytes (8 Bits)
- Words (16 Bits)
- Long Words (32 Bits)

In addition, operations on other data types such as memory addresses, status word data, etc. are provided in the instruction set.

The 14 addressing modes, shown in Table 1, include six basic types:

- Register Direct
- Register Indirect
- Absolute

- Program Counter Relative
- Immediate
- Implied

Included in the register indirect addressing modes is a capability to do post-incrementing, pre-decrementing, offsetting, and indexing. The program counter relative mode can also be modified via indexing and offsetting.

Table 1 Addressing Modes

Addressing Modes	Symbol
Register Direct Addressing Data Register Direct Address Register Direct	Dn An
Absolute Data Addressing Absolute Short Absolute Long	###W ###L
Program Counter Relative Addressing Relative with Offset Relative with Index Offset	g(PC) g(PC, Xn)
Register Indirect Addressing Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset	(An) (An) (An) (An) g(PC, An) g(PC, An, Xn)
Immediate Data Addressing Immediate Quick Immediate	### #1-#8
Implied Addressing Implied Register	SR USP SPC

NOTES

- Dn Data Register
- An Address Register
- Bn Address of Data Register used as Index Register
- SR Status Register
- PC Program Counter
- SP Stack Pointer
- USP User Stack Pointer
- g Effective Address
- g 8 Bit Offset Displacement
- g 16 Bit Offset Displacement
- ### Immediate Data

INSTRUCTION SET OVERVIEW

The MC68000 instruction set is shown in Table 2. Some additional instructions are variations or sub-sets of these and they appear in Table 3. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any of the 14 addressing modes. Combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned multiply and divide, "quick" arithmetic operations, BCD arithmetic, and expanded operations through traps.

Table 2. Instruction Set Summary

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical AND
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BCLR	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMF	Compare
DBCC	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive OR
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LNK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical OR
PLA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
RQXR	Rotate Right with Extend
RTI	Return from Exception
RTN	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Sec	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Values
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

Table 3. Variations of Instruction Types

Instruction Type	Variation	Description
ADD	ADD	Add
	ADDA	Add Address
	ADDO	Add Offset
	ADDF	Add Immediate
AND	AND	Logical AND
	ANDP	AND Immediate
	ANDI to CCR	AND Immediate to Condition Codes
	ANDI to SR	AND Immediate to Status Register
CMP	CMP	Compare
	CMFA	Compare Address
	CMFM	Compare Memory
EOR	EOR	Exclusive OR
	EORI to CCR	Exclusive OR Immediate to Condition Codes
	EORI to SR	Exclusive OR Immediate to Status Register
MOVE	MOVE	Move
	MOVEA	Move Address
	MOVEM	Move Multiple Registers
	MOVEP	Move Peripheral Data
	MOVEO	Move Offset
	MOVE from SR	Move from Status Register
MOVE to SR	Move to Status Register	
NEG	MOVE to CCR	Move to Condition Codes
	MOVE LSP	Move User Stack Pointer
NEG	NEG	Negate
	NEGR	Negate with Extend
OR	OR	Logical OR
	ORI to CCR	OR Immediate to Condition Codes
	ORI to SR	OR Immediate to Status Register
SUB	SUB	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBO	Subtract Offset
SUBX	Subtract with Extend	

SIGNAL DESCRIPTION

The input and output signals are illustrated functionally in Figure 4 and are described in the following paragraphs.

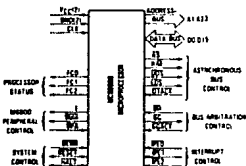


Figure 4. Input and Output Signals

ADDRESS BUS (A1 THROUGH A13)

This 13-bit, unidirectional, three-state bus is capable of addressing 18 megabytes of data. It provides the address for bus operation during all cycles except interrupt cycles. During interrupt cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A4 through A13 are set to a logic high.

DATA BUS (D0 THROUGH D15)

This 16-bit, bidirectional, three-state bus is the general purpose data path. It can transfer and accept data in either word or byte length. During an interrupt acknowledge cycle, the external device supplies the vector number on data lines D0-D7.

ASYNCHRONOUS BUS CONTROL

Asynchronous data transfers are handled using the following control signals: address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are explained in the following paragraphs.

Address Strobe (\overline{AS})

This signal indicates that there is a valid address on the address bus.

Read/Write (\overline{RW})

This signal defines the data bus transfer as a read or write cycle. The \overline{RW} signal also works in conjunction with the data strobes as explained in the following paragraph.

Upper and Lower Data Strobe (\overline{UDS} , \overline{LDS})

These signals control the flow of data on the data bus, as shown in Table 4. When the \overline{RW} line is high, the processor will read from the data bus as indicated. When the \overline{RW} line is low, the processor will write to the data bus as shown.

Data Transfer Acknowledge (\overline{DTACK})

This input indicates that the data transfer is completed. When the processor recognizes \overline{DTACK} during a read cycle, data is latched and the bus cycle terminated. When \overline{DTACK} is recognized during a write cycle, the bus cycle is terminated.

BUS ARBITRATION CONTROL

The three signals, bus request, bus grant, and bus grant acknowledge, form a bus arbitration circuit to determine which device will be the bus master device.

Bus Request (\overline{BR})

This input is wired ORed with all other devices that could be bus masters. This input indicates to the processor that some other device desires to become the bus master.

Bus Grant (\overline{BG})

This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

Bus Grant Acknowledge (\overline{BGACK})

This input indicates that some other device has become the bus master. This signal should not be asserted until the following four conditions are met:

- 1 a bus grant has been received,
- 2 address strobe is inactive which indicates that the microprocessor is not using the bus,
- 3 data transfer acknowledge is inactive which indicates that neither memory nor peripherals are using the bus, and
- 4 bus grant acknowledge is inactive which indicates that no other device is still claiming bus mastership.

Table 4. Data Strobe Control of Data Bus

\overline{UDS}	\overline{LDS}	\overline{RW}	D0-D15	D0-D7
High	High	—	No Valid Data	No Valid Data
Low	Low	High	Valid Data Bits 0-15	Valid Data Bits 0-7
High	Low	High	No Valid Data	Valid Data Bits 0-7
Low	High	High	Valid Data Bits 0-15	No Valid Data
Low	Low	Low	Valid Data Bits 0-15	Valid Data Bits 0-7
High	Low	Low	Valid Data Bits 0-7*	Valid Data Bits 0-7
Low	High	Low	Valid Data Bits 0-15*	Valid Data Bits 0-15*

*These conditions are a result of current implementation and may not appear on future devices.

INTERRUPT CONTROL (IPLO, IPL1, IPC)

These input pins indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. Level seven cannot be masked. The least significant bit is given in IPLO and the most significant bit is contained in IPL2. These lines must remain stable until the processor signals interrupt acknowledge (ICACK) are all high) to ensure that the interrupt is recognized.

SYSTEM CONTROL

The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred. The three system control inputs are explained in the following paragraphs.

Bus Error (BERA)

This input informs the processor that there is a problem with the cycle currently being executed. Problems may be a result of:

1. nonresponding devices.
2. interrupt vector number acquisition failure.
3. illegal access request as determined by a memory management unit, or
4. other application dependent errors.

The bus error signal interacts with the halt signal to determine if the current bus cycle should be re-executed or if exception processing should be performed.

Reset (RESET)

This bidirectional signal line acts to reset (start a system initialization sequence) the processor in response to an external reset signal. An internally generated reset (result of a RESET instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external HALT and RESET signals applied at the same time.

Halt (HALT)

When this bidirectional line is driven by an external device it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all three state lines are put in their high-impedance state.

When the processor has stopped executing instructions, such as in a double bus fault condition, the HALT line is driven by the processor to indicate to external devices that the processor has stopped.

M6800 PERIPHERAL CONTROL

These control signals are used to allow the interfacing of synchronous M6800 peripheral devices with the asynchronous M6800. These signals are explained in the following paragraphs.

Enable (E)

The signal is the standard enable signal common to all M6800 type peripheral devices. The period for this output is ten M6800 clock periods (six clocks low, four clocks high). Enable is generated by an internal ring counter which may come up in any state (i.e., at power on, it is impossible to guarantee phase relationship of E to CLK). E is a free-running clock and runs regardless of the state of the bus or the MPU.

Valid Peripheral Address (VPA)

This input indicates that the device or region addressed is an M6800 Family device or region addressed is an M6800 Family device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt during an IACK cycle.

Valid Memory Address (VMA)

This output is used to indicate to M6800 peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable. This signal only responds to a valid peripheral address (VPA) input which indicates that the peripheral is an M6800 Family device.

PROCESSOR STATUS (FC0, FC1, FC2)

These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed, as shown in Table 5. The information indicated by the function code outputs is valid whenever address strobe (AS) is active.

Table 5. Function Code Outputs

Function Code Output			Cycle Time
FC2	FC1	FC0	
Low	Low	Low	(Undefined, Reserved)
Low	Low	High	User Data
Low	High	Low	User Program
Low	High	High	(Undefined, Reserved)
High	Low	Low	(Undefined, Reserved)
High	Low	High	Supervisor Data
High	High	Low	Supervisor Program
High	High	High	Interrupt Acknowledge

CLOCK (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input should not be gated off at any time and the clock signal must conform to minimum and maximum pulse width times.

DATA TRANSFER OPERATIONS

Transfer of data between devices involves the following leads:

1. address bus A1 through A23
2. data bus D0 through D15, and
3. control signals.

The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure in all cycles. The bus master assumes responsibility for destashing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for destashing the acknowledge and data signals from the slave device.

The following paragraphs explain the read, write, and read-modify-write cycles. The indivisible read-modify-write cycle is the method used by the MC68000 for interleaved multiprocessor communications.

READ CYCLE

During a read cycle, the processor receives data from the memory of a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both upper and lower bytes simultaneously by asserting both upper and lower data strobes. When the instruction specifies byte operation, the processor uses an internal AD bit to determine which byte to read and then issues the data strobe required for that byte. For byte operations, when the AD bit equals zero, the upper data strobe is issued. When the AD bit equals one, the lower data strobe is issued. When the data is received, the processor corrects positions if internally.

WRITE CYCLE

During a write cycle, the processor sends data to either the memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes. When the instruction specifies a byte operation, the processor uses an internal AD bit to determine which byte to write and then issues the data strobe required for that byte. For byte operations, when the AD bit equals zero, the upper data strobe is issued. When the AD bit equals one, the lower data strobe is issued.

READ-MODIFY-WRITE CYCLE

The read-modify-write cycle performs a read, modifies the data in the arithmetic logic unit, and writes the data back to the same address. In the MC68000, this cycle is indivisible in that the address strobe is asserted through out the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment.

This instruction is the only instruction that uses the read-modify-write cycles and since the test and set instruction only operates on bytes, all read-modify-write cycles are byte operations.

PROCESSING STATES

The MC68000 is always in one of three processing states: normal, exception, or halted.

NORMAL PROCESSING

The normal processing state is that associated with instruction execution. The memory references are to fetch instructions and operands, and to store results. A special case of normal state is the stopped state which the processor enters when a stop instruction is executed in this state; no further references are made.

EXCEPTION PROCESSING

The exception processing state is associated with interrupts, trap instructions, tracing, and other exception conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor may handle unusual conditions.

HALTED PROCESSING

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restore a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

INTERFACE WITH M6800 PERIPHERALS

Motorola's extensive line of M6800 peripherals are directly compatible with the MC68000. Some of these devices that are particularly useful are:

- MC6521 Peripheral Interface Adapter
- MC6840 Programmable Timer Module
- MC6843 Floppy Disk Controller
- MC6845 CRT Controller
- MC6850 Asynchronous Communications Interface Adapter

MC6854 Advanced Data Link Controller

To interface the synchronous M6800 peripherals with the asynchronous MC68000, the processor modifies its bus cycle to meet the M6800 cycle requirements whenever an M6800 device address is detected. This is possible since both the processors use memory mapped I/O.

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V _{CC}	-0.3 to +7.0	V
Input Voltage	V _{in}	-0.3 to +7.0	V
Operating Temperature Range MC58000 MC5800C	T _A	T ₁ to T ₂ 0 to 70 -40 to 85	°C
Storage Temperature	T _{stg}	-65 to 150	°C

The device contains circuitry to protect the inputs against damage due to high static voltages of electric fields. However, normal precautions should be taken to avoid application of voltages higher than maximum-rated voltages to these high-impedance circuitry. Tying unused inputs to the appropriate logic voltage level (e.g., either GND or V_{CC}) enhances reliability of operation.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Symbol	Value	Rating
Thermal Resistance (Still Air) Ceramic, Type L LC Ceramic, Type R MC Plastic, Type P Plastic, Type FN	*JA	30 33 30 45	*JC	15* 15* 15* 25*	°C/W

*Estimate

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vol ± 5%, GND = 0 Vol; T_A = T₁ to T₂)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V _{IH}	2.0	V _{CC}	V
Input Low Voltage	V _{IL}	GND - 0.3	0.8	V
Input Leakage Current - 5.25 V	MRA, BQACK, BA, DTACK, CLK, #PL3-#PL5, DPA, HALT, RESET	-	25	µA
Three State (OH State) Input Current - 2.4 V 0.6 V	A5, A1-A23, B0, DO-D15, FCO-FC2, IDS, RW, UDS, VMEA	-	20	µA
Output High Voltage (I _{OH} = -400 µA) (I _{OH} = -400 µA)	E, A5, A1-A23, B0, DO-D15, FCO-FC2, IDS, RW, UDS, VMEA	V _{OH}	V _{CC} - 0.75	V
Output Low Voltage (I _{OL} = 3.2 mA) (I _{OL} = 8.0 mA) (I _{OL} = 9.3 mA)	A1-A23, B0, FCO-FC2, RESET E, A5, DO-D15, IDS, RW, UDS, VMEA	V _{OL}	0.5 0.5 0.5	V
Power Dissipation (see POWER CONSIDERATIONS)	P _D **	-	-	W
Capacitance (V _{in} = 0 V, T _A = 25 °C, Frequency = 1 MHz)**	C _{in}	-	20.0	pF
Load Capacitance	C _L	-	70 130	pF

*With external pullup resistor of 1.1 Ω

**Capacitance is periodically sampled rather than 100% tested

***During normal operation instantaneous V_{CC} current requirements may be as high as 1.5 A

POWER CONSIDERATIONS

The average de- junction temperature, T_J , in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

- T_A = Ambient Temperature, °C
- θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
- P_D = $P_{INT} + P_{I/O}$
- $P_{INT} = I_{CC} \cdot V_{CC}$, Watts — Chip Internal Power
- $P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} \ll P_{INT}$ and can be neglected. An appropriate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \cdot (T_J + 273) \quad (2)$$

Solving equations (1) and (2) for K gives

$$K = P_D \cdot (T_A + 273) / (\theta_{JA} \cdot P_D) \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at thermal equilibrium) for a known T_A . Using this value of K , the values of P_D and T_J can be obtained by solving equations (1) and (2) respectively for any value of T_A .

The curves shown in Figure 11-1 gives the graphic solution to the above equations for the specified power dissipation of 1.5 watts over the ambient temperature range of -55 °C to 125 °C using a maximum θ_{JA} of 45

°C/W. Ambient temperature is that of the still air surrounding the device. Lower values of θ_{JA} cause the curve to shift downward slightly, for instance, for θ_{JA} of 40 °C/W, the curve is just below 1.4 watts at 25 °C.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA} , representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient air (θ_{CA}). These terms are related by the equation

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling, and thermal convection. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θ_{JC} . Substitution of θ_{JC} for θ_{JA} in equation 3 results in a lower semiconductor junction temperature.

Table 6 summarizes maximum power dissipation and average junction temperature for the curve drawn in Figure 5, using the minimum and maximum values of ambient temperature for different packages and substituting θ_{JC} for θ_{JA} (assuming good thermal management). Table 7 provides the maximum power dissipation and average junction temperature assuming that no thermal management is applied (i.e., still air).

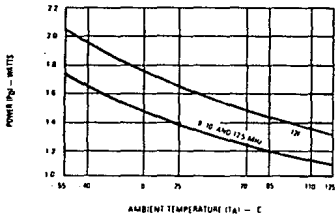


Figure 5. MCE9000 Power Dissipation (P_D) vs Ambient Temperature (T_A)

Table 6. Power Dissipation and Junction Temperature vs Temperature ($\theta_{JC} = \theta_{JA}$)

Package	T _A Range	P _{DC} (C/W)	P _D (W)		T _J (°C)	
			@ T _A Min	@ T _A Max	@ T _A Min	@ T _A Max
LJC	0°C to 70°C	15	1.5	23	1.7	88
	-40°C to 85°C	15	1.7	-14	1.2	103
	0°C to 85°C	15	1.5	23	1.2	103
P	0°C to 70°C	15	1.5	23	1.2	88
RRC	0°C to 70°C	15	1.5	23	1.2	88
	-40°C to 85°C	15	1.7	-14	1.2	103
	0°C to 85°C	15	1.5	23	1.2	103
FN	0°C to 70°C	25	1.5	28	1.2	101

NOTE: Table does not include values for the MC68000 12T

Table 7. Power Dissipation and Junction Temperature vs Temperature ($\theta_{JC} \neq \theta_{JA}$)

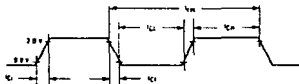
Package	T _A Range	P _{JA} (C/W)	P _D (W)		T _J (°C)	
			@ T _A Min	@ T _A Max	@ T _A Min	@ T _A Max
LJC	0°C to 70°C	30	1.5	23	1.2	88
	-40°C to 85°C	30	1.7	-14	1.2	103
	0°C to 85°C	30	1.5	23	1.2	103
P	0°C to 70°C	30	1.5	23	1.2	88
RRC	0°C to 70°C	33	1.5	23	1.2	88
	-40°C to 85°C	33	1.7	-14	1.2	103
	0°C to 85°C	33	1.5	23	1.2	103
FN	0°C to 70°C	40	1.5	28	1.2	101

NOTE: Table does not include values for the MC68000 12T

AC ELECTRICAL SPECIFICATIONS — CLOCK TIMING (see Figure 6)

Num.	Characteristics	Symbol	8 MHz*		10 MHz*		12.5 MHz*		15.87 MHz* "12T"		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
	Frequency of Operation	f	4.0	8.0	4.0	10.0	4.0	12.5	8.0	16.7	MHz
1	Cycle Time	T _{CY}	125	750	100	250	80	250	60	125	ns
2,3	Clock Pulse Width (Measured from 1.5 V to 1.5 V for 12T)	T _{CL}	55	125	45	125	35	125	27	62.5	ns
		T _{CH}	55	125	45	125	35	125	27	62.5	ns
4,5	Clock Rise and Fall Times	t _r	—	10	—	10	—	5	—	5	ns
		t _f	—	10	—	10	—	5	—	5	ns

*These specifications represent an improvement over previously published specifications for the 8-, 10-, and 12.5-MHz MC68000 and are valid only for product bearing date codes of 8827 and later.



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volt and high a voltage of 2.0 volts, unless otherwise noted. The voltage swing through the rings should start outside and pass through the rings such that the rise or fall will be faster between 0.8 volt and 2.0 volts.

Figure 6. Clock Input Timing


MOTOROLA
PERIPHERAL INTERFACE ADAPTER (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the MC6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8 Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines, Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on A1 and B Side Buffers
- TTL Compatible
- Static Operation

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic 1.5" x 7.5"	1.0	0°C to 70°C	MC6821L
	1.0	-40°C to 85°C	MC6821GL
	1.5	0°C to 70°C	MC6821L
	1.5	-40°C to 85°C	MC6821GL
	2.0	0°C to 70°C	MC6821P
Ceramic 5.5" x 7.5"	1.0	0°C to 70°C	MC6821S
	1.0	-40°C to 85°C	MC6821CS
	1.5	0°C to 70°C	MC6821S
	1.5	-40°C to 85°C	MC6821CS
Plastic 9.5" x 7.5"	1.0	0°C to 70°C	MC6821P
	1.0	-40°C to 85°C	MC6821CP
	1.5	0°C to 70°C	MC6821P
	1.5	-40°C to 85°C	MC6821CP

MC6821
MOS

 IN-CHANNEL, SILICON GATE,
DEPLETION LOAD

**PERIPHERAL INTERFACE
ADAPTER**

 L SUFFIX
CERAMIC PACKAGE
CASE 715

 B SUFFIX
CERAMIC PACKAGE
CASE 714

 P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT

VSS	1	40	CA1
PA0	2	39	CA2
PA1	3	38	CA3
PA2	4	37	CA4
PA3	5	36	CA5
PA4	6	35	CA6
PA5	7	34	CA7
PA6	8	33	CA8
PA7	9	32	CA9
PB0	10	31	CA10
PB1	11	30	CA11
PB2	12	29	CA12
PB3	13	28	CA13
PB4	14	27	CA14
PB5	15	26	CA15
PB6	16	25	CA16
PB7	17	24	CA17
CS1	18	23	CA18
CS2	19	22	CA19
CS3	19	21	CA20
YCC	20	21	CA21

MC6821

MAXIMUM RATINGS

Characteristic	Symbol	Value	Unit
Supply Voltage	V _{CC}	-0.3 to +7.0	V
Input Voltage	V _{IN}	-0.3 to +7.0	V
Operating Temperature Range MC6821, MC68A21, MC68B21 MC6821C, MC68A21C	T _A	T _L to T _H 0 to 70 -40 to +85	°C
Storage Temperature Range	T _{stg}	-55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Cerdip	θ _{JA}	50 100 80	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from

$$T_J = T_A + (P_D + P_{PRT}) \theta_{JA} \quad (1)$$

Where

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = P_{INT} + P_{PORT}

P_{INT} = I_{CC} × V_{CC}, Watts — Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts — User Determined

For most applications P_{PORT} = P_{INT} and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J if P_{PORT} is neglected is

$$P_D = K - (T_J - 273)C \quad (2)$$

Solving equations 1 and 2 for K gives

$$K = P_D \theta_{JA} + 273C \theta_{JA} + P_D \theta_{JA}^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ± 5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise stated)

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R-W, Enable, RESET, RS0, RS1, CS0, CS1, CS2)					
Input High Voltage	V _{IH}	V _{SS} + 2.0	—	V _{CC}	V
Input Low Voltage	V _{IL}	V _{SS} - 0.3	—	V _{SS} + 0.8	V
Input Leakage Current (I _{IN}) (V _{IN} = 0 to 5.25 V)	I _{IN}	—	1.0	2.5	μA
Capacitance (C _{IN}) (V _{IN} = 2.5 V, f = 10 MHz)	C _{IN}	—	—	7.5	pF
INTERRUPT OUTPUTS (IR0A, IR0B)					
Output Low Voltage (I _{OL}) (I _{OL} = 1.0 mA)	V _{OL}	—	—	V _{SS} + 0.4	V
H-Z Output Leakage Current (I _{OZ})	I _{OZ}	—	1.0	10	μA
Capacitance (C _{OUT}) (V _{IN} = 0, T _A = 25°C, f = 1.0 MHz)	C _{OUT}	—	—	6.0	pF
DATA BUS (D0-D7)					
Input High Voltage	V _{IH}	V _{SS} + 2.0	—	V _{CC}	V
Input Low Voltage	V _{IL}	V _{SS} - 0.3	—	V _{SS} + 0.8	V
H-Z Input Leakage Current (I _{IN}) (V _{IN} = 0 to 2.4 V)	I _{IN}	—	2.0	10	μA
Output High Voltage (I _{OH}) (I _{OH} = -200 μA)	V _{OH}	V _{SS} + 2.0	—	—	V
Output Low Voltage (I _{OL}) (I _{OL} = 1.0 mA)	V _{OL}	—	—	V _{SS} + 0.4	V
Capacitance (C _{IN}) (V _{IN} = 0, T _A = 25°C, f = 1.0 MHz)	C _{IN}	—	—	12.5	pF

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields. However, it is advised that normal precautions be taken to avoid applications of any voltage higher than maximum rated voltages to the high impedance input. For proper operation it is recommended that V_{IN} and V_{OUT} be constrained to the range GND ≤ V_{IN} or V_{OUT} ≤ V_{CC}.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

MC6821

DC ELECTRICAL CHARACTERISTICS (Continued)

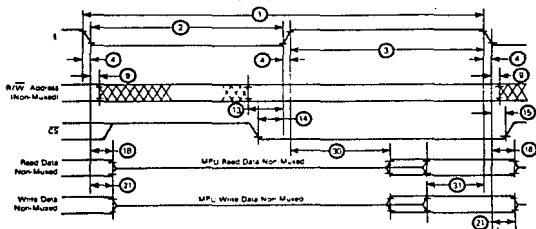
Characteristic		Symbol	Min	Typ	Max	Unit
PERIPHERAL BUS (PAD, PA7, PDD, PB7, CA1, CA2, CB1, CB2)						
Input Leakage Current ($V_{in} = 0$ to 5 V)	R/W, RESET, R50, R51, C50, C51, C52, CA1, CB1 Enable	I_{in}	-	10	28	μ A
H-Z Input Leakage Current ($V_{in} = 0$ to 2.4 V)	PB0-PB7, CB2	I_{HZ}	-	20	10	μ A
Input High Current ($V_{in} = 2.4$ V)	PAD, PA7, CA2	I_{IH}	-200	-400	-	μ A
Drive/Output Current ($V_{out} = 1.5$ V)	PB0-PB7, CB2	I_{OH}	-10	-	-10	mA
Input Low Current ($V_{in} = 0.8$ V)	PAD, PA7, CA2	I_{IL}	-	-13	-3.4	mA
Output High Voltage ($I_{load} = -200$ μ A) ($I_{load} = -10$ μ A)	PAD, PA7, PDD, PB7, CA2, CB2 PAD, PA7, CA2	V_{OH}	$V_{SS} + 2.4$ $V_{CC} - 1.0$	-	-	V
Output Low Voltage ($I_{load} = 3.3$ mA)		V_{OL}	-	-	$V_{SS} + 0.4$	V
Capacitance ($V_{in} = 0$, $f_A = 25$ °C, $f = 1.0$ MHz)		C_{in}	-	-	10	pF
POWER REQUIREMENTS						
Internal Power Dissipation (Measured at $T_A = 25$ °C)		P_{INT}	-	-	500	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Item Number	Characteristic	Symbol	MC6801		MC68A21		MC68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{CYC}	1.0	10	0.67	10	0.5	10	μ s
2	Pulse Width, E Low	PW_{EL}	430	-	280	-	210	-	ns
3	Pulse Width, E High	PW_{EH}	460	-	280	-	220	-	ns
4	Clock Rise and Fall Time	$t_{r, f}$	-	25	-	25	-	30	ns
6	Address Hold Time	t_{AH}	10	-	10	-	10	-	ns
13	Address Setup Time Before E	t_{AS}	80	-	80	-	40	-	ns
14	Chip Select Setup Time Before E	t_{CS}	80	-	80	-	40	-	ns
15	Chip Select Hold Time	t_{CH}	10	-	10	-	10	-	ns
18	Read Data Hold Time	t_{DH}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DWH}	10	-	10	-	10	-	ns
30	Output Data Delay Time	t_{DDR}	-	250	-	160	-	150	ns
31	Input Data Setup Time	t_{DSW}	100	-	80	-	60	-	ns

*The data bus output buffers are no longer sourcing or sinking current if the data is high impedance.

FIGURE 1 - BUS TIMING



Notes:

- 1 Voltage levels shown are $V_L \pm 0.4$ V, $V_H \pm 2.4$ V, unless otherwise specified.
- 2 Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

MC6821

PERIPHERAL TIMING CHARACTERISTICS $V_{CC}=5.0\text{ V} \pm 5\%$, $V_{SS}=0\text{ V}$, $T_A=0\text{ }^{\circ}\text{C}$ to T_H unless otherwise specified

Characteristic	Symbol	MC6821		MC68A31		MC68B31		Unit	Reference Fig. No.
		Min	Max	Min	Max	Min	Max		
Data Setup Time	IPDS	200	-	136	-	100	-	ns	6
Data Hold Time	IPDH	0	-	0	-	0	-	ns	6
Delay Time: Enable Negative Transition to CA2 Negative Transition	ICA2	-	1.0	-	0.670	-	0.500	μs	3, 7, 8
Delay Time: Enable Negative Transition to CA2 Positive Transition	ICSA2	-	1.0	-	0.670	-	0.500	μs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	$t_{r, f}$	-	1.0	-	1.0	-	1.0	μs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	ICSA2	-	2.0	-	1.36	-	1.0	μs	3, 8
Delay Time: Enable Negative Transition to Data Valid	IPDV	-	1.0	-	0.670	-	0.5	μs	3, 9, 10
Delay Time: Enable Negative Transition to CMOS Data Valid (PAD PA7, CA2)	ICMOS	-	2.0	-	1.36	-	1.0	μs	9
Delay Time: Enable Positive Transition to CB2 Negative Transition	ICB2	-	1.0	-	0.670	-	0.5	μs	3, 11, 12
Delay Time: Data Valid to CB2 Negative Transition	IDC	20	-	20	-	20	-	ns	3, 10
Delay Time: Enable Positive Transition to CB2 Positive Transition	ICB1	-	1.0	-	0.670	-	0.5	μs	3, 11
Control Output Pulse Width: CA2/CB2	PWCT	500	-	375	-	250	-	ns	3, 11
Rise and Fall Time to CB1 and CB2 Input Signals	$t_{r, f}$	-	1.0	-	1.0	-	1.0	μs	12
Delay Time: CB1 Active Transition to CB2 Positive Transition	ICB2	-	2.0	-	1.36	-	1.0	μs	3, 12
Interrupt Release Time (IRDA and IRDB)	t_{IR}	-	1.60	-	1.10	-	0.85	μs	5, 14
Interrupt Response Time	IRSD	-	1.0	-	1.0	-	1.0	μs	5, 13
Interrupt Input Pulse Time	IPWI	500	-	500	-	500	-	ns	13
RESET Low Time*	TRL	1.0	-	0.66	-	0.5	-	μs	15

*The RESET line must be high a minimum of 1.0 μs before addressing the PIA.

FIGURE 2 - BUS TIMING TEST LOADS

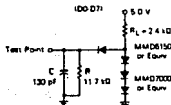


FIGURE 4 - CMOS EQUIVALENT TEST LOAD

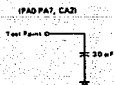


FIGURE 3 - TTL EQUIVALENT TEST LOAD

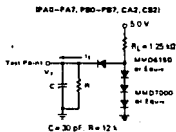
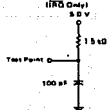


FIGURE 5 - NMOS EQUIVALENT TEST LOAD



MC6821

FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES (Read Mode)



FIGURE 7 — CA2 DELAY TIME (Read Mode; CRA-6 = CRA3 = 1, CRA-4 = 0)

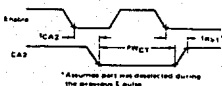


FIGURE 8 — CA2 DELAY TIME (Read Mode; CRA-6 = 1, CRA-3 = CRA-4 = 0)



FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES (Write Mode; CRA-6 = CRA-3 = 1, CRA-4 = 0)



FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES (Write Mode; CRB-6 = CRB-3 = 1, CRB-4 = 0)

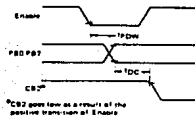


FIGURE 11 — CB2 DELAY TIME (Write Mode; CRB-6 = CRB-3 = 1, CRB-4 = 0)

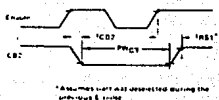


FIGURE 12 — CB2 DELAY TIME (Write Mode; CRB-6 = 1, CRB-3 = CRB-4 = 0)



FIGURE 13 — INTERRUPT PULSE WIDTH AND \overline{IRQ} RESPONSE



Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

MC6821

FIGURE 14 - \overline{RD} RELEASE TIME

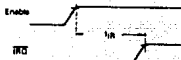
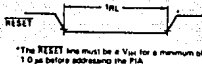


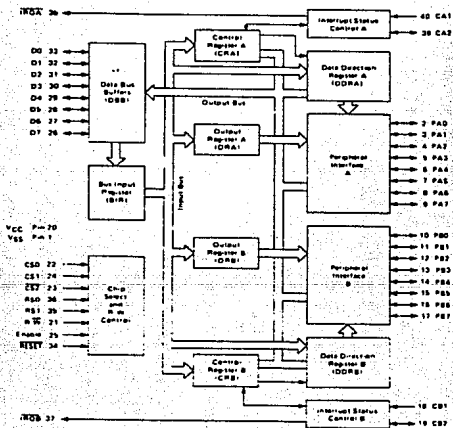
FIGURE 15 - \overline{RD} LOW TIME



*The \overline{RD} low time must be a V_{OH} for a minimum of 1.0 μ s before addressing the PIA.

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 16 - EXPANDED BLOCK DIAGRAM



MC6821

PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the M8000 bus with an 8 bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802, or MC6808 microprocessors, VMA should be used as an active part of the address decoding.

Bidirectional Data (DD/DI) - The bidirectional data lines (DD/DI) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three state devices that remain in the high impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when the PIA is selected for a read operation.

Enable (E) - The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

Read/Write (R/W) - This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse are present.

RESET - The active low RESET line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and CS2) - These three input signals are used to select the PIA. CS0 and CS1 must be high and CS2 must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable

for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) - The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request (IROA and IROB) - The active low Interrupt Request lines (IROA and IROB) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load) devices on the chip. This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each interrupt Request line has two internal interrupt flag bits that can cause the interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Service an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the active edge (D) the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8 bit bidirectional data buses and four interrupt control lines for interfacing to peripheral devices.

Section A Peripheral Data (PAD-PA7) - Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written in to the register will cause a "high" on the corresponding data

line while a "0" results in a "low." Data in Output Register A may be read by an MPU. Read Peripheral Data operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PBO-PB7) - The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PAD-PA7. They have three state capability, allowing them to enter a high impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines

MC6821

PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of at least 1 milliamperes at 1.5 volts to directly drive the base of a transistor switch.

Interrupt Input (CA1 and CB1) — Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) — The peripheral control line CA2 can be programmed to act as an interrupt input or as a

peripheral control output. As an output, this line is compatible with standard TTL, as an input the internal pullup resistor on the line represents 1.5 standard TTL loads. The function of the signal line is programmed with Control Register A.

Peripheral Control (CB2) — Peripheral control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliamperes at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A **RESET** has the effect of zeroing all PIA registers. This will set PAD-PAT, PBD-PBT, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RSD and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1. Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 — INTERNAL ADDRESSING

		Control Register Bit		Location Selected
RS1	RSD	CRB-2	CRA-2	
0	0	1	0	Peripheral Register A
0	0	0	0	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 17, the MC6821 has a pair of I/O ports whose characteristics differ greatly. The A side is designed to drive CMOS logic to normal 30% to 70% levels and incorporate an internal output device that remains connected even in the input mode. Because of this, the A side requires more drive current in the input mode than Port B. In contrast, the B side uses a normal three-state CMOS buffer which cannot pullup to CMOS levels without external resistors. The B side can drive extra loads such as LEDs without problem. When the PIA comes out of reset, the A port represents inputs with pullup resistors, whereas the B side input mode itself will float high or low, depending upon the load connected to it.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction Register when the proper register select signals are applied to RSD and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) — The four interrupt flag bits are set by active transitions of signals on the four interrupt and peripheral control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

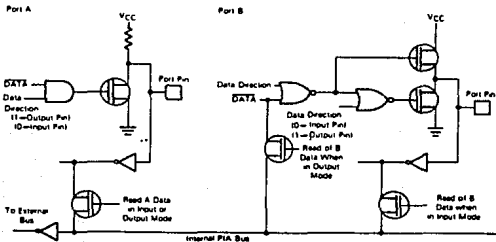
Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) — Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input; if signal to CA1 (CB1) is high, CA2 (CB2) is high; CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

MC6821

Control of CA1 and CB1 interrupt input lines (CRA-0, CRB-0, CRA-1, and CRB-1) - The two lowest order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to

enable the MPU interrupt signals IRQA and IRQB, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 17 - PORT A AND PORT B EQUIVALENT CIRCUITS



ORDERING INFORMATION

MCEBA71CP	
Monopole Integrated Circuit	
MM8000 Family	
Busess = 1.0 Mm	
A = 1.5 MHz	
B = 2.0 MHz	
Device Designation in MM8000 Family	
Temperature Range	
Base = 0° - +70°C	
C = -40° - +85°C	
Package	
P = Plastic	
S = Cer-Dip	
T = Ceramic	

BETTER PROGRAM

Better program processing is available on all 1, pin wets. Add suffix letters to part number.

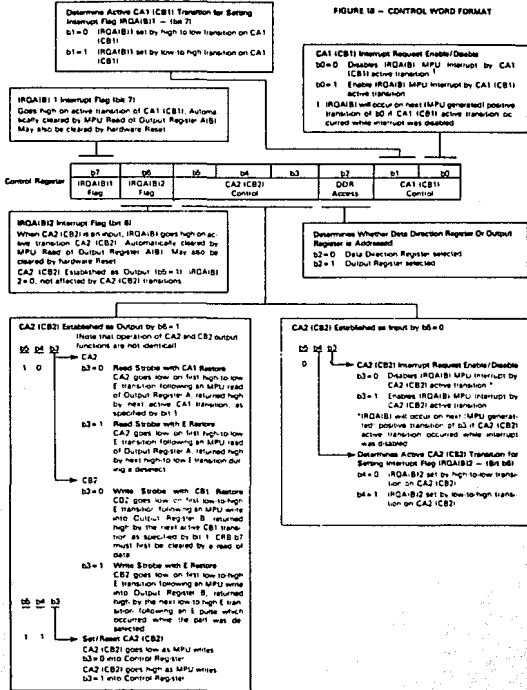
Level 1 add S Level 2 add D Level 3 add DS

Level 1 S = 10 Temp Cycles - 1 = 25 to 150°C
 96 Temp testing at 1.5 max

Level 2 D = 168 hour Burn-in at 125°C

Level 3 DS = Combination of Level 1 and 2

MC6821





MC6850

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC68000 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC68000, 0-600 bps digital modem.

- 5- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional -1, -10, and -64 Clock Modes
- Up to 10 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS IN-CANNEL SILICON GATE ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



8 BUFFERS
CERDIP PACKAGE
CASE 827

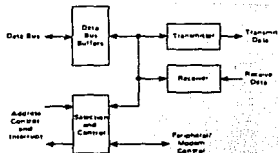


9 BUFFERS
PLASTIC PACKAGE
CASE 708



1 BUFFERS
CERAMIC PACKAGE
CASE 716

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT

V _{SS}	1	24	CTS
Rx Data	2	25	DCD
Rx CLK	3	26	DD
Tx CLK	4	27	D1
RTS	5	28	D2
Tx Data	6	29	D3
TRD	7	30	D4
CS0	8	1	D5
CS1	9	2	D6
CS2	10	3	D7
RS	11	4	E
V _{CC}	12	5	R/W

MC6850

MAXIMUM RATINGS

Characteristic	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	V
Input Voltage	V_{in}	-0.3 to +7.0	V
Operating Temperature Range	T_A	T_L to T_H 0 to 70 -40 to +85	°C
Storage Temperature Range	T_{STG}	-35 to +150	°C

The device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to the high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	120	°C/W
Ceramic		60	
Cardip		60	

POWER CONSIDERATIONS

The average chip-junction temperature, T_J , in °C can be obtained from

$$T_J = T_A + (P_D + P_{PORT}) \theta_{JA} \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

P_D = $P_{INT} + P_{PORT}$

$P_{INT} = ICC \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases of sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is

$$P_D = K - (T_J + 273) \quad (2)$$

Solving equations 1 and 2 for K gives

$$K = P_D (T_A + 273) \theta_{JA} + \theta_{JA} P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A . Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A .

DC ELECTRICAL CHARACTERISTICS ($V_{CC} = 5.0$ Vdc $\pm 5\%$, $V_{SS} = 0$, $T_A = T_L$ to T_H , unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	$V_{SS} + 2.0$	-	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS} - 0.3$	-	$V_{SS} + 0.8$	V
Input Leakage Current ($V_{in} = 0$ to 5.25 V)	I_{IL}	-	-1.0	2.5	μ A
μ -2 I/O State Input Current ($V_{in} = 0$ to 2.4 V)	DD/D†	-	2.0	10	μ A
Output High Voltage ($I_{load} = -200$ μ A, Enable Pulse Width < 25 μ s)	V_{OH}	$V_{SS} + 2.4$	-	-	V
($I_{load} = -100$ μ A, Enable Pulse Width < 25 μ s)	†a Data RTS	$V_{SS} + 2.4$	-	-	V
Output Low Voltage ($I_{load} = 1.6$ mA, Enable Pulse Width < 25 μ s)	V_{OL}	-	-	$V_{SS} + 0.4$	V
Output Leakage Current (I/O State) ($V_{OH} = 2.4$ V)	I_{OH}	-	1.0	10	μ A
Internal Power Dissipation (Measured at $T_A = 0^\circ$ C)	P_{INT}	-	300	525*	mW
Internal Input Capacitance ($V_{in} = 0$, $T_A = 25^\circ$ C, $f = 10$ MHz)	DD/D†	-	10	12.5	pF
(T_A CLK, R_2 CLK, R/W , R_5 , R_6 Data, CS0, CS1, CS7, CTS, DCD)	RTS, T_A Data IRD	-	7.0	7.5	pF
Output Capacitance ($V_{in} = 0$, $T_A = 25^\circ$ C, $f = 10$ MHz)	C_{out}	-	-	10	pF
	IRD	-	-	5.0	pF

*For temperatures less than $T_A = 0^\circ$ C, P_{INT} maximum will increase.

MC6850

SERIAL DATA TIMING CHARACTERISTICS

Characteristic	Symbol	MC6850		MC6840		MC6850		Unit
		Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low (See Figure 1)	PW_{CL}	800	—	400	—	280	—	ns
Data Clock Pulse Width, High (See Figure 2)	PW_{CH}	800	—	400	—	280	—	ns
Data Clock Frequency	f_C	—	0.8	—	1.0	—	1.8	MHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)	TD_{DD}	—	800	—	540	—	400	ns
Receive Data Setup Time (See Figure 4)	TD_{DS}	200	—	100	—	30	—	ns
Receive Data Hold Time (See Figure 5)	TD_{DH}	250	—	100	—	30	—	ns
Interrupt Request Release Time (See Figure 6)	t_{IR}	—	1.2	—	0.9	—	0.7	μ s
Request-to-Send Delay Time (See Figure 6)	$t_{R\overline{TS}}$	—	800	—	480	—	400	ns
Input Rise and Fall Times for 10% of the pulse width if present	t_r, t_f	—	1.0	—	0.8	—	0.25	μ s

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

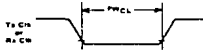


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

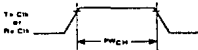
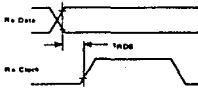
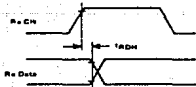
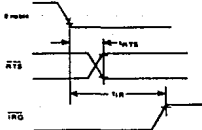


FIGURE 3 — TRANSMIT DATA OUTPUT DELAY

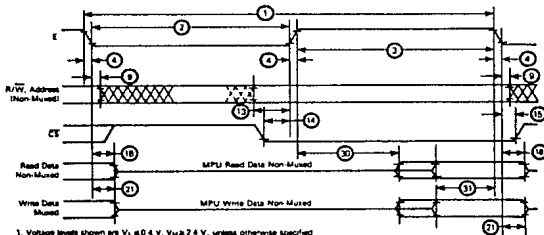
FIGURE 4 — RECEIVE DATA SETUP TIME
(-1 Model)FIGURE 5 — RECEIVE DATA HOLD TIME
(-1 Model)FIGURE 6 — REQUEST-TO-SEND DELAY AND
INTERRUPT-REQUEST RELEASE TIMES

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

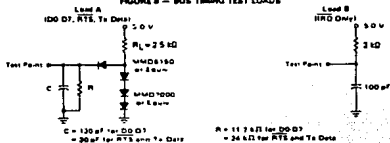
MC6850
BUS TIMING CHARACTERISTICS (See Notes 1 and 2 and Figure 7)

Metric Number	Characteristic	Symbol	MC6850		MC68A0		MC68B0		Units
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{CYC}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, \bar{E} Low	$PW_{\bar{E}(L)}$	430	8000	280	8000	210	8000	ns
3	Pulse Width, \bar{E} High	$PW_{\bar{E}(H)}$	460	8000	280	8000	220	8000	ns
4	Clock Rise and Fall Time	$t_{r/f}$	—	25	—	25	—	20	ns
8	Address Hold Time	t_{AH}	10	—	10	—	10	—	ns
13	Address Setup Time Before \bar{E}	t_{AS}	80	—	80	—	40	—	ns
14	Chip Select Setup Time Before \bar{E}	t_{CS}	80	—	80	—	40	—	ns
16	Chip Select Hold Time	t_{CH}	10	—	10	—	10	—	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	—	10	—	10	—	ns
30	Output Data Delay Time	t_{ODR}	—	200	—	150	—	150	ns
21	Input Data Setup Time	t_{OSW}	100	—	80	—	80	—	ns

 *The data bus output buffers are no longer sourcing or sinking current by t_{DHR} (max High Impedance)

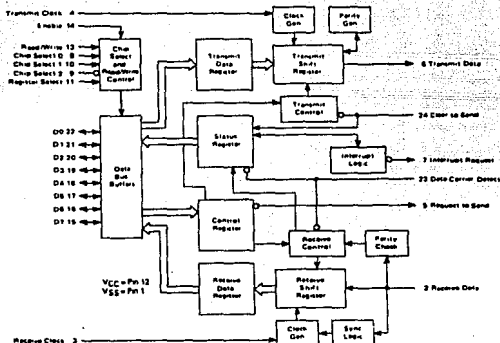
FIGURE 7 — BUS TIMING CHARACTERISTICS


1. Voltage levels shown are $V_L \leq 0.4$ V, $V_{CC} \geq 2.4$ V, unless otherwise specified
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified

FIGURE 8 — BUS TIMING TEST LOADS


MC6850

FIGURE 8 - EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset (CR0, CR1) should be set during system initialization to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. During the first master reset, the IR0 and RTS outputs are held at level 1. On all other master resets, the RTS output can be programmed high or low with the IR0 output held high. Control bits CR5 and CR6 should also be programmed to define the state of RTS whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on, transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The

power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one of two stop bits, parity (even, odd, or none), etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of

MC6850

double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock to its data, while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is insured by the detection of 8 or 32 low samples on the receive line in the divide-by-16 and 64 modes respectively. False start bit detection capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receive Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected (or a 7-bit word (7 bits plus parity), the receiver strips the parity bit (D7=0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the M6800 MPU with an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (DD-D7) — The bidirectional data lines (DD-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 $\phi 2$ Clock or MC6809 E clock.

Read/Write (R/W) — The Read/Write line is a high-impedance input that is TTL-compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are

turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, CS2) — These three high-impedance TTL-compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and CS2 is low. Transfers of data to and from the ACIA are then performed under the control of the Enable Signal, Read/Write, and Register Select.

Register Select (RS) — The Register Select line is a high-impedance input that is TTL-compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request (IRQ) — Interrupt Request is a TTL-compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The IRQ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The IRQ status bit, when high, indicates the IRQ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (ICRS/CR6), and the Transmitter Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send (CTS) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmitter Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CRS or CR6 or by the loss of CTS which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect (DCD) has gone high. An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of DCD are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16, or 64 times the data rate may be selected.

Transmit Clock (Tx CLK) — The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK) — The Receive Clock input is used for synchronization of received data. In the ± 1 mode, the clock and data must be synchronized externally. The receiver samples the data on the positive transition of the clock.

MC6850

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) — The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used.

Transmit Data (Tx Data) — The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send (CTS) — This high-impedance TTL-compatible input provides automatic control of the transmitting end of a communications link via the modem. Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send (RTS) — The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The RTS output corresponds to the state of the Control Register bits CR5 and CR6. When CR6 = 0 or both CR5 and CR6 = 1, the RTS output is low; the active state. This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect (DCD) — This high-impedance TTL-compatible input provides automatic control, such as in the receiving end of a communications link, by means of a modem Data Carrier Detect output. The DCD input inhibits and initializes the receiver section of the ACIA when high. A low-to-high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set. The Rx CLK must be running for proper DCD operation.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed with RS high and R/W low. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 1-bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer to shift register upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/W high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

TABLE 1 — DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	RS = R/W	RS = R/W	RS = R/W	RS = R/W
	Transmit Data Register	Receive Data Register	Control Register	Status Register
0	Write Command	Read Data 0	Control Register	Status Register
1	Data B-0*	Data B-1*	Control Register Select 1, CR0	Transmit Data Register Empty, TDRE
2	Data B-2*	Data B-2*	Control Register Select 2, CR1	Data Carrier Detect, DCD
3	Data B-3	Data B-3	Control Register Select 2, CR2	Clear to Send, CTS
4	Data B-4	Data B-4	Control Register Select 2, CR3	Request to Send, RTS
5	Data B-5	Data B-5	Transmit Control 1, CR4	Receive Data Register Full, RDRF
6	Data B-6	Data B-6	Transmit Control 2, CR5	Receive Data Register Empty, RDRE
7	Data B-7***	Data B-7***	Receive Interrupt Enable, IER	Interrupt Received, IRIF

* Leading bit = 15B = B-0

** Data B-4 and B-5 are 2-bit bus data fields

*** Data B-7 is 8-bit bus data field plus parity message

MC6850

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and R/W are low. The register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send (paraphrase/modem control output).

Counter Divide Select Bits (CR0 and CR1) — The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on CTS and DCD) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restore, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	-1
0	1	-10
1	0	-84
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) — The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even Parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) — Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send (RTS) output, and the transmission of a Break level (special). The following encoding format is used:

CR6	CR5	Function
0	0	RTS = low, Transmitting Interrupt Disabled
0	1	RTS = low, Transmitting Interrupt Enabled
1	0	RTS = high, Transmitting Interrupt Disabled
1	1	RTS = low, Transmits a Break level on the Transmitter Data Output. Transmitting interrupt Disabled.

Receive Interrupt Enable Bit (CR7) — The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low-to-high transition on the Data Carrier Detect (DCD) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. The read-only register is selected when RS is low and R/W is high. Information stored in the register indicates the status of the Transmitt Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 — Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

Transmitt Data Register Empty (TDRE), Bit 1 — The Transmitt Data Register Empty bit being set high indicates that the Transmitt Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect (DCD), Bit 2 — The Data Carrier Detect bit will be high when the DCD input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the DCD input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the DCD input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the DCD status bit remains high and will follow the DCD input.

Clear-to-Send (CTS), Bit 3 — The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low CTS indicates that there is a Clear-to-Send from the modem. In the high state, the Transmitt Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high. Master reset does not affect the Clear-to-Send status bit.

Framing Error (FE), Bit 4 — Framing error indicates that the received character is improperly framed by a start and a stop bit and is deleted by the absence of the first stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5 — Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has

MC6850

been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), BR 6 - The parity error flag indicates that the number of high tones in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data

character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver parity check results are inhibited.

Interrupt Request (IRQ), BR 7 - The IRQ bit indicates the state of the IRQ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the IRQ output is low the IRQ bit will be high to indicate the interrupt or service request status. IRQ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6850CL
	1.0	-40°C to 85°C	MC6850CL
	1.5	0°C to 70°C	MC6850SL
	1.5	-40°C to 85°C	MC6850SL
	2.0	0°C to 70°C	MC6850CL
Ceramic S Suffix	1.0	0°C to 70°C	MC6850CS
	1.0	-40°C to 85°C	MC6850CS
	1.5	0°C to 70°C	MC6850SS
	1.5	-40°C to 85°C	MC6850SS
	2.0	0°C to 70°C	MC6850CS
Plastic P Suffix	1.0	0°C to 70°C	MC6850CP
	1.0	-40°C to 85°C	MC6850CP
	1.5	0°C to 70°C	MC6850SP
	1.5	-40°C to 85°C	MC6850SP
	2.0	0°C to 70°C	MC6850CP

MOTOROLA SEMICONDUCTOR TECHNICAL DATA

MC14411

CMOS LSI

(LOW POWER COMPLEMENTARY MOS)
BIT RATE GENERATOR

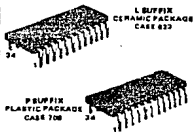
BIT RATE GENERATOR

The MC14411 bit rate generator is constructed with complementary MOS enhancement mode devices. It utilizes a frequency divider network to provide a wide range of output frequencies.

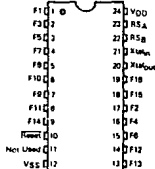
A crystal controlled oscillator is the clock source for the network. A two-bit address is provided to select one of four multiple output clock rates.

Applications include a selectable frequency source for equipment in the data communications market, such as teleprinters, printers, CRT terminals, and microprocessor systems.

- Single 5.0 Vdc $\pm 5\%$ Power Supply
- Internal Oscillator Crystal Controlled for Stability (1.8432 MHz)
- Sixteen Different Output Clock Rates
- 50% Output Duty Cycle
- Programmable Time Bases for One of Four Multiple Output Rates
- Buffered Outputs Compatible with Low Power TTL
- Noise Immunity = 40% of VDD Typical
- Diode Protection on All Inputs
- External Clock May be Applied to Pin 21
- Internal Pullup Resistor on Reset Input



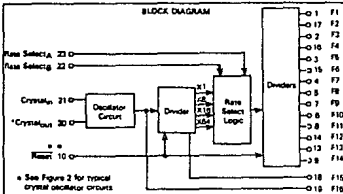
PIN ASSIGNMENT



MAXIMUM RATINGS (voltages referenced to VSS, Pin 12)

Rating	Symbol	Value	Unit
DC Supply Voltage Range	VDD	6.25 to -0.5	V
Input Voltage, All Inputs	Vin	VDD + 0.5 to VSS - 0.5	V
DC Current Drawn per Pin	I	10	mA
Operating Temperature Range	Ta	-40 to +85	°C
Storage Temperature Range	Tstg	-65 to +150	°C

BLOCK DIAGRAM



The device contains circuitry to protect the inputs against damage due to high static voltages or electric fields. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to the high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range VSS ≤ V_{in} or V_{out} ≤ VDD. Unused inputs must always be tied to an appropriate logic voltage level (e.g., either VSS or VDD).

MC14411

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	VDD VDC	-40°C			25°C			+85°C		Units
			Min	Max	Min	Max	Min	Max			
Supply Voltage	VDD	—	4.75	5.25	4.75	5.0	5.25	4.75	5.25	V	
Output Voltage "0" Level "1" Level	V _{OL} V _{OH}	5.0 5.0	— 4.95	— —	0.05 —	— 4.95	0 —	0.05 —	0.05 4.85	V	
Input Voltage (V _{OL} = 4.5 or 0.5 V) (V _{OH} = 0.5 or 4.5 Vdc)	V _{IL} V _{IH}	5.0 5.0	— 3.5	— —	1.5 —	2.25 3.6	1.5 —	— 3.5	1.5 —	V	
Output Drive Current (V _{OH} = 2.5 V) Source (V _{OL} = 0.4 V) Sink	I _{OH} I _{OL}	5.0 5.0	-0.23 0.23	— —	-0.20 0.20	-1.7 0.75	— —	-0.16 0.16	— —	mA	
Input Current Pins 21, 22, 23 Pin 10	I _{in} —	— 5.0	— —	±0.1 —	— -1.5	±0.00001 —	±0.1 -7.5	— —	±1.0 —	µA	
Input Capacitance (V _{in} = 0)	C _{in}	—	—	—	—	—	—	—	—	pF	
Quiescent Dissipation	P _Q	5.0	—	2.6	—	0.015	2.6	—	10	mW	
Power Dissipation** (Dynamic plus Quiescent) (C _L = 15 pF)	P _D	5.0	—	—	—	P _D = 17.5 mW/MHz (1 = P _Q)		—	—	mW	
Output Rise Time** (C _L = 15.0 ns/pF) C _L = 25 ns	T _{PLH}	5.0	—	—	—	70	200	—	—	ns	
Output Fall Time** (C _L = 15.0 ns/pF) C _L = 47 ns	T _{PHL}	5.0	—	—	—	70	200	—	—	ns	
Input Clock Frequency	f _{CL}	5.0	—	1.85	—	—	1.85	—	1.85	MHz	
Clock Pulse Width	t _{W(C)}	—	200	—	200	—	—	200	—	ns	
Reset Pulse Width	t _{W(R)}	—	500	—	500	—	—	500	—	ns	

†For dissipation in different ambient capacitance (C_L) refer to corresponding formula.

**P_D = P_Q + 2.6 + 10 - 3C_L - 16 pF/V_{DD}†

where P_Q in mW, C_L in pF, V_{DD} in Volts, and f in MHz.

**The formula given is for the typical characteristics only.

TABLE 1 — OUTPUT CLOCK RATES

Output Member	Rate Select			
	B	A	Rate	
	0	0	x1	
	0	1	x8	
	1	0	x16	
	1	1	x64	

Output Member	Output Rate (Hz)			
	x8	x16	x8	x1
F1	814.4 k	153.6 k	76.8 k	9600
F3	460.8 k	115.2 k	57.6 k	7200
F3	327.2 k	76.8 k	38.4 k	4800
F4	230.4 k	57.6 k	28.8 k	3600
F5	153.6 k	38.4 k	19.2 k	2400
F6	115.2 k	28.8 k	14.4 k	1800
F7	76.8 k	19.2 k	9600	1200
F8	38.4 k	9600	4800	600
F9	19.2 k	4800	2400	300
F10	12.8 k	3200	1600	200
F11	9600	2400	1200	150
F12	6613.3	2153.3	1076.6	134.6
F13	7035.5	1758.8	879.4	109.8
F14	4800	1200	600	75
F15	921.6 k	921.6 k	921.6 k	921.6 k
F16†	1.943 M	1.943 M	1.943 M	1.943 M

†F16 is buffered oscillator output.

MOTOROLA TELECOMMUNICATIONS DEVICE DATA

MC14411

FIGURE 1 - DYNAMIC SIGNAL WAVEFORMS

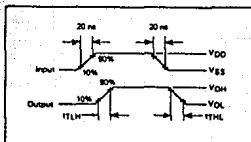
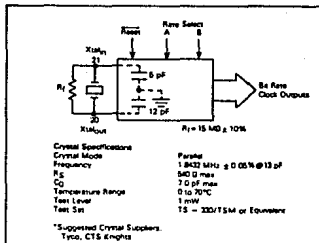


FIGURE 2 - TYPICAL CRYSTAL OSCILLATOR CIRCUIT



Circuit diagrams within Motorola products are included as a means of illustrating typical semiconductor applications. Consequently, complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor device described any license under the patent rights of Motorola Inc., or others.

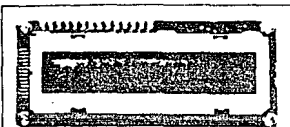
MOTOROLA TELECOMMUNICATIONS DEVICE DATA



LCD Dot Matrix Modules

16 CHARACTERS x 1 LINE

AND671

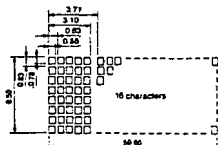


The AND671 is a compact LCD module having a dot matrix LCD panel, a controller and driver circuit. This module can display 160 kinds of letters, numerals, symbols and "Kana" letters, as well as 8 custom characters.

FEATURES

- Compact, integrated display module.
- High contrast, clear display with large characters
- Low voltage, +5V single power supply
- Wide operating temperature range (0°C to +50°C)
- 5 x 7 dot character format and cursor line
- Built-in control LSI with display RAM and character generator ROM.
- Direct interface to 4 or 8 bit CPU.
- 11 commands for control
- Transflective module is available as an option

Dot Matrix Dimensions (in millimeters)



Mechanical Characteristics

Item	Specification	Unit
Outline Dimension	80 W x 38 H x 12 D	mm
Character Size	3.1 W x 5.76 H	mm
Number of Characters	16 x 1 (8) Characters (5 x 7 dot = Cursor)	-
Viewing Area	64.5 W x 13.8 H	mm
Bezel Opening	64.5 W x 13.8 H	mm
Dot Size	0.58 W x 0.78 H	mm
Dot Pitch	0.63 W x 0.83 H	mm
Weight	approx. 25	gram

Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Supply Voltage	V _{DD}	7	V
Input Voltage	V _{IN}	0 : V _{DD} + V _{OH}	V
Operating Temperature	T _{OP}	0 to +50	°C
Storage Temperature	T _{STG}	-20 to +70	°C

Electrical Characteristics (T_A = 25°C)

Item	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V _{DD}	4.75	5.0	5.25	V
	GND	-	0	-	V
"H" Level Input Voltage (V _{IN} = 5.0V)	V _{IH}	2.2	-	-	V
"L" Level Input Voltage (V _{IN} = 5.0V)	V _{IL}	-	-	0.6	V
"H" Level Output Voltage (I _{OH} = 0.2 mA)	V _{OH}	2.4	-	-	V
"L" Level Output Voltage (I _{OL} = 1.2 mA)	V _{OL}	-	-	0.4	V
Power Consumption	P _D	-	5.0	-	mW

Optical Characteristics (T_A = 25°C, θ = 25°, θ = 0°)

Item	Symbol	Min.	Typ.	Max.	Unit
Viewing Angle	θ	10	25	40	degree
Contrast	γ	-	3.0	-	-
Turn On Time	t _{ON}	-	200	400	ms
Turn Off Time	t _{OFF}	-	250	400	ms

Note: Refer to Applications Section for the following definitions: (a) θ and (b) Viewing Angle; (c) Contrast; (d) Turn On and Turn Off Time

Connector Pin Assignment

Pin No.	Signal	Function
1	GND	Power Supply
2	V _{CC}	
3	V _{CC}	(V _{CC} Drive voltage (DV to V _{CC}))
4	H ₁	"H" Data Input "L" Command Input
5	R/W	"H" Data Read (Module - CPU) "L" Data Write (CPU - Module)
6	E	Enable Signal
7	DB0	Data Bus 8-bit Use
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	

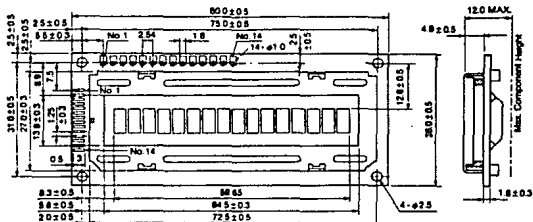


LCD Dot Matrix Module

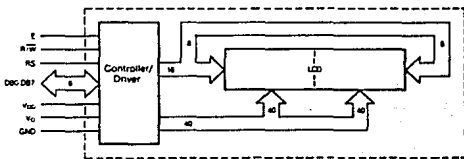
18 CHARACTERS x 1 LINE

AND671

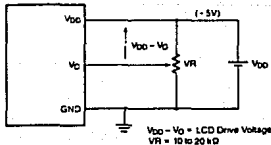
Dimensional Outline (in millimeters)



Block Diagram



Power Supply



Recommended Power Supply for LCD Drive (V_0)
 LCD Panel is driven by the voltage $V_{DD} - V_0$, so adjustable V_0 is required for contrast control and temperature compensation.

Temperature	V_0
0°C	0.0V
+25°C	5V
+50°C	1.0V



LCD Dot Matrix Modules

CHARACTER LCD MODULES INTERFACE DATA

Timing Characteristics (T_A = 25°C)

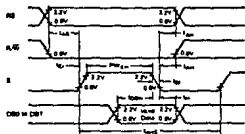
Data Write

Item	Symbol	Value		Unit
		Min.	Max.	
Enable Cycle Time	t _{cycle}	1000		ns
Enable Pulse Width	PW _{EN}	450		
Enable Rise/Fall Time	t _{r, f}		25	
Set Up Time	t _{su}	140		
Address Hold Time	t _{ah}	10		
Data Set Up Time	t _{dsw}	195		
Data Hold Time	t _{dh}	10		

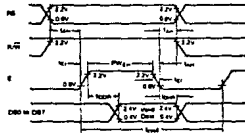
Data Read

Item	Symbol	Value		Unit
		Min.	Max.	
Enable Cycle Time	t _{cycle}	1000		ns
Enable Pulse Width	PW _{EN}	450		
Enable Rise/Fall Time	t _{r, f}		25	
Set Up Time	t _{su}	140		
Address Hold Time	t _{ah}	10		
Data Delay Time	t _{dca}		320	
Data Hold Time	t _{dch}	20		

Data Write



Data Read

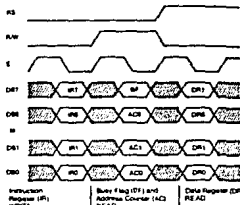


Data Transfer Example

The data can be sent in the form of either 2 cycles of 4-bit data or 1 cycle of 8-bit data so that it can be connected to both 4 and 8-bit CPU's.

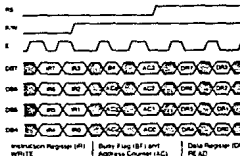
- (1) When data is 8 bits long, the data is transferred by using 8 data lines of DB0 to DB7.

8 bit Data Transfer Example



- (2) When data is 4 bits long, it is transferred by using only 4 lines of DB7 to DB4, DB3 to DB0 are not used. Data transfer between the module and a 4-bit CPU is completed when the higher order 4 bits are transferred first, followed by the lower order 4 bits.

4 bit Data Transfer Example





Function of Registers

1. Instruction Register and Data Register

The LCD module's built-in controller has two 8-bit registers, an Instruction Register (IR) and a Data Register (DR). IR stores instruction codes such as display clear and cursor shift, address information of display data RAM (DD RAM), and character generator RAM (CG RAM).

IR can be written to by a CPU, but a CPU cannot read IR.

The DR temporarily stores data to be written into the DD RAM or the CG RAM. Data written into the DR is automatically sent to the DD RAM or the CG RAM as an internal operation. The DR is also used for data storage when reading data from the DD RAM or the CG RAM. When address information is written into the IR, data is transferred to the DR from the DD RAM or the CG RAM as an internal operation. Then, the CPU reads the DR and data transfer is completed. After the CPU reads the DR, data of the DD RAM or the CG RAM at the next address is sent to DR for the next reading.

Register Selector (RS) signals select these two registers.

Table 1. Register Selection

RS	R/W	Operation
0	0	IR write as internal operations (display clear, etc.)
0	1	Read of a Busy Flag (DB7) and Address Counter (DD0 to DD6)
1	0	DR Write as internal operations (DR - CG or CG RAM)
1	1	DR Read as internal operations (DD or CG RAM - DR)

2. Busy Flag (BF)

When the Busy Flag is "1", the LCD module is in the internal operation mode, and the next instruction is not accepted at the time. As shown in Table 1, the Busy Flag is shown in DB7 when RS = 0 and R/W = 1. The next instruction must be written after checking that the Busy Flag is "0".

3. Address Counter (AC)

The address counter (AC) assigns DD and CG RAM address. When an instruction for address setting is written in IR, the address information is sent from IR to AC.

Selection of either the DD or CG RAM is also determined by an instruction. After writing into (or reading from) DD or CG RAM display data, AC is automatically incremented by +1 (or decremented by -1). Data in address counters (AC) are in DD6 to DD0 when RS = 0 and R/W = 1, as shown in Table 1.

4. Display Data RAM (DD RAM)

The display data RAM (DD RAM) stores display data represented in 8-bit character codes.

Relationship between the DD RAM address and display position on LCD Display is shown on page 210 and 211.

Commands

The command code refers to the signal through which the LCD module is accessed through the CPU. The LCD module begins operation upon receipt of the code input.

As the internal processing operation of the LCD module is started with a timing that does not affect the LCD display, the busy status continues longer than the CPU cycle time.

Under the busy status (when the busy flag is set to "1"), the LCD module does not execute any commands other than the busy flag read.

For this reason, the CPU has to verify that the busy flag is set to "0" prior to the input of the command code.

Table 2 shows the commands and the execution times for the commands. The commands can be divided into the following 4 types.

- Commands that designate the module functions such as display format, data length, etc.
- Commands that give internal RAM addresses.
- Commands that perform data transfer with internal RAM.
- Other commands.



LCD Dot Matrix Modules

CHARACTER LCD MODULES INTERFACE DATA

Table 2. Command List

Command	Command Code										Description	Execution Time (max) Mode (1)	Execution Time (max) Mode (2)	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0				
Clear Display Note (4)	C	0	0	0	0	0	0	0	0	0	1	Clear all of the display and return the cursor to the home position (Address 0)	184ms	48ms
Return home	0	0	0	0	0	0	0	0	0	1	X	Return the cursor to the home position (Address 0). Also return the display being shifted to the original position. DD RAM contents remain unchanged.	184ms	48ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S		Set the cursor move direction and specify to or not to shift the display. These operations are performed during data write.	40µs	120µs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B		Set ON/OFF of entire display (D), cursor ON/OFF (C), and banking of cursor position (B).	40µs	120µs
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	X	X		Move the cursor and shift the display without changing DD RAM contents.	40µs	120µs
Function Set	0	0	0	0	1	DL	N	F	X	X		Set interface data length (DL) number of display lines (L) and character font (F).	40µs	120µs
Set RAM Address	0	0	0	1						ACG		Set the CG RAM address. CG Ram data is sent and received after this setting.	40µs	120µs
Set DD RAM Address	0	0	1							ADD		Set the DD RAM address. DD RAM data is sent and received after this setting.	40µs	120µs
Read Busy Flag & Address	0	1	BF							AC		Read Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	40µs	120µs
Write Data to CG or DD RAM	1	0								Write Data		Write Data from DD RAM or CG RAM.	40µs	120µs
Read Data to CG or DD RAM	1	1								Read Data		Read Data from DD RAM or CG RAM.	40µs	120µs
	I/D = 1 Increment I/D = 0 Decrement S = 1 Accompanes display shift S/C = 1 Display shift S/C = 0 Cursor move R/L = 1 Shift to the right R/L = 0 Shift to the left DL = 1 8 bits DL = 0 4 bits N = 1 2 lines N = 0 1 line F = 1 5 × 10 dots F = 0 5 × 7 dots BF = 1 Internally operating BF = 0 Can accept instruction										DD RAM	Display Data RAM		
											CG RAM	Character Generator RAM		
											ACG	CG RAM Address		
											ADD	DD RAM Address		
												Corresponds to Cursor Address		
											AC	Address Counter used for both DD and CG RAM Addresses		

X Don't care

Notes

- 1 Applies to AND61 AND671 AND681 AND691 AND511
- 2 Applies to AND591 AND791 AND731 AND771
- 3 Instruction Cursor and Display Shift are enabled for the AND671
- 4 The repeat time interval of instruction Clear Display must be 13 ms min (5 × 7 dot font) and 18 ms min (5 × 10 dot font)

AND

LCD Dot Matrix Modules

CHARACTER LCD MODULES INTERFACE DATA

1. Clear Display

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	0	0	0	0	1

Write space code "20" (Hexadecimal) into all the DD RAM addresses. The cursor returns to address 0 (DD RAM Address = "00H") and display, if it has been shifted, it returns to the original position. In other words, display disappears and the cursor goes to the left edge of the first line.

2. Return Home

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	0	0	0	1	X

Return the cursor to character position 1 (DD RAM Address = "00H") and returns the display to the original position if it has been shifted (5 in the instruction register = 1). The DD RAM contents remain unchanged.

3. Entry Mode Set

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	0	0	1	I/D	S

I/D: Increment (I/D=1) or decrement (I/D=0) the DD RAM address by one upon writing a character code into the DD RAM or reading a character code from the DD RAM. The cursor moves to the right when I/D=1, and to the left when I/D=0.

S: Shift the entire display to the right (when I/D=0, S=1) or the left (when I/D=1, S=1) when writing to the DD RAM. Therefore, the cursor looks as if it stopped still and display only moves. Display is not shifted when reading from the DD RAM. Display is not shifted when S=0.

4. Display ON/OFF Control

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	0	1	D	C	B

D: Display is turned ON when D=1 and OFF when D=0. When display is turned off due to D=0, the display data remains in the DD RAM and they can be displayed immediately by setting D=1.

C: The cursor is displayed when C=1 and not displayed when C=0. Even if the cursor disappears, function of I/D, etc. does not change during "display data write". The cursor is displayed at the 8th line when the 5 x 7 dots character font is selected.

B: The character at the cursor position blinks when B=1. The blink is gone by switching between all black dots and display characters at 0.4 second interval. The cursor and the blink can be set concurrently.

X: Don't care

5. Cursor or Display Shift

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	0	1	S/C	R/L	X

Shift the cursor position or display position to the right or the left without writing or reading the display data. This function can be used for correction or search of display.

S/C	R/L	Function
0	0	Shift the cursor position to the left. (AC is decremented by one.)
0	1	Shift the cursor position to the right. (AC is incremented by one.)
1	0	Shift the entire display to the left. The cursor follows the display shift.
1	1	Shift the entire display to the right. The cursor follows the display shift.

6. Function Set

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	0	1	DL	N	F	X

DL: Set the interface data length. Data is sent or received in 8-bit length (D87 to D80) when DL=1 and 4-bit length (D87 to D84) when DL=0. When 4-bit length is selected, data must be sent or received twice.

N: Set number of display lines.

F: Set character font. The 5 x 7 dots character font is selected when F=1 while 5 x 10 dots character font is selected when F=0 and N=0.

N	F	No. of Display Lines	Character Font	Duty Ratio	Module Type No.
0	0	1	5x7 Dots	1/8	AND241, AND691, AND601
0	1	1	5x10 Dots	1/11	AND241, AND691, AND601
1	0	2	5x7 Dots	1/16	AND671, AND491, AND501, AND771, AND581
1	0	4	4x7 Dots	1/16	AND721, AND721

7. Set CG RAM Address

RS	R/W	D87	D86	D85	D84	D83	D82	D81	D80
Code	0	0	0	1	A	A	A	A	A

Set the CG RAM address to a binary number of AAAAAA in the address counter. After execution of this instruction, all the data from MPU is written into the CG RAM and all the data is read from CG RAM.

AND

LCD Dot Matrix Modules

CHARACTER LCD MODULES INTERFACE DATA

9. Set DD RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	An	A	A	A	A	A	A

Set the DD RAM address to a binary number of AnAAAAAA in the address counter. (An = 0 for the first line, An = 1 for the second line). After execution of this instruction, all the data from MPU is written into the DD RAM and all the data is read from DD RAM.

9. Read Busy Flag and Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	A	A	A	A	A	A	A

Read Busy Flag (BF) and the value of the address counter (AAAAAAA). BF = 1 indicates that an internal operation is going on; the next command is not accepted until BF becomes "0". It is necessary to check the BF status before the next write operation. The address counter is used for the CG or DD RAM address.

10. Write Data to CG RAM or DD RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D	D	D	D	D	D	D	D

Write binary 8 bit data DDDDDDDD to the CG RAM or the DD RAM. Whether the CG RAM or the DD RAM is to be written is determined by the previous designation (CG RAM address setting or DD RAM address setting). After writing, the address is automatically incremented or decremented by one according to entry mode. Display shift also follows the entry mode.

11. Read Data from CG RAM or DD RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D	D	D	D	D	D	D	D

Read binary 8-bit data DDDDDDDD from the CG RAM or the DD RAM. Whether the CG RAM or the DD RAM is to be read is determined by the previous designation. Prior to executing this read command, either the CG RAM address set command or the DD RAM address set command must be executed. If it is not done, the first data read is invalid, and the second data read of the next address can be read normally. After reading, the address is automatically incremented or decremented by one according to the entry mode. However, display shift is not performed regardless of entry mode.

Character Patterns and Character Codes**1. Character Generator ROM (CG ROM)**

The character generator ROM generates 5 x 7 dot (160 kinds) character patterns or 5 x 10 dot (32 kinds) character patterns from an 8-bit DD RAM character code signal.

When the 8-bit character code of the CG ROM is written into the DD RAM, the character pattern of the CG ROM corresponding to the code is displayed on the LCD display position corresponding to the DD RAM address. Table 3 shows the relation between character patterns and character codes.

Note:

AND671, AND491, AND501, AND591, AND771, AND731 and AND721 can only use 5 x 7 dot character patterns.

2. Character Generator RAM (CG RAM)

The character generator RAM is used for original character patterns other than for the CG ROM. The CG RAM has the capacity (64 bytes = 512 bits) to write 8 types of character patterns with 5 x 7 font, and 4 types with 5 x 10 font. When displaying character patterns stored in the CG RAM, write 8-bit character codes (00 to 07 or 02 to 0F, hex.) on the left side as shown in Table 3.

Table 4 shows the relation between CG RAM addresses and data and display patterns for 5 x 7 dots.

Table 5 shows the relation between CG RAM addresses and data and display patterns for 5 x 10 dots.

AND

LCD Dot Matrix Modules

CHARACTER LCD MODULES INTERFACE DATA

Table 3. Character Pattern and Character Code

Upper 4 bit Lower 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
XXXX0000	CG RAM (1)												
XXXX0001	(2)												
XXXX0010	(3)												
XXXX0011	(4)												
XXXX0100	(5)												
XXXX0101	(6)												
XXXX0110	(7)												
XXXX0111	(8)												
XXXX1000	(1)												
XXXX1001	(2)												
XXXX1010	(3)												
XXXX1011	(4)												
XXXX1100	(5)												
XXXX1101	(6)												
XXXX1110	(7)												
XXXX1111	(8)												



Relation between CG RAM Address and Character Code (DD RAM) and Character Pattern (CG RAM Data)

Table 4. 5 x 7 Dots Character Pattern

Character Code (DD RAM Data)	CG RAM Address	Character Pattern (CG RAM Data)
7 6 5 4 3 2 1 0	5 4 3 2 1 0	7 6 5 4 3 2 1 0
0 0 0 0 X 0 0 0	0 0 0	X X X 0 0 0 0 0
	0 0 1	X X X 0 0 0 0 0
	0 1 0	X X X 0 1 0 0 0
	0 0 0 0 1 1	X X X 0 1 0 0 1
	1 0 0	X X X 1 0 0 0 0
0 0 0 0 X 0 0 1	0 0 1	X X X 0 0 0 0 0
	0 1 0	X X X 0 0 0 0 0
	0 0 1 0 1	X X X 0 1 0 0 1
	1 0 0	X X X 1 0 0 0 0
	1 0 1	X X X 0 1 0 0 1

X = Don't care

Notes:

- Character code bits 0 to 2 correspond to CG RAM address bits 3 to 5 (3 bits 8 types)
- CG RAM address bits 0 to 2 designate character pattern line position. The 0th line is the cursor position and the display is presented in logical OR with cursor
- Character pattern row positions correspond to CG RAM data bits 0 to 4 as shown in the figure (bit 4 being at the left end). Since the CG RAM data bits 5 to 7 are not used for the display, they can be used as general data RAM
- As shown in Table 4, CG RAM character patterns are selected when character code bits 4 to 7 are all "0". However, since character code bit 3 is the "Don't Care bit", "0" display in the character pattern for example, is selected by character code "00" or "08"
- "1" for CG RAM data corresponds to selection for display and "0" for non-selection

Table 5. 5 x 10 Dots Character Pattern

Character Code (DD RAM Data)	CG RAM Address	Character Pattern (CG RAM Data)
7 6 5 4 3 2 1 0	5 4 3 2 1 0	7 6 5 4 3 2 1 0
0 0 0 0 X 0 0 X	0 0 0 0	X X X 0 0 0 0 0
	0 0 0 1	X X X 0 0 0 0 0
	0 0 1 0	X X X 0 1 0 0 0
	0 0 1 1	X X X 0 1 0 0 0
	0 1 0 0	X X X 1 0 0 0 0
0 0 0 0 X 0 0 1	0 0 0 1 0 1	X X X 0 1 0 0 1
	0 1 0 0	X X X 1 0 0 0 0
	1 0 0	X X X 0 1 0 0 1
	1 0 1	X X X 0 1 0 0 1
	1 1 0	X X X 1 0 0 0 0

X = Don't care

Notes:

- Character code bits 1 and 2 correspond to CG RAM address bits 4 and 5 (2 bits 4 types)
- CG RAM address bits 0 to 3 designate character pattern line position. The 1th row is the cursor position
- Character pattern row positions correspond to CG RAM data bits 0 to 4, as shown in the figure (bit 4 being at the left end)
- As shown in Table 5, CG RAM character patterns are selected when character code bits 4 to 7 are all "0". However, since character code bit 0 and 3 are the "Don't Care bit", "0" display in the character pattern, for example, is selected by character codes "00", "01", and "08" and "09"
- "1" for CG RAM data corresponds to selection for display and "0" for non-selection

Initialization (Reset)

1. Automatic Initialization

The LCD module is automatically initialized when the power is turned on (using internal reset circuit). The following commands are executed in initialization. The busy flag is kept in the busy state (BF #ED 1) until initialization ends. The busy state is kept about 10ms after VDD level reaches 4.5V

(a) Clear display

(b) Function set

- Data length of interface with MPU: 8 bit (DL = 1)
- LCD: 1-line display (N = 0)
- Character font: 5 x 7 dots (F = 0)

(c) Display ON/OFF control

- Display: Display OFF (D = 0)
- Cursor: Cursor OFF (C = 0)
- Blank: Blank OFF (B = 0)

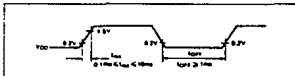
(d) Entry mode set

- Address counter: Increment + 1 (I/D = 1)
- Display shift: No shift (S = 0)

(e) DD RAM is selected

Above function set command in automatic initialization does not always meet each module, in this case it is necessary to reset "Function Set" command (refer to Command section, page 5-25)

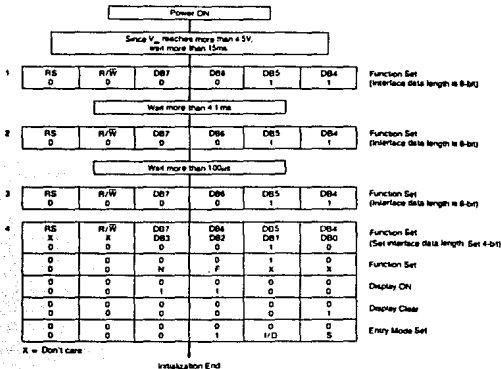
Note: Power on timing shown below is necessary in order to perform automatic initialization. When the above power supply condition is not satisfied, the internal reset circuit will not operate normally. In this case, perform the initialization by sending commands from CPU after turning the power ON.





2. Manual Initialization Procedure

(a) When the interface data length is 4-bit



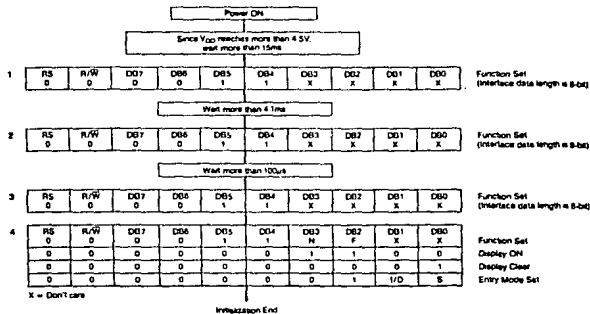
Notes:

- (1) Before initialize step 1, 2 and 3, can not check busy flag
- (2) After initialize step 4, cannot change function set Mode, number of display lines and character font.



2. Manual Initialization Procedure (Continued)

(b) When the interface data length is 8-bit



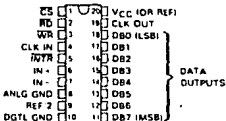
Notes:

- (1) Before make step 1, 2 and 3, can not check busy flag
- (2) After make step 4, cannot change function set, number of display lines and character font

ADC0804, ADC0804C
8-BIT ANALOG-TO-DIGITAL CONVERTER
WITH DIFFERENTIAL INPUTS
07766 OCTOBER 1982 - REVISED OCTOBER 1988

- 8-Bit Resolution
- Ratiometric Conversion
- 100- μ s Conversion Time
- 135-ns Access Time
- No Zero Adjust Requirement
- On-Chip Clock Generator
- Single 5-V Power Supply
- Operates with Microprocessor or as Stand-Alone
- Designed to be Interchangeable with National Semiconductor and Signetics ADC0804

N DUAL IN LINE PACKAGE
 (TOP VIEW)




description

The ADC0804 is a CMOS 8 bit successive-approximation analog-to-digital converter that uses a modified potentiometric (256R) ladder. The ADC0804 is designed to operate from common microprocessor or control buses, with the three-state output latches driving the data bus. The ADC0804 can be made to appear to the microprocessor as a memory location or an I/O port. Detailed information on interfacing to most popular microprocessors is readily available from the factory.

A differential analog voltage input allows increased common-mode rejection and offset of the zero-input analog voltage value. Although a reference input (REF/2) is available to allow 8-bit conversion over smaller analog voltage spans or to make use of an external reference, ratiometric conversion is possible with the REF/2 input open. Without an external reference, the conversion takes place over a span from VCC to analog ground (ANLG GND). The ADC0804 can operate with an external clock signal or, with an additional resistor and capacitor, can operate using an on-chip clock generator.

The ADC0804 is characterized for operation from -40°C to 85°C. The ADC0804C is characterized for operation from 0°C to 70°C.

PREDICTED DATA documents contain information derived as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.


**TEXAS
 INSTRUMENTS**
POST OFFICE BOX 518012 • DALLAS, TEXAS 75245-1202

Copyright © 1988, Texas Instruments Incorporated

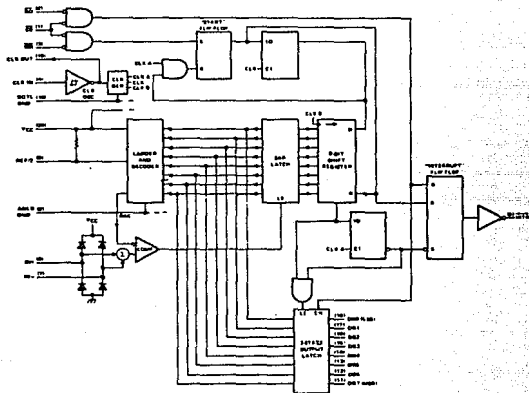
2-29

Data Sheets 2

ADC08041, ADC0804C
8-BIT ANALOG-TO-DIGITAL CONVERTER
WITH DIFFERENTIAL INPUTS

functional block diagram (positive logic)

2
 Data Sheets



2-30

TEXAS
 INSTRUMENTS

POST OFFICE BOX 6552-2 • DALLAS, TEXAS 75265

ADC08041, ADC0804C 8-BIT ANALOG-TO-DIGITAL CONVERTER WITH DIFFERENTIAL INPUTS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	6.5 V
Input voltage range: CS, RD, WR	-0.3 V to 18 V
Output voltage range	-0.3 V to $V_{CC} + 0.3$ V
other inputs	-0.3 V to $V_{CC} - 0.3$ V
Operating free-air temperature range: ADC08041	-40°C to 85°C
ADC0804C	0°C to 70°C
Storage temperature range	-65°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to digital ground (DGT1, CND1 with DGT1, GND and ANLG GND connected together) unless otherwise noted.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	6.3	V
Voltage at REF/2, $V_{REF/2}$ (see Note 2)	0.25	3.5		V
High-level input voltage at CS, RD or WR, V_{IH}	2	15		V
Low-level input voltage at CS, RD or WR, V_{IL}		0.8		V
Analog ground voltage (see Note 3)	-0.05	0	1	V
Analog input voltage (see Note 4)	-0.05		$V_{CC} - 0.05$	V
Clock input frequency, f_{CLOCK} (see Note 5)	100	640	1450	kHz
Duty cycle for f_{CLOCK} , 8-640 kHz (see Note 5)	40		60	%
Pulse duration (clock input (high or low)) for $f_{CLOCK} < 640$ kHz, $t_{PH}(CLK)$ (see Note 5)	275	781		ns
Pulse duration, WR input low (start conversion), $t_{WL}(WR)$	100			ns
Operating free-air temperature, T_A	ADC08041	-40	85	°C
	ADC0804C	0	70	°C

NOTES: 2 The internal reference voltage is equal to the voltage applied to REF/2, or approximately equal to one-half of the V_{CC} when REF/2 is left open. The voltage at REF/2 should be one-half the full scale differential input voltage between the analog inputs. Thus, the differential input voltage when REF/2 is open and $V_{CC} = 5$ V is 0 to 5 V. $V_{REF/2}$ for an input voltage range from 0.5 V to 3.5 V (full scale differential voltage of 3 V) is 1.5 V.

3 These values are with respect to DGT1, GND.

4 When the differential input voltage ($V_{IN+} - V_{IN-}$) is less than or equal to 0 V, the output code is 0000 0000.

5 Total unadjusted error is specified only at an f_{CLOCK} of 640 kHz with a duty cycle of 40% to 60% (pulse duration 625 ns to 837 ns). For frequencies above this limit or pulse duration below 625 ns, error may increase. The duty cycle limits should be observed for an f_{CLOCK} greater than 640 kHz. Below 640 kHz, this duty cycle limit can be exceeded provided $t_{PH}(CLK)$ remains within limits.

ADC08041, ADC0804C
8-BIT ANALOG-TO-DIGITAL CONVERTER
WITH DIFFERENTIAL INPUTS
electrical characteristics over recommended operating free-air temperature range, $V_{CC} = 5$ V, $f_{clock} = 640$ kHz, $V_{REF/2} = 2.5$ V (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TPY ¹	MAX	UNIT	
V_{OH}	High level output voltage	All outputs DB and INTR	$V_{CC} = 4.75$ V, $I_{OH} = -350$ μ A	2.4		V	
			$V_{CC} = 4.75$ V, $I_{OH} = -10$ μ A	4.6			
V_{OL}	Low level output voltage	Data outputs	$V_{CC} = 4.75$ V, $I_{OL} = 1.6$ mA		0.4	V	
		INTR output	$V_{CC} = 4.75$ V, $I_{OL} = 1$ mA		0.4		
		CLK OUT	$V_{CC} = 4.75$ V, $I_{OL} = 360$ μ A		0.4		
V_T^+	Clock positive going threshold voltage		2.7	3.1	3.5	V	
V_T^-	Clock negative going threshold voltage		1.5	1.8	2.1	V	
$V_T^+ - V_T^-$	Clock input hysteresis		0.6	1.3	2	V	
I_{IH}	High level input current			0.005	1	μ A	
I_{IL}	Low level input current			-0.005	-1	μ A	
I_{OZ}	Off state output current	$V_O = 0$			-3	μ A	
		$V_O = 5$ V			3		
I_{OHS}	Short circuit output current	Output high	$V_O = 0$, $T_A = 25^\circ\text{C}$	-4.5	-6	mA	
I_{OLS}	Short circuit output current	Output low	$V_O = 5$ V, $T_A = 25^\circ\text{C}$	8	18	mA	
I_{CC}	Supply current plus reference current	REF/2 open, CS at 5 V, $T_A = 25^\circ\text{C}$		1.9	2.6	mA	
REF/2	Input resistance to reference ladder	See Note 6		1	1.2	k Ω	
C_i	Input capacitance (control)				5	7.5	pF
C_o	Output capacitance (DB)				5	7.5	pF

operating characteristics over recommended operating free-air temperature range, $V_{CC} = 5$ V, $V_{REF/2} = 2.5$ V, $f_{clock} = 640$ kHz (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TPY ¹	MAX	UNIT
	Supply voltage variation error (See Notes 2 and 3)	$V_{CC} = 4.5$ V to 5.5 V		$\pm 1/16$	$\pm 1/8$	LSB
	Total unadjusted error (See Notes 3 and B1)	$V_{REF/2} = 2.5$ V			± 1	LSB
	DC common mode error (See Note B1)			$\pm 1/16$	$\pm 1/8$	LSB
t_{EN}	Output enable time	$C_L = 100$ pF		135	200	ns
t_{DS}	Output disable time	$C_L = 10$ pF, $R_L = 10$ k Ω		125	200	ns
$t_{RSTINTR}$	Delay time to reset INTR			300	450	ns
t_{CONV}	Conversion cycle time (See Note B1)	$f_{clock} = 100$ kHz to 1.46 MHz	65%		72%	clock cycles
	Conversion time			103	114	μ s
CR	Free running conversion rate	INTR connected to WR, CS at 0 V			8827	convs

¹All typical values are at $T_A = 25^\circ\text{C}$.

NOTES 2 The internal reference voltage is equal to the voltage applied to REF-2, or approximately equal to one half of the V_{CC} when REF-2 is left open. The voltage at REF-2 should be one half the full scale differential input voltage between the analog inputs. Thus, the differential input voltage when REF-2 is open and $V_{CC} = 5$ V is 0 to 5 V. $V_{REF/2}$ for an input voltage range from 0.5 V to 3.5 V, full scale differential voltage of 3 V, is 1.5 V.

3 The resistance is calculated from the current drawn from a 5 V supply applied to pins 5 and 9.

4 These parameters are specified for the recommended analog input voltage range.

5 All errors are measured with reference to an ideal straight line through the end points of the analog to digital transfer characteristic.

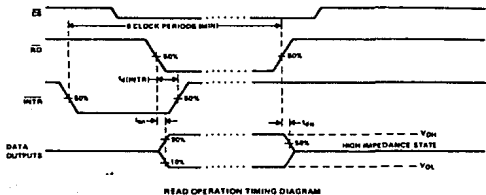
6 Although internal conversion is completed in 64 clock periods, a CS or WR low to high transition is followed by 1 to 8 clock periods before conversion starts. After conversion is completed, part of another clock period is required before a high to low transition of INTR completes the cycle.

2

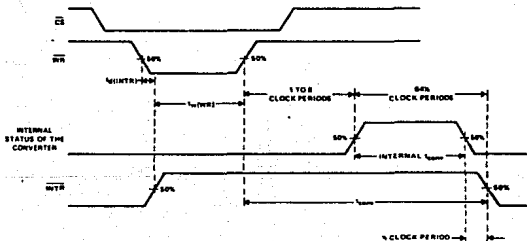
Data Sheets

ADC08041, ADC0804C
8-BIT ANALOG-TO-DIGITAL CONVERTER
WITH DIFFERENTIAL INPUTS

timing diagrams



READ OPERATION TIMING DIAGRAM



WRITE OPERATION TIMING DIAGRAM

2
 Data Sheets

TEXAS
 INSTRUMENTS

POST OFFICE BOX 455012 • DALLAS, TEXAS 75245

2-33

ADC0804, ADC0804C 8-BIT ANALOG-TO-DIGITAL CONVERTER WITH DIFFERENTIAL INPUTS

PRINCIPLES OF OPERATION

The ADC0804 contains a circuit equivalent to a 256-resistor network. Analog switches are sequenced by successive approximation logic to match an analog differential input voltage ($V_{IN+} - V_{IN-}$) to a corresponding tap on the 256-resistor network. The most-significant bit (MSB) is tested first. After eight comparisons (64 clock periods), an 8-bit binary code (1111 1111 = full scale) is transferred to an output latch and the interrupt (INTR) output goes low. The device can be operated in a free-running mode by connecting the INTR output to the write (WR) input and holding the conversion start (CS) input at a low level. To ensure start-up under all conditions, a low-level WR input is required during the power-up cycle. Taking CS low anytime after that will interrupt a conversion in process.

When the WR input goes low, the ADC0804 successive approximation register (SAR) and 8-bit shift register are reset. As long as both CS and WR remain low, the ADC0804 remains in a reset state. One to eight clock periods after CS or WR makes a low-to-high transition, conversion starts.

When the CS and WR inputs are low, the start flip-flop is set and the interrupt flip-flop and 8-bit register are reset. The next clock pulse transfers a logic high to the output of the start flip-flop. The logic high is ANDed with the next clock pulse, placing a logic high on the reset input of the start flip-flop. If either CS or WR have gone high, the set signal to the start flip-flop is removed, causing it to be reset. A logic high is placed on the D input of the 8-bit shift register and the conversion process is started. If the CS and WR inputs are still low, the start flip-flop, the 8-bit shift register, and the SAR remain reset. This action allows for wide CS and WR inputs with conversion starting from one to eight clock periods after one of the inputs goes high.

When the logic high input has been clocked through the 8-bit shift register, completing the SAR search, it is applied to an AND gate controlling the output latches and to the D input of a flip-flop. On the next clock pulse, the digital word is transferred to the three-state output latches and the interrupt flip-flop is set. The output of the interrupt flip-flop is inverted to provide an INTR output that is high during conversion and low when the conversion is completed.

When a low is at both the CS and RD inputs, an output is applied to the DBO through DB7 outputs and the interrupt flip-flop is reset. When either the CS or RD inputs return to a high state, the DBO through DB7 outputs are disabled (returned to the high-impedance state). The interrupt flip-flop remains reset.

BIBLIOGRAFÍA.

Capítulo 1.

Arthur E. Cote & Jim L. Linville. 1986.

FIRE PROTECTION HANDBOOK (SIXTEENTH EDITION).

National Fire Protection Association (NFPA), EE. UU.

Sección 5, Fire hazards of materials 5/39-5/58.

Capítulo 5, Gases (Wilbur L. Walls).

**GAS DATA. 1ST SOLID STATE ELECTROLYTIC CELL GAS SENSOR
FOR AREA AIR QUALITY & SAFETY APPLICATIONS.**

International Sensor Technology, EE. UU.

Gas data. p.p. 2-14.

810, COMBUSTIBLE GAS DETECTION SYSTEM, OPERATION MANUAL.

Rexnord, Gas Detection Products. EE. UU.

Capítulo 4, Theory of operation, p.p. 4/1.

Sección 1, Basic definitions.

Capítulo 2.

Morris Katz. 1977.

METHODS OF AIR SAMPLING AND ANALYSIS.

American Public Health Association (APHA) Intersociety Committee. EE. UU.

Parte I, capítulo 12, Photometric Methods for Chemical Analysis. p.p. 68-76.

Parte I, capítulo 13, Ultraviolet Absorption Spectroscopy. p.p. 76-79.

Parte I, capítulo 14, Infrared Absorption Spectroscopy. p.p. 79-84.

Parte I, capítulo 15, Atomic Absorption Spectrophotometry. p.p. 84-88.

Parte I, capítulo 16, Gas Chromatography. p.p. 88-98.

Harry N. Norton. 1984.

SENSORES Y ANALIZADORES. COLECCION ELECTRONICA/INFORMATICA.

Editorial Gustavo Gili. España.

Capítulo 8, Propiedades y análisis de composiciones químicas. p.p. 541-565.

Inciso 6, Analizadores fotométricos.

Inciso 7, Analizadores cromatográficos.

Inciso 8, Analizadores espectroradiométricos.

L. Jack Bradshaw. 1966.

INTRODUCTION TO MOLECULAR BIOLOGICAL TECHNIQUES.

Prentice Hall. EE. UU.

Capítulo 3, An introduction to spectroscopy. p.p. 24-34.

David Abbott, R. S. Andrews. 1973.

INTRODUCCION A LA CROMATOGRAFIA.

Editorial Alhambra. España.

Capítulo 6, Cromatografía de gases. p.p. 73-86.

John R. Dyer. 1973.

**APLICACIONES DE ESPECTROSCOPIA DE ABSORCIÓN EN
COMPUESTOS ORGANICOS.**

Editorial Prentice Hall Internacional. España.

Capítulo 1, Introducción. p.p. 1-3.

Capítulo 2, Espectroscopía ultravioleta. p.p. 4-8.

Inciso 1, Sistemas de medición.

Inciso 2, Excitación electrónica.

Capítulo 3, Espectroscopía infrarroja. p.p. 22-28.

Inciso 1, Vibración molecular.

Inciso 2, Sistema de medición.

Capítulo 3.

Morris Katz. 1977.

METHODS OF AIR SAMPLING AND ANALYSIS.

American Public Health Association (APHA) Intersociety Committee. EE. UU.

Parte I, capítulo 5, Sampling and storage of gases and vapors. p.p. 38-48.

Jack P. Holman. 1986.

MÉTODOS EXPERIMENTALES PARA INGENIEROS.

Editorial McGraw-Hill. México.

Capítulo 13, Muestreo y medición de la contaminación del aire.

Inciso 4, Tren general de muestreo de aire. p.p. 495-496.

Capítulo 4.

José Mompín Poblet. 1990.

TRANSDUCTORES Y MEDIDORES ELECTRONICOS.

Marcombo, Boixeraux Editores. España.

Parte I, Transductores para detección y análisis de gases. p.p. 144-155.

Harry N. Norton. 1984.

SENSORES Y ANALIZADORES. COLECCION ELECTRONICA/INFORMATICA.

Editorial Gustavo Gili. España.

Capítulo 8, Propiedades y análisis de composiciones químicas. p.p. 519-539.

Inciso 2, Analizadores electrométricos.

Inciso 3, Sensores de gas resistivos.

Inciso 4, Analizadores térmicos.

Arthur E. Cote & Jim L. Linville. 1986.

FIRE PROTECTION HANDBOOK (SIXTEENTH EDITION).

National Fire Protection Association (NFPA), EE. UU.

Sección 16, Fire alarm systems, detection systems, and guard service, 16/40-16/45.

Capítulo 5, Gas and vapor testing.

HAZARDOUS GAS INSTRUMENTS & DATA.

International Sensor Technology. EE. UU.

Sección 1, Introducing IST's solid state sensor. p.p. 1-7.

Sección 2, Operating principle of other gas monitoring instruments. p.p. 8-16.

Bukowski W. Richard, O'Laughlin J. Robert. 1988.

FIRE ALARM SIGNALING SYSTEMS HANDBOOK.

National Fire Protection Association & Society of Fire Protection Engineers.
EE. UU.

Capítulo 3, Signal Initiation. p.p. 58-61.

Gas sensing fire detectors.

Jack P. Holman. 1986.

METODOS EXPERIMENTALES PARA INGENIEROS.

Editorial McGraw-Hill. México.

Capítulo 13, Muestreo y medición de la contaminación del aire. p.p. 491-492.

Inciso 2, Unidades de la medición de la contaminación.

Sears-Zemanski. 1981.

FISICA GENERAL.

Editorial Aguilar. España.

Capítulo 16, Cantidad de calor. p.p. 290-291.

Inciso 7, Calor de combustión.

Capítulo 17, Propagación del calor. p.p. 299-302.

Inciso 1, Conducción.

Capítulo 5.

Willian David Cooper. 1989.

INSTRUMENTACION ELECTRONICA Y MEDICIONES.

Editorial Prentice Hall. México.

Capítulo 14, Sistemas de adquisición de datos digitales y analógicos. p.p. 452-455.

Inciso 1, Sistemas de instrumentación.

Arthur B. Williams. 1988.

**CIRCUITOS LOGICOS Y CONVERSION DE A/D Y D/A. SERIE DE
CIRCUITOS INTEGRADOS.**

Editorial McGraw-Hill. México.

Capítulo 1, Conversión A/D y D/A. p.p. 53-56.

Inciso 7, Sistemas completos de adquisición de datos.

Capítulo 6.

José Mompín Poblet. 1984.

MICROPROCESADORES Y MICROCOMPUTADORES. SERIE MUNDO ELECTRONICO.

Marcombo, Boixeraux Editores. México.

Proceso de diseño y realización de sistemas con microprocesadores. p.p.287-289.

La elección de un microprocesador adecuado.

José Mompín Poblet. 1984.

INTERCONEXION DE PERIFERICOS A MICROPROCESADORES. SERIE MUNDO ELECTRONICO.

Marcombo, Boixeraux Editores. México.

Capítulo 5, Entradas-salidas en serie. p.p. 100-113.

Capítulo 7, Convertidores A/D y D/A. p.p. 133-146.

SPECIFICATION DATA.

COMBUSTIBLE GAS SENSOR.

Detector Electronics Corporation. 1989. EE. UU.

Description, features and specifications.

SPECIFICATION DATA.

UNITIZED COMBUSTIBLE GAS DETECTION SYSTEM.

C7061C/U8700 DETECTOR/CONTROLLER.

Detector Electronics Corporation. 1987. EE. UU.

Application.

Description of C7061C detector.

Specifications.

SPECIFICATION DATA.
COMBUSTIBLE GAS DETECTION SYSTEM.
C7061C DETECTOR, R8460 CONTROLLER.
Detector Electronics Corporation. 1987. EE. UU.
Application.
System description, Combustible gas detector.
Specifications.

Capítulo 7

James L. Antonakos. 1990.
THE 68000 MICROPROCESSOR. HARDWARE AND SOFTWARE
PRINCIPLES AND APPLICATIONS.
Maxwell Macmillan International Editions. EE. UU.

Wunnavu V. Subbarao. 1991.
16/32-BIT MICROPROCESSORS 68000/68010/68020. SOFTWARE,
HARDWARE, AND DESIGN APPLICATIONS.
Maxwell Macmillan International Editions. EE. UU.

William Cramer, Gerry Kane. 1986.
68000 MICROPROCESSOR HANDBOOK. INCLUDES 68008, 68010 &
68020.
Osborne McGraw-Hill. EE. UU.

MC68000, MC68008, MC68010, MC68HC000. 8-/16-/32-BIT
MICROPROCESSOR USER'S MANUAL.
Motorola. 1990. EE. UU.

M68000 PROGRAMER'S REFERENCE MANUAL.

Motorola. 1989. EE. UU.

TELECOMMUNICATIONS DEVICE DATA.

Motorola. 1989. EE. UU.

8-BIT MICROPROCESSOR & PERIPHERICAL DATA.

Motorola. 1983. EE. UU.

LED AND LCD PRODUCTS. DISPLAY PRODUCTS CATALOG 1991.

AND. 1991. EE. UU.

CMOS LOGIC DATABOOK

National Semiconductors. 1989. EE. UU.

LS/S/TTL LOGIC DATA BOOK

National Semiconductors. 1989. EE. UU.

MEMORY DATA BOOK

National Semiconductors. 1990. EE. UU.

LINEAR CIRCUITS

DATA ACQUISITION AND CONVERSION

DATA BOOK VOL. 2

Texas Instruments. 1989. EE. UU.

**LINEAR CIRCUITS
AMPLIFIERS, COMPARATORS, AND SPECIAL FUNCTIONS.
DATA BOOK VOL. 1**
Texas Instruments. 1989. EE. UU.

INDICE.

INTRODUCCIÓN.	1
PRINCIPALES GASES LETALES DE USO INDUSTRIAL. CASOS DE ESTUDIO.	4
I.Introducción.	4
II.Definición de Gas.	5
III.Clasificación de los Gases.	6
A.Clasificación por propiedades químicas.	6
1.Gases inflamables.	7
2.Gases no inflamables.	7
3.Gases reactivos.	8
4.Gases tóxicos.	9
B.Clasificación por propiedades físicas.	9
1.Gases comprimidos.	10
2.Gases licuados.	11
3.Gases criogénicos.	11
C.Clasificación por su uso.	12
1.Gases combustibles.	12
2.Gases industriales.	13
3.Gases médicos.	13
IV.Peligros en el manejo de los gases.	13
A.Gases encerrados en un contenedor.	14
1.Gases comprimidos.	14
<u>a.Ley de Boyle.</u>	15
<u>b.Ley de Charles.</u>	15

2.Gases licuados y crlogénicos.	16
B.Fallas en contenedores.	18
1.Ruptura de contenedores.	18
a.Gases comprimidos.	18
b.Gases licuados.	19
2.Combustión de gases dentro de contenedores.	20
C.Gases que escapan de un contenedor.	21
1.Gases tóxicos o venenosos.	21
2.Oxígeno y otros gases oxidantes.	22
3.Gases licuados y crlogénicos.	22
4.Gases inflamables.	22
a.Explusiones por combustión.	23
b.Incendios de Gases Inflamables.	24
V.Casos de Estudio.	25
A.Acetileno.	25
1.Clasificación.	25
2.Propiedades químicas.	25
3.Utilización.	26
B.Amoníaco anhidro.	26
1.Clasificación.	26
2.Propiedades químicas.	26
3.Utilización.	26
C.Dióxido de carbono.	26
1.Clasificación.	26
2.Propiedades químicas.	27
3.Utilización.	27
D.Cloro.	27
1.Clasificación.	27
2.Propiedades químicas.	27
3.Utilización.	27
E.Etileno.	28
1.Clasificación.	28
2.Propiedades químicas.	28
3.Utilización.	28

F.Hidrógeno.	28
1.Clasificación.	28
2.Propiedades químicas.	28
3.Utilización.	28
G.Gas natural licuado (GNL).	29
1.Clasificación.	29
2.Propiedades químicas.	29
3.Utilización.	29
H.Gas licuado de petróleo (GLP).	29
1.Clasificación.	29
2.Propiedades químicas.	29
3.Utilización.	29
I.Oxígeno.	30
1.Clasificación.	30
2.Propiedades químicas.	30
3.Utilización.	30

SISTEMAS PARA EL ANÁLISIS DE GASES.

I.Introducción.	31
II.Fotometría.	32
A.Principio del método.	32
B.El Fotómetro.	36
1.La fuente de energía radiante.	37
2.Filtros o monocromadores.	37
3.Celdas de absorción.	37
4.Detectores fotoeléctricos.	38
III.Cromatografía.	38
A.Principio del método.	39
B.El Cromatógrafo.	40
1.El gas portador.	40
2.Sistema de introducción de la muestra.	41

3.La columna.	42
4.Detectores.	42
<u>a.Conductividad térmica (catarómetro).</u>	42
<u>b.Densidad del gas.</u>	43
<u>c.Ionización por flama.</u>	43
<u>d.Ionización por rayos beta.</u>	43
<u>e.Ionización por luz.</u>	43
<u>f.Descarga luminiscente.</u>	43
<u>g.Temperatura de flama.</u>	43
<u>h.Constante dieléctrica.</u>	43
IV.Espectroscopía.	44
A.Principio del Método.	44
1.Infrarrojo.	46
2.Luz visible.	46
3.Ultravioleta.	46
4.Rayos X.	46
B.El Espectroscopio.	47
MÉTODOS PARA MUESTREO DE GASES.	50
I.Introducción.	50
II.Acondicionamiento de la muestra.	51
A.Filtrado.	51
B.Secado.	51
C.Reacciones químicas.	51
III.Precauciones en la recolección de la muestra.	52
A.Cambios químicos.	52
B.Adsorción.	52
C.Fugas.	52
IV.Métodos de recolección de la muestra.	53

A.Recolección de la muestra por adsorción.	53
1.Clasificación de los materiales adsorbentes.	54
a.Materiales adsorbentes polares.	54
b.Materiales adsorbentes no polares.	54
2.Comparación entre materiales adsorbentes polares y no polares. Caso particular.	55
3.Deadsorción.	56
a.Deadsorción en carbono activado.	56
b.Deadsorción en óxido de silicio gelatinoso.	57
B.Recolección de la muestra por absorción.	57
C.Recolección de la muestra por condensación.	58
D.Recolección de la muestra por desplazamiento.	59
1.Contenedores rígidos al vacío.	60
2.Contenedores rígidos a presión atmosférica.	61

TRANSDUCTORES PARA DETECCIÓN

DE GASES.	62
I.Introducción.	62
II.Unidades de medición para gases.	63
III.Transductores basados en la conductividad térmica.	64
A.El catarómetro.	65
B.Aplicaciones.	67
IV.Transductores basados en el calor de combustión.	67
A.Transductor catalítico.	68
B.Aplicaciones.	69
V.Transductores de luz infrarroja.	69

A. Detectores fotoeléctricos.	70
B. Aplicaciones.	70
VI. Transductores basados en la conducción electrolítica.	71
A. La celda electrolítica.	71
B. Aplicaciones.	74
VII. Transductores basados en la detección de iones específicos.	74
A. La celda voltaica.	75
B. Aplicación.	77
VIII. Detectores basados en la adsorción. Principio de funcionamiento.	77

SISTEMA DE ADQUISICIÓN DE DATOS.

I. Introducción.	80
II. Sistema digital de adquisición de datos.	81
A. Elementos de un sistema digital de adquisición de datos.	83
1. Transductor eléctrico y circuitería de soporte.	83
2. Circuito acondicionador de señal del transductor.	83
3. Filtro acondicionador de señal del transductor.	83
4. Canal analógico multiplexado y líneas de control.	85
5. Circuito acondicionador de señal del multiplexor.	85
6. Circuito para muestreo y retención.	85
7. Sistema de conversión analógica-digital (A/D).	85

8.Unidad de procesamiento.	86
<u>a.Programa de control.</u>	86
<u>b.Memoria.</u>	86
<u>c.Puerto de entrada/salida.</u>	86
9.Sistema de conversión digital-analógica (D/A).	86
10.Circuito acondicionador de señal del convertidor digital-analógico.	87
III.Características de un sistema digital de adquisición de datos.	87
A.Inmunidad a ruido.	87
B.Detección y corrección de errores.	88
C.Comunicación.	88
CONSIDERACIONES DE DISEÑO.	89
I.Introducción.	89
II.El proceso del diseño.	90
A.Principios del diseño.	90
1.Modularidad.	90
2.Regularidad.	90
3.Localidad.	91
<u>a.Localidad a nivel sistema.</u>	91
<u>b.Localidad a nivel componente.</u>	91
4.Conectividad.	91
B.Factores del diseño.	91
1.Precisión.	92
2.Rango.	92
3.Resolución.	92
5.Ergonomía.	92
6.Condiciones de falla.	93
7.Mantenimiento.	93
<u>a.Mantenimiento correctivo.</u>	93
<u>b.Mantenimiento preventivo.</u>	93

C.Técnica de diseño funcional descendente ("Top-Down").	93
D.Normas comerciales.	94
III. Selección de una familia de microprocesadores.	95
A. Microprocesadores contra lógica combinacional y secuencial.	95
1. Lógica combinacional y secuencial.	95
2. Microprocesadores.	96
B. Técnica de selección de una familia de microprocesadores.	96
C. Criterios adicionales para seleccionar una familia de microprocesadores.	97
IV. Limitaciones de diseño.	99
V. Planteamiento de especificaciones.	99
VI. Diseño del "hardware".	101
A. Partición del sistema en módulos.	101
B. Selección de la familia de microprocesadores.	102
C. Selección de dispositivos para el sensado de gases combustibles.	104
1. Transductor DET-TRONICS C7061C.	105
2. Convertidor analógico digital ADC0804.	107
D. Selección de dispositivos y estándares para comunicación.	108
1. Estándar de comunicación RS-232-C.	109
2. Pantalla digital de cristal líquido AND 671.	113
VII. Diseño del "software".	113
A. Partición del algoritmo en macros.	114
B. Selección del lenguaje de programación.	114

C.Partición funcional de cada macro.	115
D.Almacenamiento de información.	116
1.Almacenamiento de información de todo evento.	116
2.Almacenamiento de información de eventos seleccionados.	116
E.Esquema de la programación.	117

DISEÑO DEL SISTEMA PARA DETECCIÓN DE GASES COMBUSTIBLES.

	123
I.Introducción.	123
II.El "hardware" del sistema.	124
A.Transductor catalítico.	124
B.Convertidor analógico-digital.	125
C.Unidad de procesamiento.	127
1.Microprocesador MC68008.	128
2.Memoria ROM 2764.	129
3.Circuito de reloj.	130
4.Circuito para inicializar el sistema.	132
5.Circuito para decodificar el mapa de memoria.	134
6.Circuito para habilitar operaciones de lectura o escritura.	138
7.Circuito para generar la señal DTACK.	140
8.Circuito para generar las señales VPA y VMA.	142
9.Circuito para generación de interrupciones.	145
D.Unidad de almacenamiento.	145
E.Comunicación serial con Computadoras Personales.	148
1.Puerto serial ACIA 6850.	148

2.Generador de frecuencias MC14411.	148
3.Interfaz RS-232-C.	149
F.Comunicación con el usuario.	150
1.Puerto paralelo PIA 6821.	150
2.Teclado.	150
3.Señalización luminosa.	152
4.Pantalla y su Interfaz.	154
III.Software del sistema.	155

PRUEBAS AL SISTEMA PARA DETECCIÓN DE GASES COMBUSTIBLES.

I.Introducción.	174
II.Cuidados en la realización del "hardware".	175
III.Comunicación con el puerto paralelo PIA 6821.	175
IV.Comunicación con la memoria RAM 6264.	176
V.Comunicación con el puerto serial ACIA 6850.	178
VI.Comunicación con el convertidor analógico-digital ADC0804.	179
VII.Comunicación con la pantalla de cristal líquido AND 671.	181
VIII.Comunicación con el teclado.	182

CONCLUSIONES Y COMENTARIOS.	184
APÉNDICES.	
A.ECUACIÓN GENERAL DE LOS GASES. EJEMPLO DE APLICACIÓN.	188
B.REGISTROS DEL MICROPROCESADOR Y CADENAS DE CARACTERES.	190
C.ESTANDAR DE COMUNICACIÓN RS-232.	193
D.MANUAL DE OPERACIÓN.	196
E.PROGRAMACIÓN DE LA COMPUTADORA PERSONAL.	202
F.HOJAS DE ESPECIFICACIONES.	208
BIBLIOGRAFÍA.	259
ÍNDICE.	268

FIGURAS.

Figura 2.1: Esquema funcional de un <i>foto</i> metro.	36
Figura 2.2: Esquema funcional de un <i>cr</i> omatógrafo.	41
Figura 2.3: Esquema funcional de un <i>espectrómetro de emisión</i> .	48
Figura 2.4: Esquema funcional de un <i>espectrómetro de absorción</i> .	49
Figura 3.1: Esquema funcional de un dispositivo para recolectar muestras por <i>condensación</i> .	59
Figura 3.2: Esquema funcional de un dispositivo para recolectar muestras por <i>desplazamiento</i> .	60
Figura 4.1: Esquema funcional de la <i>unidad fundamental</i> de un catarómetro.	65
Figura 4.2: Esquema funcional de un <i>catarómetro</i> .	66
Figura 4.3: Esquema funcional de un <i>transductor catalítico</i> .	68
Figura 4.4: Esquema funcional de una <i>celda electrolítica</i> .	72
Figura 4.5: Esquema funcional de una <i>celda voltaica</i> .	75
Figura 4.6: Esquema funcional de un transductor asociado al fenómeno de <i>adsorción</i> .	78
Figura 5.1: Tipos de señales: (A) analógica; (B) discreta; (C) cuantizada; (D) digital.	82
Figura 5.2: Diagrama a bloques de un sistema digital de adquisición de datos.	84
Figura 6.1: Técnica de diseño funcional descendente (" <i>Top-Down</i> ").	94
Figura 6.2: Diagrama de bloques del Sistema para Detectar Gases Combustibles.	101
Figura 6.3: Diagrama de bloques del "hardware".	102
Figura 6.4: Diagrama de bloques de una arquitectura tipo SISD.	104
Figura 6.5: Transductor Det-Tronics C7061C.	105
Figura 6.6: Conexión RS-232-C simplificada entre terminales.	111
Figura 6.7: Circuito para una línea de transmisión con estándar RS-232-C.	112
Figura 7.1: Conexión del transductor catalítico Det-Tronics C7061C.	125
Figura 7.2: Circuito de conversión analógica-digital.	126
Figura 7.3: Terminales del microprocesador MC68008.	128
Figura 7.4: Circuito de reloj.	131
Figura 7.5: Circuito de " <i>Power-on RESET</i> ".	133
Figura 7.6: Diagrama de bloques y temporización de la lógica de decodificación de memoria.	134
Figura 7.7: Lógica de decodificación de memoria.	135
Figura 7.8: Circuito selector de lectura/escritura.	139

Figura 7.9: Diagrama de bloques y temporización del circuito de retardo para "DTACK".	141
Figura 7.10: Circuito de retardo para "DTACK".	142
Figura 7.11: Circuito generador de VMA y VPA.	144
Figura 7.12: Temporización del circuito generador de VMA y VPA.	143
Figura 7.13: Circuito para generación de interrupciones.	146
Figura 7.14: Circuitos de memoria RAM y ROM.	147
Figura 7.15: Circuito del puerto serial ACIA 6850.	149
Figura 7.16: Circuito del puerto paralelo PIA 6821.	151
Figura 7.17: Teclado	152
Figura 7.18: Teclado (continuación)	153
Figura 7.19: Pantalla de cristal líquido AND 671.	154