



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

“ENLACE INFRARROJO ENTRE UN COMPUTADOR Y  
UN MICROCONTROLADOR”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECANICO ELECTRICO

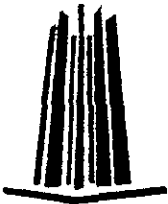
P R E S E N T A :

JESÚS LEONARDO LUJÁN CORTÉS

ASESOR

ING. RAUL ROBERTO BRIBIESCA CORREA

28737



México.

2000



Universidad Nacional  
Autónoma de México



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

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

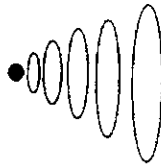
Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

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.

33

# **ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR**

**Jesús Leonardo Luján Cortés**



*“La ingeniería es el arte de realizar una aplicación práctica del conocimiento científico”*

*A Mis padres:*

*Les agradezco por su paciencia, por su constancia, por sus preocupaciones, por sus consejos, por sus desvelos, por el apoyo, y por su responsabilidad de educación hacia mi y mis hermanos.*

*El apoyo de mis padres fue fundamental para que terminara mis estudios, así como lo sigue siendo para que comience mi desarrollo profesional y como lo seguirá siendo para que no deje de obtener las metas que están plasmadas en mi vida. Este apoyo esta compartido por diversas personas, los cuales lo único que puedo hacer es agradecerles y mencionarles correspondiendo la atención, lo cual es poco por lo mucho que ellos me dieron, estas personas son:*

*Mis Hermanos ⇨ Paola Lizeth Luján Cortés  
Flavio Moisés Luján Cortés*

*Mi Abuelito ⇨ Moisés Cortes Mirón*

*Mi Tío ⇨ Oscar Guillermo Luján Mora*

*Un gran amigo ⇨ Alfredo Duarte*

*Un gran profesor ⇨ Eduardo Lira*

*Un gran grupo de trabajo ⇨ Fernando Araujo Castillo  
Manuel Lara Sánchez  
Miguel Angel Arriola  
Luis Andrés Méndez  
Mario Pérez Marin  
Florencio Sánchez  
Arturo García Guerrero  
Vicente García Jolalpa  
Fernando Castro Pérez  
Hugo Rogelio Sánchez*

# Indice

<b>INTRODUCCIÓN</b>	4
<b>OBJETIVOS, META, ALCANCE Y LIMITANTE.</b>	7
<b>Capítulo I</b>	
<b>1. LA RADIACIÓN INFRARROJA (IR) COMO MEDIO DE COMUNICACIÓN.</b>	8
1.1. ¿Qué es la radiación infrarroja (IR)?	9
1.2. ¿Cómo emplear este tipo de radiación para establecer un enlace de comunicación?	11
1.3. Ventajas y desventajas.	13
1.4. Futuras aplicaciones.	14
<b>Capítulo II</b>	
<b>2. COMUNICACIÓN ASÍNCRONA.</b>	16
2.1. El transmisor-receptor asíncrono universal (UART).	17
2.2. La interface estándar RS-232.	21
2.3. Códigos y protocolos.	26
2.4. Operación del canal de comunicación.	29
2.5. La interface de comunicación serie (SCI).	31
<b>Capítulo III</b>	
<b>3. REALIZACIÓN DEL ENLACE DE COMUNICACIÓN IR.</b>	44
3.1. Ejemplo de un sistema de comunicación inalámbrico IR.	44
3.2. Fundamentos.	47
3.3. Diseño de los circuitos de transmisión y recepción.	52
3.4. Software de comunicación.	72
3.5. Implementación y pruebas.	92

**Capítulo IV**

**4. RESULTADOS.** 113

**5. CONCLUSIONES.** 116

**6. BIBLIOGRAFÍA.** 117

**7. ANEXOS.**

**A. El microcontrolador MC68HC811E2** 119

**B. Registros de la Interface de Comunicación Serie (SCI)** 138

**C. SCORBOT-ER III** 147

# Introducción

El presente trabajo consiste en establecer un enlace de comunicación inalámbrico entre un computador y un microcontrolador, el cual tendrá la tarea de controlar los motores de un vehículo robotizado, permitiendo establecer mediante este sistema de comunicación un control a distancia sobre este vehículo empleando rayos de luz infrarrojos a una distancia de recepción moderada: seis metros y mínimo 30°. Cabe señalar que el diseño y construcción del vehículo, y el tipo de control que se emplea para los motores no corresponde a este trabajo, solo la comunicación computador-microcontrolador, siendo la principal aplicación, pero puede haber aplicaciones opcionales. Si el microcontrolador utiliza un control PID o difuso para el control de velocidad y posición de los motores no son temas que se aborden en la presente tesis.

En el caso del vehículo robotizado, es indispensable que el desplazamiento de éste sea totalmente independiente, por lo que resulta necesario llevar a cabo una comunicación que permita transmitir comandos o datos de control y que además se interpreten para realizar una tarea dada. Pero, cuando se desee comunicar a un computador, o a un brazo robot, o a cualquier otro dispositivo que necesite transmitir información o ser manipulado a distancia, las consideraciones para el enlace serían distintas debido a que sería un enlace móvil o un enlace fijo, pero las soluciones para la interconexión estarían dadas dentro de las especificaciones del propio sistema de comunicación, como por ejemplo; distancia de operación máxima (garantizada), ángulos de transmisión y recepción, velocidad de transmisión, etc.

El diseño del transmisor y el receptor infrarrojo es bastante económico y además se aplica el concepto bajo costo-confiabilidad. Es importante destacar las limitantes de la luz infrarroja, aunque esta energía puede ser rebotada por cualquier superficie lisa permitiéndonos aplicar la física de la reflexión de la luz para resolver los problemas de propagación aumentando así el área de operación, que en este caso para el dispositivo remoto sería más grande, lo cual para los fines de nuestra aplicación, el diseño es factible de realizarse.

Este sistema (que puede ser unidireccional, bidireccional o con repetidores) permitirá establecer un enlace de comunicación confiable (confiabilidad en la transmisión de los datos y susceptibilidad en la

recepción de comandos externos que sean transmitidos de alguna fuente extraña) entre el computador y el microcontrolador, donde el microcontrolador estará instalado dentro de un prototipo (vehículo robotizado) realizando tareas o procesos solicitados desde el computador. Además este sistema no sólo estará dedicado a un microcontrolador, como ya se había mencionado, tiene la flexibilidad de conectarse a un computador, para establecer un enlace computador-computador, o a un controlador de brazo robot, para establecer un enlace computador-brazo robot.

En el primer capítulo de este trabajo se presentan dos puntos de vista distintos de la radiación IR, dando una introducción de como esta radiación puede ser empleada para establecer un medio de comunicación, mencionando también sus ventajas, desventajas, y además el futuro muy prometedor en nuevas aplicaciones y en nuevos desarrollos orientados principalmente a la transmisión de datos.

Más adelante, en el segundo capítulo, se mencionan los elementos fundamentales de la comunicación asíncrona como el transmisor-receptor asíncrono universal (UART), la norma de nivel físico RS-232, el código ASCII, operación en simplex, half-dúplex, y full-dúplex, y la interface de comunicación serie para el microcontrolador 68HC811E2. Todo lo anterior proporciona al final del capítulo un panorama muy amplio de la transmisión asíncrona, así como también el principio de lo que se pretende con el presente trabajo de tesis.

Posteriormente, en el tercer capítulo, se presenta la parte de diseño de todo el sistema de comunicación inalámbrico IR, proporcionando un ejemplo del sistema orientado a como se verá operando en la práctica y los fundamentos básicos para poder explicar a detalle el diseño de los circuitos. También en este capítulo se menciona el aspecto software, el cual desempeña un papel muy importante en este proyecto. Al finalizar este capítulo se menciona la implementación de todo el sistema así como las pruebas, dando pie al desarrollo de las especificaciones que proporcionan los parámetros y las condiciones que este proyecto presenta en operación.

Finalmente, en el cuarto capítulo, se proporcionan todos los resultados que arrojo el sistema, así como una evaluación económica que permite obtener resultados sobre los costos que arrojo el desarrollo del presente proyecto de tesis.



Lo descrito anteriormente es mi trabajo de tesis, el cual es un proyecto de investigación y desarrollo con aplicación real en las instalaciones del Instituto Nacional de Investigaciones Nucleares (ININ).

Actualmente, en el departamento de Automatización e Instrumentación del ININ, sé esta realizando un sistema autónomo de salvaguarda, cuya función es el mantenimiento y ajuste de determinadas áreas del Reactor Nuclear TRIGA Mark III instalado dentro de esta institución. Otra de sus funciones será la ejecución de determinadas actividades que implican el manejo de material radiactivo, las cuales significan un peligro constante para los recursos humanos que las realizan hasta el momento.

Los investigadores a cargo de este proyecto, han dividido en varias secciones su realización, siguiendo una metodología bien definida donde se contempla la incorporación de estudiantes interesados en llevar a cabo el desarrollo de proyectos para efecto de titulación, ya que cada sección cuenta con los requisitos necesarios para ser elaborados en calidad de tesis de diferentes grados. Como en este caso la presente tesis se titula "Enlace infrarrojo entre un computador y un microcontrolador" se pretende desarrollar el enlace de comunicación inalámbrico del sistema autónomo de salvaguarda, el cual permitirá aplicar los conocimientos adquiridos en la Universidad Nacional y además es una buena oportunidad para estar en contacto directo con investigadores y con el desarrollo de ciencia y tecnología, mismas que se pretenden impulsar cada vez más en nuestro país.

## **OBJETIVO GENERAL DE LA TESIS**

Realizar un enlace de comunicación infrarrojo que permita transferir datos de control, de manera confiable para la realización de una aplicación real.

## **OBJETIVO DEL PROYECTO**

Lograr la comunicación inalámbrica entre un computador y un microcontrolador empleando señales infrarrojas.

## **OBJETIVO DE LA INVESTIGACIÓN**

Comprender la radiación infrarroja para diseñar e implementar el sistema de comunicación inalámbrico.

## **META**

Hacer tangible la comunicación infrarroja desde el computador hasta el microcontrolador de manera confiable, considerando las ventajas y desventajas de este tipo de radiación.

## **ALCANCE**

Terminar el proyecto hasta su montaje así como realizar diversas pruebas operativas, como por ejemplo; entablar una comunicación de un computador a otro.

## **LIMITANTE**

Emplear los componentes o dispositivos electrónicos disponibles en el mercado.

# Capítulo I

## 1. LA RADIACIÓN INFRARROJA (IR) COMO MEDIO DE COMUNICACIÓN

Todos los objetos emiten, absorben, y reflejan radiación infrarroja (IR). Lo anterior tiene dos implicantes. La primera, se refiere al estudio de la intensidad y distribución de la longitud de onda interactuando con objetos, información concerniente sobre el objeto es posible que se obtenga. Esta información se puede emplear, por ejemplo, para distinguir un cuerpo o para identificar un material desconocido. La segunda, la radiación es en sí misma importante, puede ser usada para cambiar las características de un material o para establecer un medio que lleve, transporte o transmita información de un punto a otro.

Aunque la existencia de la radiación IR se conoce desde hace más de un siglo<sup>1</sup>, sólo de las dos décadas anteriores a la fecha ha tenido un gran campo de acción en el desarrollo y producción de nuevas fuentes y detectores, dando como resultados más comunes y cotidianos, desarrollos para televisores y para controles VCR remotos. Una gran variedad de aplicaciones ingenieriles son ahora practicadas. En vista de esta situación, los especialistas emprendieron la tarea de diseñar diferentes tipos de instrumentos infrarrojos, así como desarrollar tecnología para la investigación en conductividad, física térmica, iluminación y óptica.

El uso de la radiación IR para propósitos militares y comerciales es bastante amplia, siendo que después de la segunda guerra mundial se reafirmo como principal importancia militar. Aunque las aplicaciones comerciales en los próximos años pasarán a primer plano, debido a que las grandes empresas están desarrollando nuevas aplicaciones. También hay que mencionar que existen diversas organizaciones relacionadas con la tecnología IR, las cuales cuentan con información detallada de productos y dispositivos que emplean esta tecnología para la transmisión de datos así como estándares que proporcionan plataformas de comunicación y soluciones.

<sup>1</sup> Sir John Herschel, es el descubridor de la radiación IR, demostró la existencia de absorción infrarroja de un material así como de igual manera esta radiación es emitida por el mismo objeto.

El campo de la tecnología IR esta dividido en cuatro categorías. La primera contempla la naturaleza de la radiación IR, por ejemplo; el espectro y distribución angular de radiación, la reflexión, refracción, difracción, absorción, disipación y los diversos efectos fotoeléctricos. La segunda categoría esta compuesta por componentes y materiales infrarrojos incluyendo fuentes de radiación y detectores.

La integración de componentes infrarrojos a sistemas constituye la tercera categoría. Esta contempla óptica (estudios sobre radiación IR y su interacción son aspectos fundamentales e importantes para la óptica) y electrónica. Aplicaciones tanto comerciales como militares, constituyen la cuarta y última categoría.

## 1.1. ¿Qué es la radiación infrarroja (IR)?

La radiación IR es un tipo de radiación electromagnética en el espectro infrarrojo, con longitudes de onda de aproximadamente 0.75 a 1,000 micras, o bien esta situada entre la región visible y la región de microondas, ver figura 1.1. Esta radiación se le denomina también luz negra, debido a que la luz es una radiación electromagnética cuyas longitudes de onda pueden producir la sensación de visión, extendiéndose desde 4,000 Angstroms (extremo violeta) a 7,700 Angstroms (extremo rojo), por este motivo, lo anterior se aplica actualmente a la radiación infrarroja invisible.

Desde un punto de vista molecular, la radiación IR puede ser absorbida y convertida en energía vibracional por medio de una molécula orgánica; es decir, que el espectro IR de una molécula es el resultado de la transición entre dos niveles diferentes de energía vibracional.

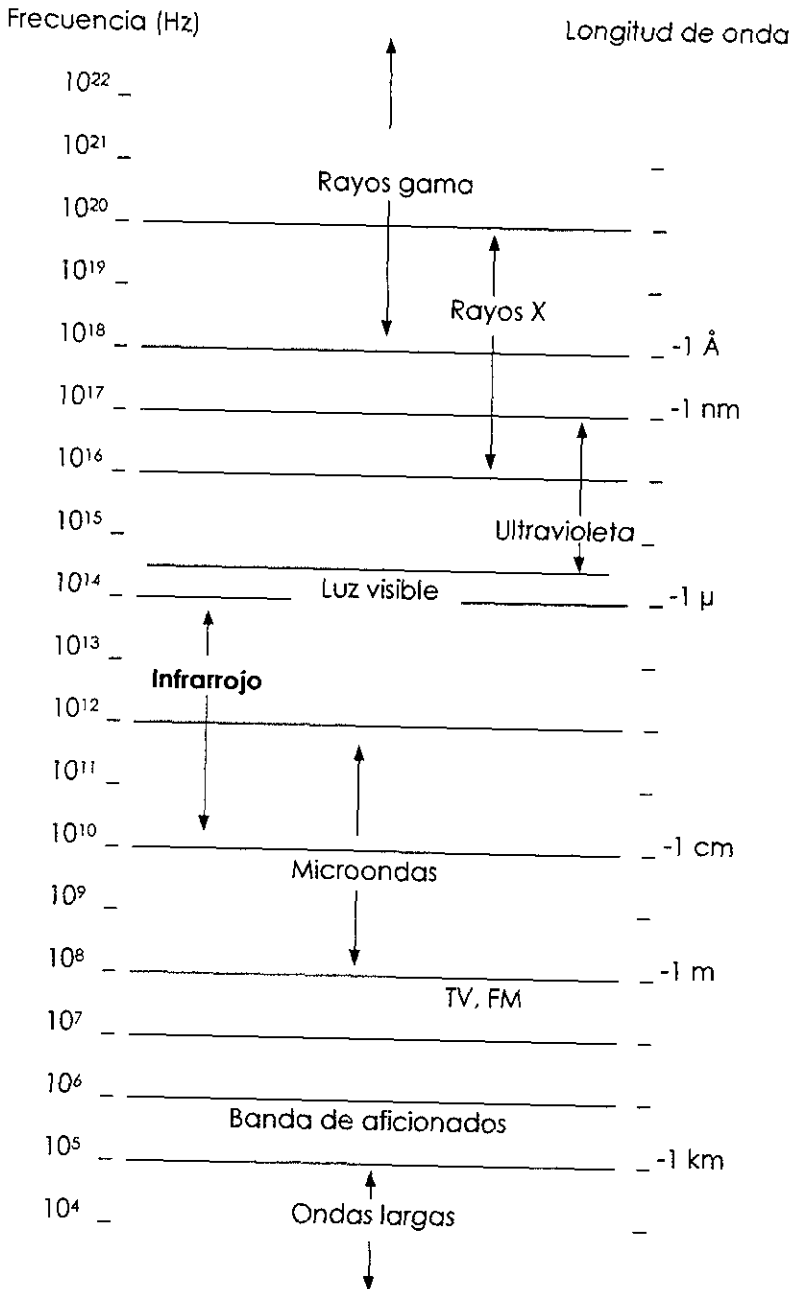


Fig. 1.1. Espectro electromagnético.

El espectro electromagnético es el ordenamiento de la radiación de acuerdo a la longitud de onda y a la frecuencia. Lo anterior tiene establecido experimentalmente ondas de todas longitudes desde la fracción más pequeña de un milímetro hasta varios kilómetros. Ello es un mecanismo de detección que permite utilizar completamente el espectro electromagnético, el cual tiene separaciones bastante definidas llamadas regiones. Las bases para estas subdivisiones o separaciones tienen generalmente un acuerdo con varios medios de generación, aislamiento y detección de radiación.

Los límites de las regiones del espectro electromagnético pueden expresarse por su longitud de onda, por su frecuencia, por su energía fotónica, o por el número de onda. Lo anterior se presenta en la siguiente tabla.

Tabla 1.1. Límites de la región IR

	Límite inferior	Límite superior
Longitud de onda	$7.5 \times 10^{-4}$ mm	1 mm
	0.75 $\mu$ (micras)	$10^3 \mu$
	750 m $\mu$ (milimicras)	$10^6$ m $\mu$
	7,500 Å (Angstroms)	$10^7$ Å
Frecuencia	$3 \times 10^{11}$ Hz ó 300 GHz (Hertz)	$4 \times 10^{14}$ Hz ó 400 THz
Energía fotónica	$1.23 \times 10^{-3}$ ev	1.72 ev
Número de onda	$10 \text{ cm}^{-1}$	$1.3 \times 10^4 \text{ cm}^{-1}$

## 1.2. ¿Cómo emplear este tipo de radiación para establecer un enlace de comunicación?

Una fuente de luz que emite radiación IR y un detector que recibe este tipo de radiación o luz, proporciona un enlace de comunicación que puede transmitir datos e información de un punto a otro, esto es, estamos empleando el haz de radiación IR para transportar la información que nosotros deseemos, utilizando el espacio libre como medio de propagación. Ver la figura 1.2.

El haz de radiación invisible, por ejemplo; es la señal portadora y la información o los datos es la señal moduladora que queremos hacer llegar desde el punto donde nos encontremos hasta el punto remoto que queremos que llegue, es decir, se están cambiando las características del

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

haz de radiación IR conforme a las características de la información que se quiere transmitir.

En lugar de emplear señales eléctricas para representar la información, estamos utilizando fuentes de luz, como diodos emisores infrarrojos para representar los datos binarios.

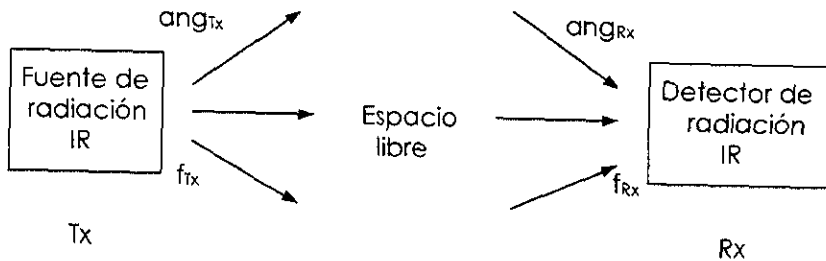


Fig. 1.2. Enlace de comunicación empleando radiación IR.

Para poder decir lo anterior tuvieron que pasar muchos años de investigación y desarrollo en óptica y electrónica, hasta comprender el hecho de que la luz es una onda electromagnética.

Con los experimentos ópticos de Newton funda la espectroscopia y con ello da un impulso importante a estudios sobre la naturaleza de la luz. El estudio de la radiación IR es bastante lento debido a muchas razones, como por ejemplo; a la carencia de sensibilidad y precisión en los detectores, así como el tiempo de desarrollo de la instrumentación y las técnicas para la medición de cantidades eléctricas. Pero durante 1830 existen detectores disponibles con numerosas características físicas.

Las ideas de aquellos años giraban entorno a que la radiación IR es del todo similar a la luz, pero en principio no fue aceptada, se aceptaba como dos conceptos diferentes: la radiación en su totalidad, es una lluvia de partículas o es una onda, pero no pueden ser ambos.

Probablemente el experimento más notable relacionado con el estudio de la radiación IR es la teoría clásica de la radiación electromagnética de Maxwell en 1862. Partiendo de ese experimento Henry Hertz, en 1887, pudo producir por medios eléctricos radiación IR con longitudes de onda muy largas. Este aparente aumento en la longitud de onda no tiene diferencia alguna entre la producción de ondas electromagnéticas, térmicas o eléctricas.

Con otros experimentos Hertz, dio pie a extensas investigaciones del efecto fotoeléctrico, debido a que radiación de suficiente longitud de onda corta es posible que ionice átomos.

Unos años después, en 1900, Planck tuvo el concepto de cuantización de energía, afirmando que la emisión de radiación no toma parte continua pero sí pasos discretos. La explicación afortunada de Einstein sobre el efecto fotoeléctrico, basado en la hipótesis de Planck, también contribuye al establecimiento de la validación de la teoría cuántica. Estos hombres contribuyeron a que aquella idea sobre la radiación de que es luz u onda fuera desechada y se adoptara el concepto de que la radiación IR es un tipo de luz invisible (el ojo humano no es capaz de percibirla) formada por fotones y además es una onda que puede reflejarse, refractarse, difractarse, etc. Partiendo de esta nueva idea se comenzaron a desarrollar nuevos estudios y desarrollos en distintos campos.

Después de las dos grandes guerras se dieron avances muy importantes en la electrónica (aparece el transistor y con ello la juntura p-n) y en la óptica, donde posteriormente estas dos disciplinas se combinaron para formar la optoelectrónica, dando pie al desarrollo de nuevas aplicaciones de la radiación IR, obviamente para usos militares, como en detectores con mayor sensibilidad y rapidez, los cuales son fundamentalmente sensores que dependen de un fotón o de las características de una partícula de radiación térmica. Otros desarrollos hostiles fueron implementados como por ejemplo; equipos compactos y ligeros para el guiado de misiles, sistemas para rastrear y dirigir ataques hacia tropas enemigas, así como para custodiar los movimientos enemigos en completa obscuridad, imágenes térmicas de áreas descubiertas tal como calles, edificios, y sitios de lanzamiento de misiles, y finalmente sistemas de transmisión de datos.

### 1.3. Ventajas y desventajas

La radiación IR se comporta exactamente como la luz visible, los rayos IR proporcionan muchas posibilidades, pero sufren las mismas limitantes que la luz visible al momento de su propagación. Un haz infrarrojo atravesará evidentemente con una gran atenuación la superficie transparente de un objeto, pero para los obstáculos opacos se incrementa el problema.



# Capítulo II

## 2. COMUNICACIÓN ASÍNCRONA

Una forma de evitar la sincronización entre dos dispositivos que tengan que entablar una comunicación entre sí, es mediante una conexión serie asíncrona que permita enviar información de un lugar a otro por medio de tramas conformadas por un bit de inicio, 8 bits de datos y un bit de paro, los cuales le dirán al receptor en donde inicia y en donde termina la información para que así pueda ser entendida por el dispositivo, lo anterior se denomina: Comunicación Asíncrona.

La conexión serie permite llevar 8 bits de información uno detrás de otro por un único hilo, evitando así 8 hilos que realizarían la misma operación. Para llevar a cabo lo anterior se necesita de un registro de desplazamiento como indica la figura 2.1.

Los 8 bits de información, disponibles en paralelo en el computador, se aplican a un registro de desplazamiento que recibe una señal cuadrada de frecuencia perfectamente estable y definida. Debido a la propiedad de este circuito, vamos a encontrar a su salida una señal que tiene el aspecto que se indica en esta misma figura (se utiliza la palabra de 8 bits 10011101), señal que no es otra que nuestros bits de información colocados unos detrás de los otros al ritmo de la señal cuadrada, que recibe el nombre de reloj de transmisión.

En el otro extremo, un registro de desplazamiento idéntico al anterior recibe las informaciones en serie, así como un reloj de transmisión cuya frecuencia debe ser, evidentemente, la misma que la utilizada en el lado del computador. Estos dos relojes deben estar sincronizados con el fin de que el registro de desplazamiento receptor pueda tener en cuenta los datos que recibe en el momento adecuado. Si se juntan todas estas condiciones, nuestra palabra de 8 bits estará disponible a la salida del registro de desplazamiento receptor sin ninguna dificultad. Respecto a una conexión en paralelo, se ha ganado en el número de hilos, ya que aquí sólo 2 son útiles, el hilo de la señal que acabamos de ver y un hilo de tierra. Si la conexión tiene que ser bidireccional, basta entonces con 3 hilos, 2 hilos de datos (uno para cada sentido) y un hilo de tierra; la ganancia es apreciable, sobre todo para grandes longitudes.

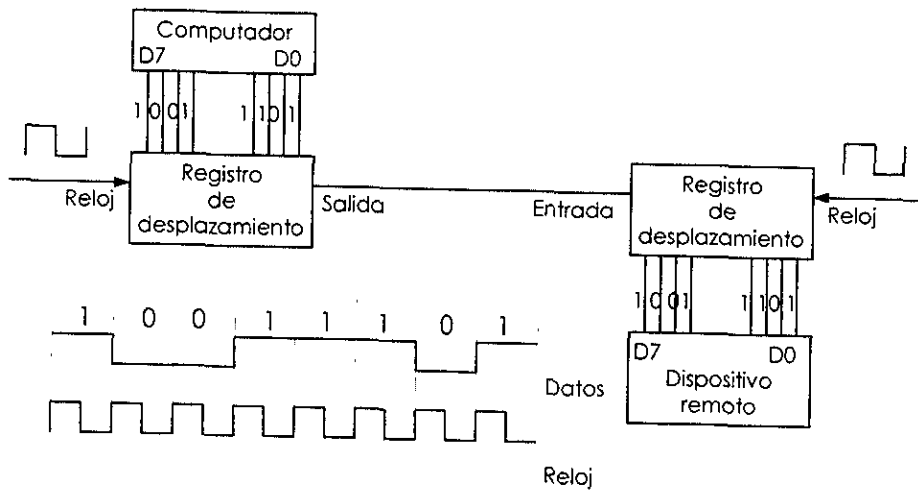


Fig. 2.1. Esquema de una conexión serie.

## 2.1. El transmisor-receptor asíncrono universal (UART)

Como se observa en la figura 2.1, se emplean solamente registros de desplazamiento, pero no siempre este tipo de circuitería de comunicación es tan simple, al contrario, se emplean dispositivos más complejos que se denominan UART, lo que significa Universal Asynchronous Receiver Transmitter, es decir, transmisor receptor asíncrono universal.

Estos circuitos son dispositivos de interfaz para microcontroladores que llevan internamente, además de un registro de desplazamiento, toda una circuitería apta para generar las señales especiales de una conexión serie asíncrona. Como su nombre lo indica, todos los UARTs son dobles, es decir, que contienen un transmisor y un receptor. Basta, por tanto, con un dispositivo para establecer una conexión serie bidireccional.

Toda la astucia de la conexión serie asíncrona descansa en la forma de las señales que se envían, señales que permiten una resincronización del receptor cada vez que se recibe un carácter. Examine la figura 2.2, que representa la transmisión asíncrona del byte 11001000. En estado de reposo, la línea de transmisión está en el estado lógico alto. La transmisión comienza con el paso a "0" de esta línea durante un periodo del reloj de transmisión, lo que constituye un bit de inicio; los bits de la palabra a

transmitir se envían a continuación detrás de este bit de inicio y, después del último bit "útil", la línea pasa de nuevo al estado alto durante uno o dos períodos de reloj para constituir lo que se llama el bit de *paro*. A este formato se le conoce como *no retorno a cero* ("Non Return to Zero", por sus siglas en inglés NRZ). Cabe mencionar que otros sistemas emplean 2 ó 1.5 bits de paro.

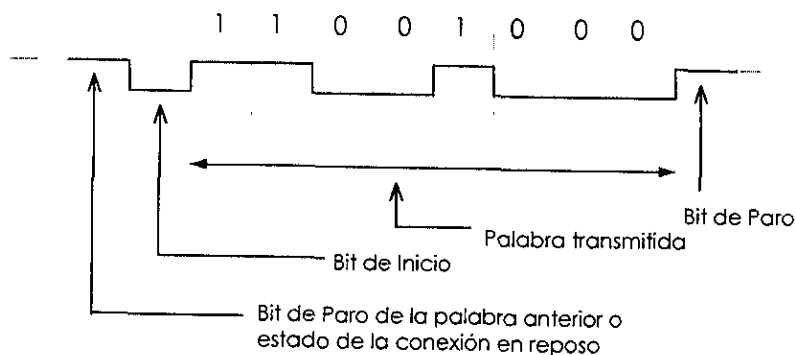


Fig. 2.2. Aspecto de un carácter en una transmisión serie asíncrona.

En otros términos, cada carácter transmitido se encuadra entre dos bits especiales que precisan el principio y el fin, estos bits son automáticamente colocados y recogidos por los circuitos UART. Por tanto, su presencia es "transparente" respecto al usuario, el cual se conforma con escribir la palabra a transmitir en un registro del UART para hacer una transmisión, igual que se limita a leer la palabra recibida en otro registro de este mismo UART durante una recepción.

Antes de seguir con el UART, es preciso mencionar un poco más sobre el formato NRZ, el cual es la vía más común para transmitir señales digitales empleando dos niveles de voltaje diferentes para dos dígitos binarios diferentes. La estrategia de este código es muy simple, se trata de mantener el nivel de voltaje constante durante un bit; si no hay transición, no retorna hacia el nivel de voltaje cero. Por ejemplo, la ausencia de voltaje puede ser usado para representar el bit "0", y un voltaje positivo constante puede ser usado para representar el bit "1". Este código, se emplea generalmente para interpretar datos digitales por terminales y otros dispositivos, para grabación magnética digital, y siendo además un formato de datos estándar muy utilizado por microcontroladores y microprocesadores.

El código NRZ es muy eficiente en el empleo del ancho de banda, donde mucha de la energía de la señal NRZ se divide entre la corriente directa (cd) y la mitad de la velocidad de transmisión. Por ejemplo, si el código NRZ es usado para generar una señal con una velocidad de 9600 bps, mucha de la energía en la señal es concentrada entre la corriente directa y 4800 Hertz (Hz).

La principal limitante de la señal NRZ es en la presencia de una componente de cd, pero en general, este código resulta muy atractivo para aplicaciones de transmisión de datos.

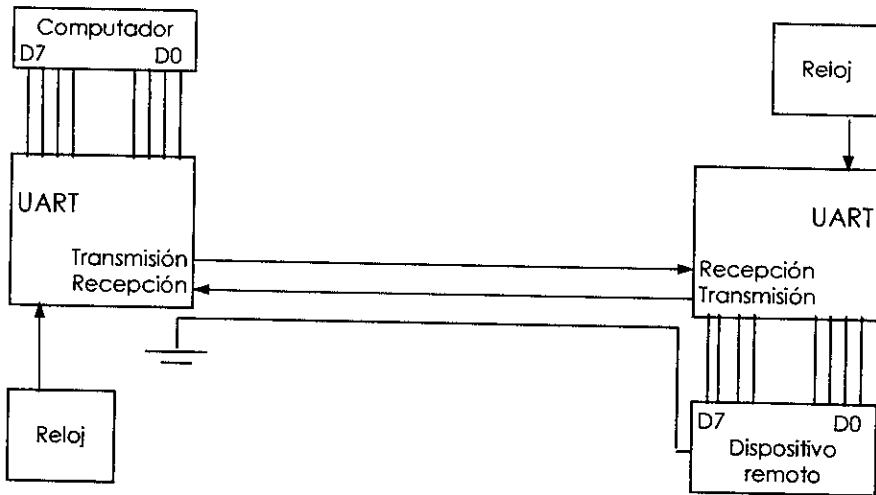


Fig. 2.3. Conexión serie asíncrona bidireccional empleando sólo 3 hilos.

Como muestra la figura 2.3 los relojes de transmisión y recepción se aplican directamente a los UART (por lo regular están integrados en estos dispositivos); ésta es la razón por la que tal conexión puede funcionar. En efecto, gracias al bit de inicio, el UART receptor se sincroniza con la señal recibida y analiza cada bit recibido en la mitad de su duración teórica, como muestra la figura 2.4. Basta entonces con que las frecuencias de los relojes transmisor y receptor sean también parecidas para no producir un desplazamiento demasiado importante en la duración de una palabra transmitida, para que todo funcione bien. Además, debido a esta resincronización al comienzo de cada carácter, los relojes transmisor y receptor no tienen necesidad de ninguna ligadura entre ellos, y son totalmente asíncronos.

por segundo que es, por tanto, igual a la velocidad en baudios dividida por 10. Así, 2400 baudios equivalen a 240 caracteres por segundo.

Hay que recordar que los caracteres transmitidos y recibidos son palabras de 8 bits, en las que no aparece ni el bit de inicio ni el de paro, es el UART el que se encarga de insertar y de extraer dichos bits.

La mayoría de los equipos de cómputo así como los microcontroladores y microprocesadores disponen de una interface serie asíncrona, lo cual habla de la posibilidad de interconexión entre diversos dispositivos, así como de las posibilidades que nos dan los circuitos UART.

Para un computador, esta interface sería el puerto de comunicaciones COM1 ó COM2 dependiendo de su configuración, donde físicamente es un conector DB-25 ó DB-9. Para un microcontrolador o microprocesador, esta interface es reconocida como línea TxD que es la salida serie asíncrona y línea RXD que es la entrada serie asíncrona, las cuales físicamente son los pines del microcircuito.

<sup>2</sup> El bit de paridad se utiliza como bit de control de la calidad de transmisión, y además ocupa el octavo bit, el cual está inutilizado por el código ASCII.

## 2.2. La interface estándar RS-232

Quando un sistema de comunicación, obviamente, de datos utiliza una conexión serie asíncrona, es muy usual que emplee el estándar RS-232. Este estándar o norma es también conocida como CCITT V.24 ó simplemente V.24, es de origen norteamericano, define los niveles eléctricos de las señales utilizadas para la transmisión, los tipos de conectores, y las asignaciones de los pines.

Los niveles de señal empleados por el estándar RS-232 son diferentes a los niveles de los circuitos lógicos digitales. Los valores nominales de las señales de datos son entre +3 y +12 Volts (espacio) para un cero lógico y entre -3 y -12 Volts (marca) para un uno lógico. Este estándar, también especifica los voltajes máximos y mínimos para ambos niveles lógicos tanto para transmisión como para recepción, así como una señal no lógica que puede ser entre +3 ó -3 Volts, lo cual es considerado como cero Volts.

Este estándar define 25 señales diferentes, aunque en la práctica no se utilizan todas. El conector que recomienda esta norma es el de 25 pines conocido como DB-25, o también el conector DB-9 de solamente 9 pines. Estos conectores están disponibles tipo macho y tipo hembra. En la figura

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

2.5 se muestran las denominaciones oficiales de las señales RS-232 para el conector DB-9, así como las asignaciones de los números de pines normalizados. Cabe señalar que se muestra solamente el conector DB-9, debido a que este se utiliza en el proyecto, y además son muy usados como puertos serie en diversos computadores.

Nombre	Abreviación	Número de pin
1	DCD	Data Carrier Detect (Detección de portadora)
2	RxD	Receive Data (Recepción de datos)
3	TxD	Transmit Data (Transmisión de datos)
4	DTR	Data Terminal Ready (Terminal preparado)
5	SG	Signal Ground (Señal de Tierra)
6	DSR	Data Set Ready (Transmisor preparado)
7	-	No asignado
8	CTS	Clear To Send (Invitación a transmitir)
9	-	No asignado

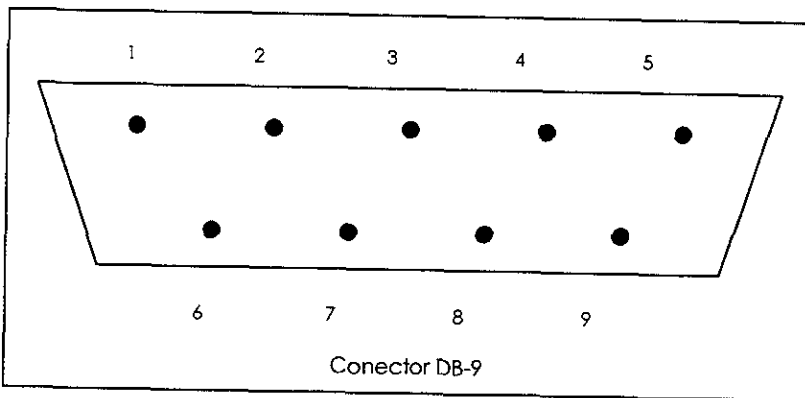


Fig. 2.5. Denominaciones de las señales RS-232 y asignación de números de pin.

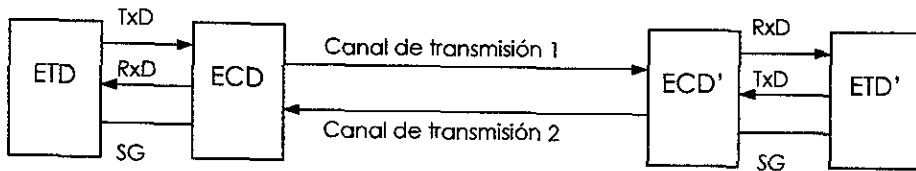
Por lo regular, solamente se requieren tres señales para llevar a cabo una comunicación. Por ejemplo, una tarjeta de evaluación EVBU para un microcontrolador (tarjeta emuladora) solamente emplea tres señales, aunque el puerto de la computadora (o terminal) pueda requerir algún otra conexión adicional. Las tres señales se muestran en la tabla 2.2.

Tabla 2.2. Señales RS-232 comúnmente empleadas (DB-9)

Nombre	Abreviación	Número de pin
Datos Transmítidos	TxD	3
Datos Recibidos	RxD	2
Señal de tierra	SG	5

La Organización Internacional de Estándares (ISO, International Standard Organization) y el Comité Consultivo Internacional de Telegrafía y Telefonía (CCITT, Committee Consultative International Telegraph and Telephone) tienen dos categorías de equipos de comunicación de datos: el Equipo Terminal de Datos (ETD), que es normalmente un equipo de usuario final, y el Equipo de Comunicación de Datos (ECD), donde su función es conectar el ETD a la línea de comunicaciones. El estándar RS-232 solamente emplea estas dos categorías.

Tomando como base el modo de conexión en dos vías que muestra la figura 2.6, es el ejemplo a seguir para la conexión del sistema de comunicación inalámbrico IR que se pretende desarrollar, ya que será una comunicación en dos vías, siendo esta la razón por lo cual se menciona con anterioridad la comunicación bidireccional, considerando además que la operación del sistema será en modo Half-dúplex, que más adelante se explicara a detalle. Las igualdades que presenta esta figura nos permiten respetar la norma RS-232 conforme a los dispositivos que comprende el sistema de comunicación IR, definiéndolos como equipos terminales de datos o como equipos de comunicación.



ETD = Computador  
 ECD = Transmisor-Receptor IR  
 ECD' = Receptor-Transmisor IR  
 ETD' = Microcontrolador o Computador o Brazo robot

Fig. 2.6. Modo de conexión bidireccional (o de dos vías) permitido por la norma RS-232.

El modo de conexión bidireccional, como ya se menciono, emplea tres señales (TxD, RxD, y SG) siendo una simplificación de las señales restantes (DCD, DTR, DSR, y CTS) que presenta esta norma. Lo anterior es suficiente para establecer una conexión serie asíncrona, pero puede dar lugar a problemas si el equipo al cual se quiere conectar utiliza ciertas señales de control. La solución es entonces un "cortocircuito" tal como el de la figura 2.7, conocido como *módem nulo*, el cual permite simular las señales ausentes mediante el cierre del circuito interno del conector. Es muy importante señalar que los conectores del sistema de comunicación IR están configurados en *módem nulo*.

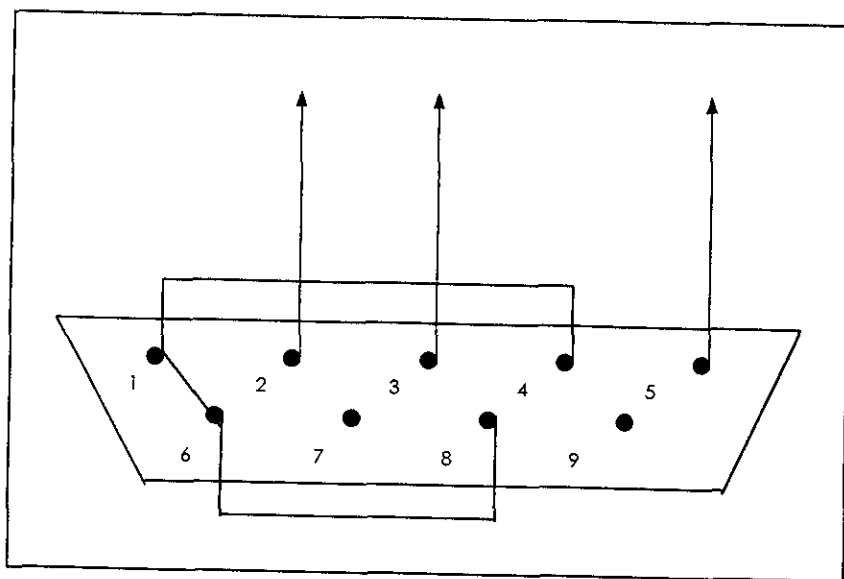


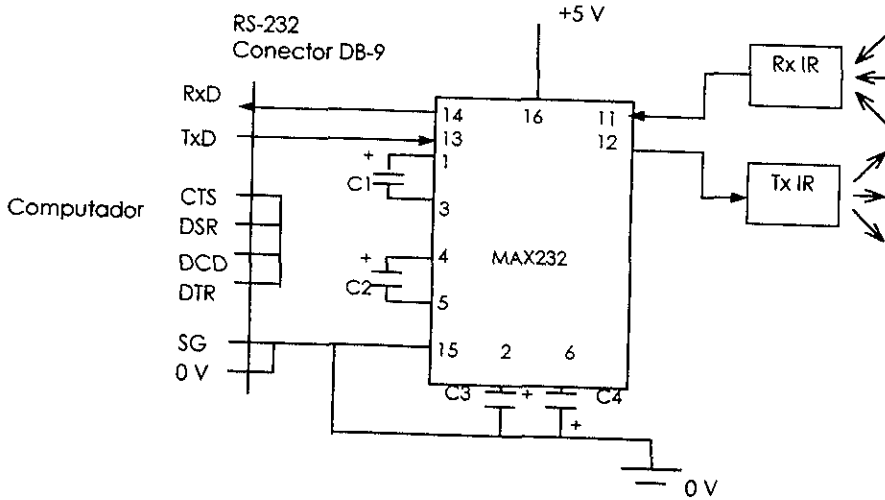
Fig. 2.7. Módem nulo para el conector DB-9.

Antes de terminar la explicación de la norma RS-232, es necesario mencionar que existen circuitos de conversión de nivel, que permiten pasar los niveles de  $\pm 12$  V (niveles RS-232) a 0 y 5 V (niveles TTL) o viceversa. Estos circuitos necesitaban una alimentación de  $\pm 12$  V, pero afortunadamente los fabricantes de semiconductores desarrollaron circuitos integrados que pueden proporcionar los niveles de las señales RS-232 empleando solamente 5 Volts de alimentación. Uno de estos circuitos es el MC145407, el cual requiere sólo 5 V de alimentación, pero existe otro circuito integrado que da mejores resultados y además es el que empleamos en este proyecto, el MAX232 o "Maxim", que con la ayuda de algunos capacitores, como se muestra en la figura 2.8, proporciona los



ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

niveles de voltaje requeridos, ya sea de TTL a RS-232 ó de RS-232 a TTL. En esta misma figura se muestra la conexión del puerto RS-232 DB-9 (COM1 del computador) configurado en módem nulo hacia el Receptor-Transmisor IR.



C1=C2=C3=C4= 1  $\mu$ F 25 V

Fig. 2.8. Conversión de niveles de señal empleando el MAX232.

La figura anterior muestra la aplicación principal del Maxim en este proyecto, proporcionando niveles entendibles entre el computador y el Receptor-Transmisor IR. La norma RS-232 en base a la figura 2.8, define que el computador es considerado como un dispositivo ETD y que el sistema de comunicación IR es conectado como un dispositivo ECD.

Finalmente, este estándar es ampliamente utilizado en muchas partes del mundo, y no es raro que equipos de cómputo para oficinas o domésticos sean compatibles con esta norma.

## 2.3. Códigos y protocolos

La mayoría de los sistemas de comunicación transmiten datos en forma binaria; es decir, 1's y 0's. Nosotros estamos familiarizados con el sistema de numeración decimal. Sin embargo, los computadores y las terminales están diseñados para representar sólo dos estados de señal (ya sea, onda de radio u óptica): un 0 ó un 1.

Estos datos binarios además de representar información numérica, es decir, números, también permiten representar otros símbolos, como las letras del alfabeto o los caracteres especiales (como el símbolo "#"). El sistema binario representa esa información mediante el uso de códigos.

Podrían existir multitud de códigos diferentes pero, afortunadamente, uno se ha normalizado y se emplea en la mayoría de los aparatos del mercado, grandes o pequeños; éste es el código ASCII, que significa American Standard Code for Information, o bien código americano estándar para intercambio de informaciones, ver la figura 2.9. El código ASCII fue desarrollado y patrocinado por el Instituto Nacional Americano de Normalización (ANSI), es además un estándar a nivel internacional, de acuerdo con el Alfabeto Internacional 5 (IA5).

El código ASCII es un código de 7 bits, por tanto, se adapta bien a los microordenadores de palabras de 8 bits, pero es una pena que se desperdicie un bit aunque muchos fabricantes lo utilizan en las conexiones serie asíncronas como bit de paridad, este punto es mencionado anteriormente. Este código puede representar 128 caracteres, lo que es más que suficiente para las aplicaciones actuales.

Podemos ver un ejemplo de control de dispositivos en la función del carácter de retroceso (BS - backspace) de los juegos de caracteres de ASCII. Por ejemplo, BS podría hacer retroceder un espacio a un determinado terminal receptor, haciendo retroceder el cursor un espacio en la pantalla.

A continuación, se describen otros caracteres ampliamente utilizados para el control de dispositivos: el tabulador horizontal (HT) hace que el dispositivo de impresión se desplace hasta la siguiente posición predeterminada antes de imprimir el siguiente carácter. El carácter de salto de línea (LF) hace que el dispositivo de impresión se desplace hacia la siguiente línea, permaneciendo en la misma columna. El carácter de tabulación terminal (VT) hace que el dispositivo de impresión avance un número predeterminado de líneas permaneciendo en la misma columna,

a partir de la línea que está siendo impresa o presentada en pantalla. El código de salto de página (FF) se puede utilizar para el dispositivo de impresión hasta el siguiente comienzo lógico de página, o hasta una línea predeterminada de la siguiente hoja o página. El código de retorno de carro (CR) retrocede hasta la primera columna de la página en curso. A menos que el retorno de carro sea seguido de un salto de línea, los caracteres que sigan al retorno de carro se imprimirán encima de los que ya habían sido impresos en esa misma línea.

La segunda función importante que hemos comentado anteriormente es la de representación y transporte de datos. Por ejemplo, la letra C se representa en ASCII como 1100001, (ó 43 en Hexadecimal) donde los niveles de tensión, las intensidades de corriente, las variaciones de las señales de radio o los cambios de estado de la luz infrarroja son las que se utilizan para representar los estados binarios de los códigos.

Posiciones de bits	7, 6, 5	000	001	010	011	100	101	110	111
	Hex 0	0	1	2	3	4	5	6	7
4 3 2 1	Hex 1								
0 0 0 0	0	NUL	DLE	SP	0	@	P	\	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	A	q
0 0 1 0	2	STX	DC2	"	2	B	R	B	r
0 0 1 1	3	ETX	DC3	#	3	C	S	C	s
0 1 0 0	4	EOT	DC4	\$	4	D	T	D	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	E	u
0 1 1 0	6	ACK	SYN	&	6	F	V	F	v
0 1 1 1	7	BEL	ETB	'	7	G	W	G	w
1 0 0 0	8	BS	CAN	(	8	H	X	H	x
1 0 0 1	9	HT	EM	)	9	I	Y	I	y
1 0 1 0	A	LF	SUB	*	:	J	Z	J	z
1 0 1 1	B	VT	ESC	+	;	K	[	K	{
1 1 0 0	C	FF	FS	,	<	L	\		:
1 1 0 1	D	CR	GS	-	=	M	]	M	}
1 1 1 0	E	SO	RS	.	>	N	^	N	~
1 1 1 1	F	SI	US	/	?	O	_	O	DEL

Fig. 2.9. El código ASCII (IA5) de ANSI y CCITT.

El código ASCII/IA5 se desarrolló varios años antes de la aparición de los terminales inteligentes. Los terminales primitivos, sin pantalla de tubo de rayos catódicos, eran fundamentalmente dispositivos tipo impresora. En consecuencia, las funciones de control eran muy limitadas, y básicamente consistían en el retroceso, la tabulación vertical, el salto de línea y el retorno de carro. Con la evolución de la industria hacia terminales mucho

más complejas, los códigos de control se revelaron insuficientes para proporcionar todas las órdenes de control de los dispositivos. Para afrontar este problema, ANSI ha desarrollado estándares que extienden el código básico. Por ejemplo, ANSI X3.41 y X3.64 utilizan el carácter de escape (ESC) para "construir" caracteres de control que permitan realizar funciones adicionales. En concreto, X3.41 define un conjunto de órdenes, como extensión del código ASCII, y X3.64 define las acciones específicas que dichos códigos originan en los dispositivos. Por ejemplo, la secuencia de códigos ESC 6 C ordena a la terminal que adelante el cursor de pantalla seis posiciones.

La tercera misión de los códigos es controlar la "conversación" entre computadores y terminales. Esta conversación se denomina *protocolo*. El término protocolo tiene muchos y muy variados usos en la industria. Prácticamente cualquier persona que trabaje en comunicación de datos hablará de protocolos, pero no siempre con el mismo sentido. En pocas palabras, un protocolo no es más que un acuerdo que establece cómo conversar. Existen muchas formas y niveles de protocolos, lo que conduce a veces a alguna confusión en la industria. Por ejemplo, el acuerdo o protocolo entre dos dispositivos que se comunican puede ser a nivel eléctrico. Los dos dispositivos pueden reconocer un uno como -12 Volts en el canal de comunicaciones, y un cero como +12 Volts. Este tipo de acuerdo sería parte del protocolo a nivel eléctrico.

Otra forma de protocolo está orientada al software o microcódigo, y se aplica con los códigos comentados anteriormente. Por ejemplo, el carácter de control de protocolo EOT (End Of Transmission) significa el final de la transmisión de los datos de usuario. Cuando un dispositivo transmite datos a otro dispositivo, envía un bloque de datos de usuario y añade después la cadena de bits que representa el carácter EOT para indicar al receptor que ya ha transmitido todos los datos. Obviamente, ambos dispositivos deben reconocer el carácter EOT en el mismo formato y contexto, o no se podrán comunicar.

Algunos fabricantes establecen que EOT significa el fin de la transmisión y la desconexión de la estación del sistema. Otros lo interpretan como fin de la transmisión, pero sin implicar la desconexión del sistema, ya que la estación puede intentar enviar datos más tarde. Estos dos protocolos serían, por tanto, incompatibles, debido a la diferente interpretación de los códigos de control de los protocolos.

Cabe señalar que existen protocolos de línea o de enlace, cuyas funciones principales son proporcionar una transmisión de datos por la

línea de comunicación libre de errores debidos ya sea por la distorsión proporcionada por el ruido o a otros problemas.

Un punto muy importante en la comunicación de datos después de los códigos y los protocolos, es el control de flujo, que se refiere al método de parar y reinicializar la transferencia de datos. Cuando un dato es transferido, tiene que seguir ciertas reglas tanto del transmisor como del receptor. En algunos casos un programa puede iniciar una transmisión desde otro dispositivo.

Algunos métodos de control de flujo más conocidos son; el **ACK/NAK** (Acknowledge/Not Acknowledge, es decir, Reconocimiento/No Reconocimiento), el **DTR/DSR** (Data Terminal Ready/Data Set Ready, es decir, Terminal preparado/Transmisor preparado), el **RTS/CTS** (Request To Send/Clear To Send, es decir, Demanda de transmisión/Invitación a transmitir), y por último el **XON/XOFF** (es decir, Encendido/Apagado). De los cuatro métodos anteriores, sin duda, el más común es el último, el cual permite que el receptor envíe una orden (por alguna razón) al transmisor para que pare la transferencia de datos. Por ejemplo, un computador envía un dato hacia un microcontrolador. Si el buffer del microcontrolador esta lleno, entonces este le ordena al computador que pare la transmisión; de lo contrario, el dato transmitido se perderá. El receptor envía una señal XOFF hacia el transmisor ordenándole que pare la transmisión. Posteriormente, cuando el receptor esta listo, envía una señal XON hacia el transmisor, ordenándole que reanude la transmisión.

Por último, las señales XON y XOFF son en ASCII los caracteres de control DC1 (Ctrl-Q) y DC3 (Ctrl-S), respectivamente.

## 2.4. Operación del canal de comunicación

El canal de comunicación es el camino que conecta a dos dispositivos. Existen tres caminos básicos para transmitir datos a lo largo de un canal de comunicaciones:

- Transmisión Simplex
- Transmisión Half-dúplex
- Transmisión Full-dúplex

Un aspecto final de la transferencia de datos que merece especial atención son las capacidades direccionales del sistema. La *transmisión simplex* (ver figura 2.10a), en la cual la información se envía de un solo lugar predeterminado, es decir en una sola dirección. Como la mayor

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

parte de las aplicaciones implica una comunicación en ambos sentidos, la *transmisión dúplex* es predominante.

En la *transmisión half-dúplex* (ver figura 2.10b), los mensajes se envían en cualquier dirección pero sólo en una dirección en un momento específico. Para la *transmisión full-dúplex* (ver figura 2.10c), la información puede enviarse simultáneamente en ambas direcciones.

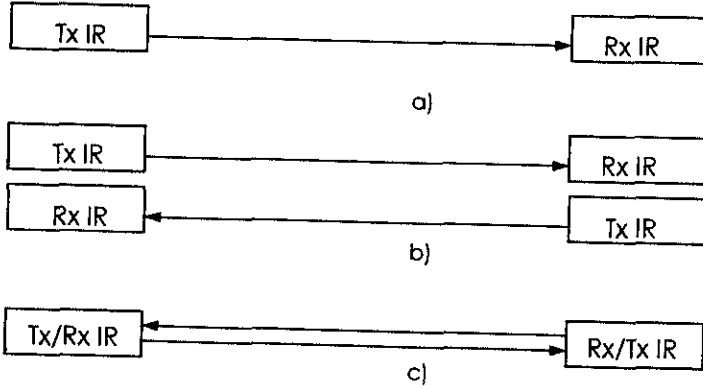


Fig. 2.10. Descripción gráfica de a) Transmisión simplex, b) Transmisión half-dúplex, y c) Transmisión full-dúplex.

Los diferentes tipos de transmisión que acabamos de mencionar tienen relación con lo que se ve en la pantalla. Cuando se escribe un mensaje por medio de una transmisión full-dúplex, las palabras que aparecen en la pantalla no vienen directamente del teclado; en su lugar, éstas abandonan la computadora, viajan a la computadora con la cual se efectúa la comunicación, después regresan al sistema como "eco" y se exhiben en la pantalla. En consecuencia, si se quiere ver lo que se está escribiendo, ya sea el Transmisor-Receptor IR o el programa debe asegurar que se muestre una copia local o *eco local* en la pantalla (ver figura 2.11).

Un sistema de comunicación de datos necesita software, el cual, llamado *programa terminal* o TTY, controla al Transmisor-Receptor IR y efectúa el manejo apropiado de la transferencia de información entre éste y la pantalla.

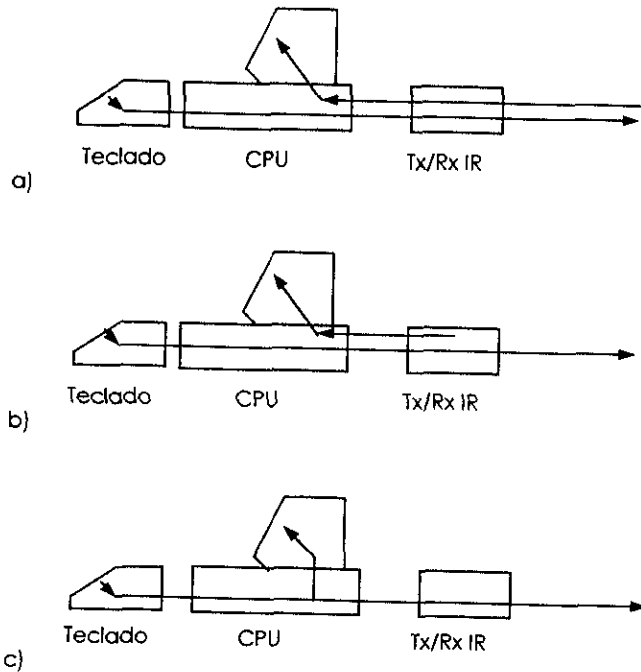


Fig. 2.11. Algunas maneras en que se exhiben en el monitor los caracteres escritos al operar en transmisión a) full-dúplex y b) y c) half-dúplex, b) y c) se distinguen según si el programa terminal TTY o el Transmisor-Receptor IR gobiernan la exhibición en el monitor.

## 2.5. La interface de comunicación serie (SCI)

El subsistema asíncrono de entrada/salida que tiene integrado el microcontrolador MC68HC811E2, el cual empleamos en este proyecto, se llama Interface de Comunicación Serie (Serial Communications Interface, por sus siglas en inglés SCI, ver pagina 120 del **Anexo A**). El SCI es un UART, que emplea el formato NRZ (un bit de inicio, 8 ó 9 bits de datos, y un bit de paro), y además cuenta con varias velocidades de transmisión disponibles.

El formato de datos del SCI requiere las siguientes condiciones:

1. Una línea inactiva en estado alto antes de la transmisión o recepción de un mensaje.

2. Un bit de inicio (cero lógico) que indique el principio de cada carácter.
3. El dato que es transmitido y recibido es el primer bit menos significativo.
4. Un bit de paro (uno lógico) empleado para indicar el fin de una trama. (Una trama consiste en un bit de inicio, un carácter de 8 ó 9 bits, y un bit de paro.)
5. Una pausa (definida como la transmisión o recepción de un cero lógico a causa de un múltiple número de tramas).

El subsistema SCI se emplea principalmente para la comunicación de datos. Lo anterior usualmente es aplicado hacia dispositivos remotos que necesitan intercambiar información, ya sea por un cable, por líneas telefónicas, por transmisores de radio, o por transmisores IR. El SCI utiliza el pin PD1 del puerto D como línea de transmisión (TxD) y el pin PD0 del mismo puerto como línea de recepción (RxD), ver la figura 2-4 del **Anexo A** (pagina 137). Estas líneas se pueden habilitar o deshabilitar por medio de uno de los registros de control del SCI (SCCR2). Cuando están deshabilitadas, el puerto D se puede utilizar como un puerto paralelo de entrada/salida determinado o configurado por medio del registro de direcciones (DDRD). Si la línea de transmisión y/o la de recepción son habilitadas, el SCI tiene el control de estas líneas, no haciendo caso de la configuración con la que cuente el DDRD. Lo anterior se encuentra mas a detalle en el **Anexo B**.

Para el 68HC811, las rutinas de programación en la transmisión de una palabra necesitan ser escritas dentro de un registro de datos (SCDR). El SCI proporciona las tramas de datos requeridas para realizar la comunicación, pero sin paridad. El microcontrolador lee la entrada serie por medio del registro de datos para obtener cada carácter recibido. El SCI, automáticamente convierte cada trama recibida (carácter) en un byte paralelo.

El SCI emplea internamente un reloj, que junto con el reloj principal del microcontrolador permiten la transferencia y muestreo de los datos. Lo anterior es realizado por otro registro de control (BAUD), que se puede programar para elegir la velocidad de transmisión a la que se transmitirán o recibirán los datos. Las velocidades de transmisión están disponibles por medio de fracciones integrales de la frecuencia de reloj (E), para mayor información ver la tabla 9-3 del **Anexo B** (pagina 145). El SCI no tiene la manera de checar la paridad, y ni de generarla, es decir, automáticamente el bit de paridad no es insertado cuando se transmite y



## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

cuando se recibe no hay bit que cheque la paridad recibida. Sin embargo pueden realizarse programas que permitan detectar errores de transmisión.

El transmisor interno del SCI puede enviar señales de pausa (SBK) por medio de la programación del registro de control 2 (SCCR2), esta señal se utiliza para que el receptor este atento. El receptor interno del SCI se puede desactivar por medio de otro bit, y ser activado por el bit de control RWU, el cual se encuentra en el mismo registro. Otro bit, el wake-up (WAKE), en el registro de control 1 (SCCR1), también permite que el receptor ponga atención.

El SCI emplea doble buffer para transmitir y para recibir información (ver figura 2.12). Este doble buffer opera por medio de un registro de datos (SCDR) cuya dirección de memoria es la \$102F, ver la tabla 4-1 del **Anexo A** (pagina 136). La forma de operar es la siguiente: Si se escribe un dato en el SCDR, este es enviado físicamente por el registro transmisor de datos (TDR), y si se lee un dato en el SCDR, este es recibido físicamente por el registro receptor de datos (RDR).

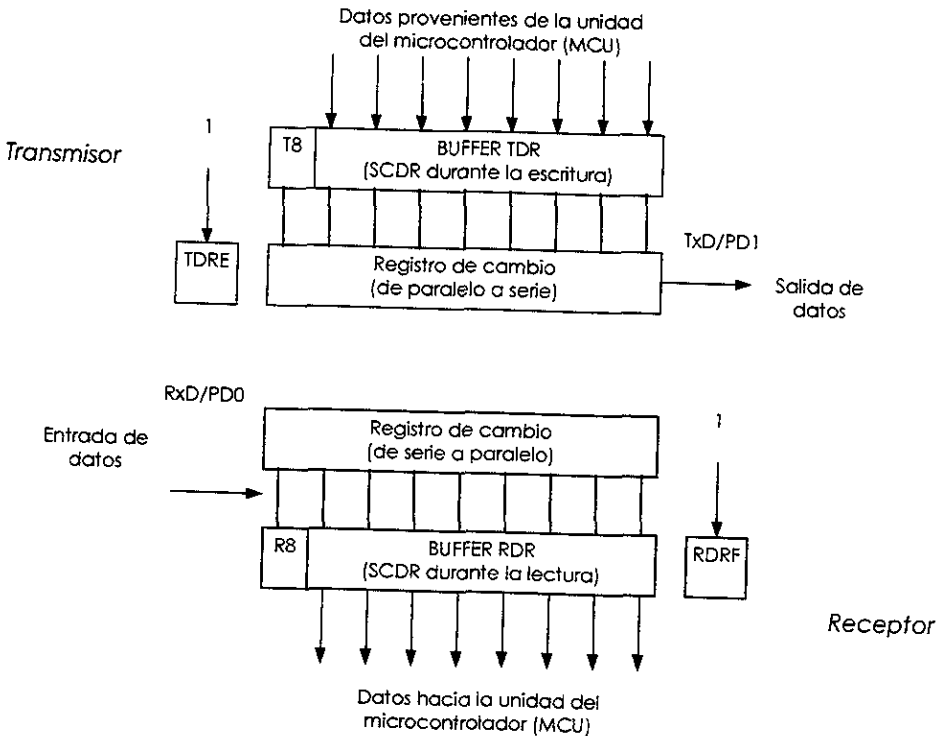


Fig. 2.12. Doble Buffer en el SCI.

La manera de operar del transmisor es la siguiente; para transmitir, el sistema escribe un byte en el registro transmisor de datos (TDR), el cual es transferido hacia la salida del registro de cambio. Si se requiere transferir un noveno bit se transmite el bit 8, el cual es habilitado en el registro SCCR1 (T8). El registro de cambio automáticamente adiciona a la salida (es decir, a los 8 bits de datos que vienen del TDR) un bit de inicio, y un bit de paro, marcando con estos bits el principio y el fin de la trama. La velocidad de transmisión se obtiene al configurar el registro BAUD.

Este sistema tiene una bandera que indica que el registro transmisor de datos esta vacío (TDRE, Transmit Data Register Empty), muchas de las ocasiones esta bandera se "prende" cuando el dato se transfiere del buffer TDR al registro de cambio. Por medio de la programación se detecta si la bandera TDRE esta "prendida" o no. La rutina que maneja el transmisor puede responder a la escritura de otro byte en el buffer TDR, esta escritura a su vez, puede limpiar ("apagar") la bandera TDRE.

El término doble buffer se refiere al hecho de que mientras el registro de cambio proporciona un byte a la salida, el buffer de datos almacena el siguiente byte a ser transmitido. Por tanto, el sistema puede escribir el segundo byte mientras el primero es tranquilamente transmitido.

Cuando el registro de cambio proporciona un byte a la salida de manera completa, el SCI automáticamente transfiere el contenido del buffer TDR hacia el registro de cambio. Esto también causa que el sistema "prenda" la bandera TDRE indicando por programación que puede escribir otro byte en el buffer TDR.

Ahora en el lado de recepción tenemos lo siguiente: Si el receptor esta esperando, es decir en condición de inactivo, la primer señal que le llega es de estado bajo, un bit de inicio, inmediatamente después se prepara para esperar los siguientes 8 bits. Si esta configurado para 9 bits, esperará los 9 bits que le lleguen. La configuración del registro BAUD determina la velocidad de muestreo. Cuando todos los bits son recibidos, el sistema los transfiere del registro de cambio al registro receptor de datos (RDR). Si se presenta una configuración para 9 bits, también se transfieren los 9 bits por medio del bit R8, habilitado mediante el registro SCCR1. Durante la transferencia de estos datos, también se "prende" una bandera, la cual indica que el receptor esta listo para recibir datos (RDRF, Receive Data Register Full), conociendo así, por programación, que dato esta listo para ser leído. Posteriormente el receptor muestrea el bit de paro y espera la llegada de otro bit de inicio.

La rutina que maneja el receptor permite leer el buffer RDR (lee el registro SCDR) mientras que el siguiente dato es recibido. Lo anterior es también un doble buffer. En este caso el buffer almacena el byte recién recibido, aunque al mismo tiempo se siguen recibiendo más datos por medio del registro de cambio. La programación permite leer el buffer antes de que el siguiente byte sea completamente recibido. Al leer el buffer RDR se vuelve a "prender" la bandera RDRF y se habilita el siguiente byte recibido.

A continuación mencionaremos la programación del SCI, la cual se apoya en el empleo de los registros de comunicación propios de este subsistema. En principio la programación permite inicializar el SCI, por medio de una subrutina que se ejecuta después de proporcionarle un reset al microcontrolador. Otra subrutina puede diseñarse para que se transmita un carácter o para que se reciba un carácter. El procedimiento de configuración del SCI es el siguiente:

1. Seleccionar la velocidad de transmisión. Utilizar el registro BAUD para seleccionar la velocidad a la que se desee transmitir (9600, 4800, ó 2400).
2. Seleccionar el formato del carácter. Utilizar el registro de control SCCR1 (Serial Communications Control Register 1) para seleccionar si se requieren 8 ó 9 bits de datos.
3. Habilitar interrupciones de transmisión y recepción. Utilizar el registro de control SCCR2 (Serial Communications Control Register 2) para habilitar el Transmitter Enable (TE) o el Receiver Enable (RE).

El procedimiento anterior nos permite inicializar el SCI, ahora para poder transmitir un byte, tenemos que seguir el siguiente procedimiento:

1. Proporcionarle al SCI la interrupción de sus circuitos lógicos. Leer el registro de estado SCSR (Serial Communications Status Register) para limpiar banderas.
2. Escribir un carácter en el registro de datos SCDR (Serial Communications Data Register) para que pueda ser transmitido.

Como ya tenemos inicializado el SCI y podemos transmitir, ahora solamente nos falta recibir caracteres, para lograr lo anterior, tenemos que seguir los siguiente puntos:

1. Proporcionarle al SCI la interrupción de sus circuitos lógicos. Leer el registro de estado SCSR para limpiar banderas. Este registro permite la detección de errores.

2. Leer el registro de datos SCDR para que el carácter pueda ser recibido.

3. Opción para detectar errores. Error de Rearranque (OR, Overrun error flag), Error de Ruido (NF, Noise error Flag), y Error de Trama (FE, Framing Error).

Como hemos visto existen 5 registros de comunicación serie que forman parte del SCI, los cuales a continuación serán revisados un poco más a detalle. Si se desea profundizar aún más sobre estos registros en el **Anexo B** se muestran los bits de cada uno de estos registros así como los nombres y funciones de sus señales.

El **SCDR** es un registro de datos paralelo que desempeña dos funciones. Esto es, es un registro receptor de datos (buffer RDR) cuando se lee, y un registro transmisor de datos (buffer TDR) cuando se escribe. Podemos observar lo anterior si revisamos la figura 2.12. Como ya se había mencionado tanto el receptor como el transmisor son un doble buffer.

El registro **BAUD** permite seleccionar la velocidad de transmisión por medio de los bits SCP0 y SCP1, los cuales funcionan como un prescaler. La combinación de estos bits proporcionan diversas velocidades de transmisión proporcionadas por la frecuencia del cristal externo. En el apéndice A se muestran las velocidades de transmisión para un cristal de 8 MHz. Hay que tener en cuenta que el sistema de reloj (E) tiene una frecuencia de  $\frac{1}{4}$  de la frecuencia del cristal (E = 2 MHz).

El microcontrolador 68HC811 tiene un generador de tasa que divide el sistema de reloj entre 16. Por tanto la máxima tasa de velocidad posible es  $E/16$  (o la frecuencia del cristal/64). Los bits en el registro BAUD proporcionan una división adicional de esas frecuencias. El sistema divide la frecuencia del cristal entre 4 y por consiguiente se dividen los factores del prescaler tal y como se especifican en el registro BAUD. Esta frecuencia se emplea para manejar el muestreo del receptor. El sistema además divide esta frecuencia de muestreo entre 16 para el reloj del transmisor. El resultado es que el receptor emplea un reloj de muestreo que es 16 veces la velocidad de transmisión. Por lo tanto, se puede introducir cada bit en 16 intervalos iguales. Una vez que el receptor tiene detectado el bit de inicio, comienza a muestrear cada bit de los ocho de datos, junto con el bit de paro en el octavo, noveno y décimo intervalo. El valor del bit es considerado como el valor de la mayoría de las muestras. Lo anterior se puede observar en la figura 2.14.

Para un dispositivo que recibe datos transmitidos desde otro dispositivo, estos deben operar a la misma velocidad de transmisión

o casi a velocidades similares. El problema es para el receptor, el cual estará determinando los límites de cada bit de inicio. Esto es necesario para el muestreo del receptor así como para obtener los valores lógicos correctos de cada bit recibido. La velocidad de transmisión tanto del transmisor como del receptor puede diferir arriba de un 4% sin que el receptor pierda algún bit. Por ejemplo, el sistema de reloj está dividido entre 13, lo cual exactamente no da como resultado una velocidad de 9600 bps, pero está dentro de los límites de tolerancia.

A continuación se presentan algunos ejemplos para configurar el registro BAUD.

EJM1.- Con un cristal de 4 MHz, la máxima velocidad de transmisión posible es de  $4 \text{ MHz}/64 = 62.5 \text{ Kbps}$ . Si el registro BAUD se configura con el código %00010011, la velocidad seleccionada es:

$$62.5 \text{ Kbps}/3/8 = 2604.17, \text{ redondeando, se tiene: } 2604 \text{ bps}$$

EJM2.- Con un cristal de 8 MHz, la máxima velocidad de transmisión posible es de  $8 \text{ MHz}/64 = 125 \text{ Kbps}$ . Si el registro BAUD se configura con el código %00110010, la velocidad seleccionada es:

$$125 \text{ Kbps}/13/4 = 2403.85, \text{ redondeando, se tiene: } 2404 \text{ bps}$$

La velocidad de transmisión anterior es de aproximadamente 2400 bps, lo cual está dentro de la tolerancia especificada.

El registro de control **SCCR1** proporciona los bits de control (bit M) que determinan el tamaño del carácter (8 ó 9 bits de datos) y selecciona el método para "despertar" (bit WAKE) a un receptor "dormido". El bit de control Wake-up se utiliza para que el receptor ponga atención a sea despertado. Existen dos métodos para despertar a un receptor. El primero es para cuando la línea está inactiva, a esto se le llama despertar línea inactiva (Idle Line Wake-up), y se utiliza para cuando los mensajes son relativamente largos (más de 10 bytes). Para seleccionar este método, la programación consiste en poner a cero el bit WAKE. Si el SCI utiliza este método, se debe de establecer un protocolo. Por ejemplo, los primeros bytes le indican la dirección al receptor. Cuando la programación en el receptor establece que el mensaje es una dirección del otro dispositivo, obliga que el receptor se ponga a dormir. El receptor no puede despertarse hasta que la línea regrese otra vez a su estado inactivo o de reposo.

El otro método es para cuando se pone a uno el bit más significativo de un carácter, a esto se le llama despertar señal de dirección (Address-

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

Mark Wake-up), considerando a este carácter como un carácter de dirección. Este método se emplea cuando los mensajes son cortos. En el caso de que se requiera elegir este método la programación consiste en poner a uno el bit WAKE.

Si el proceso Wake-up es automático, el buffer del receptor se llena y causa que la bandera RDRF, se "prenda". Esto depende de la programación, la cual detectará una interrupción.

El bit M en este registro de control como ya habíamos mencionado permite seleccionar el tamaño del carácter o el formato de 8 ó 9 bits que deseemos. Si se pone a uno el bit M ( $M = 1$ ), se selecciona el formato de 9 bits (un bit de inicio, 9 bits de datos, y un bit de paro), pero si se pone a cero ( $M = 0$ ), se selecciona el formato de 8 bits (un bit de inicio, 8 bits de datos, y un bit de paro). Esto es muy útil si una trama emplea 2 bits de paro u 8 bits de datos con paridad. Hay que tomar muy en cuenta que el bit M no permite seleccionar automáticamente la generación o detección de paridad. Pero subrutinas de programación pueden emplear el bit T8 y el bit R8 para manejar un caso como el anterior. Estos bits solo pueden ser usados para distinguir 8 bits de datos desde 8 direcciones por medio del método address-mark wake-up.

Para el bit R8 se tiene, que si M se pone en uno, se lee el noveno bit en R8, ocasionando que esta lectura actualice los bits que se encuentran en el registro SCDR. Para el bit T8 se tiene, que si M se pone en uno, se escribe el noveno bit en T8, lo cual también actualiza el registro SCDR.

El registro de control **SCCR2** proporciona los bits de control que habilitan o deshabilitan las funciones individuales del SCI. Los bits más importantes de este registro son:

El bit **TIE** (Transmit Interrupt Enable, es decir, Habilita Interrupción en el Transmisor). Si este bit se pone en uno, el SCI genera una interrupción cuando el registro SCDR esta vacío. Esto ocurre cuando el buffer TDR transfiere los datos hacia el registro de cambio, como se muestra en la figura 2.12. El MCU puede responder para escribir otro byte en el registro SCDR.

El bit **RIE** (Receive Interrupt Enable, es decir, Habilita Interrupción en el Receptor). Si este bit se pone en uno, el SCI genera una interrupción cuando el byte recibido es transferido del registro de cambio al buffer RDR. El MCU puede responder para leer el registro SCDR antes de que otro byte sea recibido.

El bit **TE** (Transmitter Enable, es decir, Transmisor Habilitado). Si este bit se pone en uno, el pin PD1 es un pin TxD, en otras palabras, para el registro de direcciones DDRD este pin esta definido como salida (transmisor).

El bit **RE** (Receiver Enable, es decir, Receptor Habilitado). Si este bit se pone en uno, el pin PD0 es un pin RxD, en otras palabras, para el registro de direcciones DDRD este pin esta definido como entrada (Receptor).

El bit **RWU** (Receiver Wake-Up Control, es decir, Control de Atención del Receptor). Si este bit se pone en uno, el receptor se duerme ignorando cualquier actividad. Como mencionamos anteriormente el bit WAKE permite seleccionar el método para despertar al receptor.

El bit **SBK** (Send Break, es decir, Envío de Pausa). Si este bit se pone en cero, el transmisor opera normalmente, pero si se pone en uno, el transmisor envía continuamente "ceros" hasta que el bit SBK sea limpiado. Cabe señalar que el carácter de pausa contiene 10 u 11 "ceros".

El quinto y último registro con que cuenta el SCI es el **SCSR**. Este registro de estado proporciona entradas hacia los circuitos lógicos internos del MCU para generar interrupciones en el SCI. Las banderas más empleadas e importantes de este registro de estado son:

**Bandera TDRE** (Transmit Data Register Empty, es decir, Registro Transmisor de Datos Vacío). Esta bandera se prende cuando el buffer TDR (registro SCDR durante la escritura) realiza una transferencia hacia el registro de cambio, en otras palabras, se prende cuando el registro de datos SCDR esta vacío. Para limpiar esta bandera, se debe de leer el registro de estado SCSR y escribir en el registro de datos SCDR.

**Bandera RDRF** (Receive Data Register Full, es decir, Registro Receptor de Datos Lleno). Esta bandera se prende cuando el registro de cambio realiza una transferencia hacia el buffer RDR (registro SCDR durante la lectura), en otras palabras, se prende si un carácter recibido esta listo para ser leído desde el registro de datos SCDR. Para limpiar esta bandera, se debe de leer el registro de estado SCSR y leer el registro de datos SCDR.

**Bandera OR** (Overrun error, es decir, Error de Rearranque). Esta bandera se prende cuando un nuevo carácter es recibido antes de que otro carácter que se recibió previamente sea leído desde el registro de datos SCDR. Cuando este error ocurre, el dato que causa el re arranque se pierde y el dato que se recibió previamente no sufre ninguna alteración. Ver la figura 2.13. Para limpiar esta bandera de error, se debe de leer el registro de estado SCSR y leer el registro de datos SCDR.

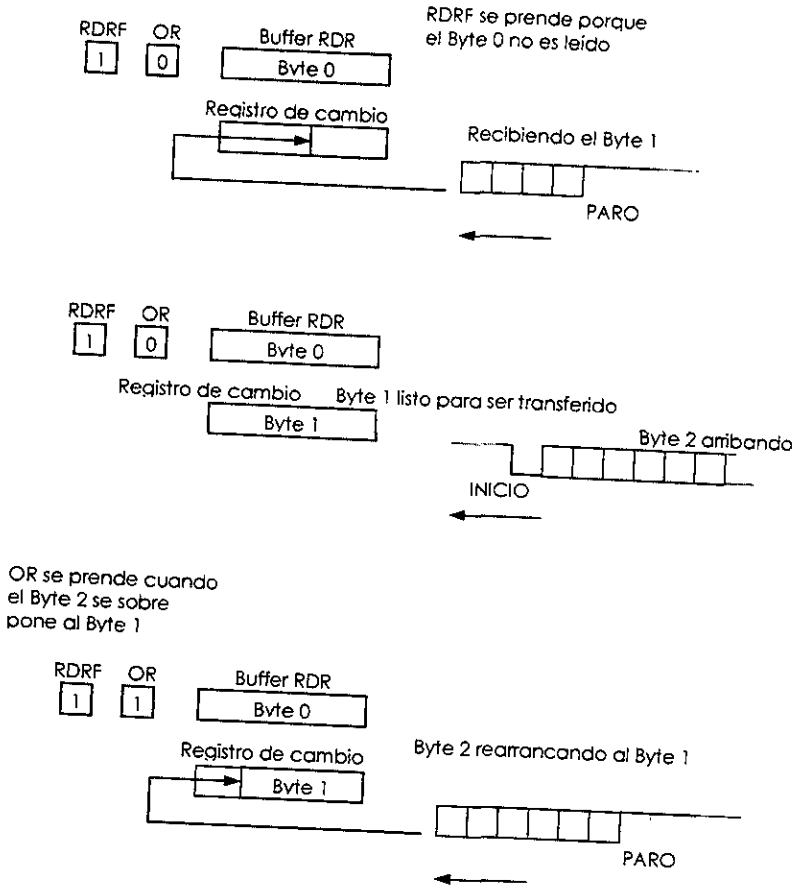


Fig. 2.13. Error de Rearranque.

**Bandera NF** (Noise error Flag, es decir, Error de Ruido). Esta bandera se prende cuando la mayoría de las muestras detectan cualquier otra cosa que no sea una muestra lógica. Como ya se había mencionado el receptor emplea un reloj de muestreo que tiene una frecuencia de 16 veces la velocidad de transmisión, cuando el receptor tiene establecidos los límites de un bit, se muestrea de acuerdo a los ciclos 8, 9 y 10 del reloj de muestreo, si durante estos ciclos se detecta una muestra que no es idéntica a las demás, quiere decir que existe un error de ruido. Ver la figura 14, donde se muestra un pico de ruido durante un bit de estado bajo.



Para limpiar esta bandera de error, se debe de leer el registro de estado SCSR y leer el registro de datos SCDR.

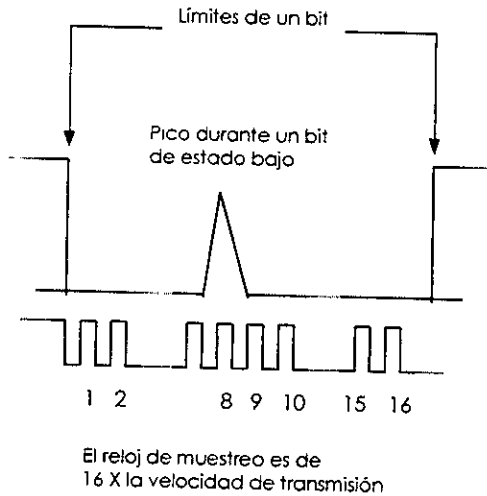


Fig. 2.14. Error de Ruido.

**Bandera FE** (Framing Error, es decir, Error de Trama). Esta bandera se prende cuando se detecta un cero en lugar de un bit de paro. En otras palabras, se prende si el sistema detecta un bit de estado bajo después de que todos los datos fueron recibidos (incluyendo la paridad). Este error ocurre debido a la velocidad de transmisión entre transmisor y receptor. Otra causa puede ser por el formato de tramas que se emplee. El ejemplo que se muestra en la figura 2.15, puede comprenderse aceptando que el transmisor emplea una trama que contiene 9 bits de datos y el receptor una trama que contiene 8 bits de datos. El receptor detecta el último bit de la trama como un bit de estado bajo ocasionando que se prenda la bandera, sin considerarlo como un bit de paro puesto que este debe de ser de estado alto (uno lógico) como lo indican las condiciones del formato de datos para el SCI. Para limpiar esta bandera de error, se debe de leer el registro de estado SCSR y leer el registro de datos SCDR.

## Capítulo III

### 3. REALIZACIÓN DEL ENLACE DE COMUNICACIÓN IR

Anteriormente se menciona que la idea de utilizar radiación IR en la comunicación no es nada nueva, siendo que existen sistemas de uso militar desde hace más de treinta años. En un sistema IR, el haz de luz es invisible y por lo tanto la existencia de un enlace de comunicación pasa desapercibida. Al contrario de los sistemas de comunicación por radio, la comunicación por radiación IR prácticamente no tiene fugas, lo cual con la ayuda de lentes, esta radiación se vuelve más directiva. Probablemente, estos dos factores de seguridad son los que han hecho a los sistemas de comunicación por radiación IR interesantes para aplicaciones militares.

Es discutible que los sistemas de comunicación por radiación IR tengan una ventaja real sobre otros métodos en aplicaciones no militares, pero sin duda representan un tipo de proyecto interesante y además, proporcionan bajo costo, fácil manejo, disponibilidad de componentes, y no se requiere licencia para transmitir como es el caso de los enlaces por RF (Radio Frecuencia).

Los sistemas IR brindan muy buenos resultados si se emplea un sistema de modulación en frecuencia que opere en onda continua (OC).

#### 3.1. Ejemplo de un sistema de comunicación inalámbrico IR

El propósito fundamental de la comunicación de datos es el intercambio de información entre dos puntos específicos (fuente y destino), independientemente del medio de transmisión que se utilice. En la figura 3.1 se muestra un sistema de comunicación IR típico, donde la información que se intercambiara es un mensaje etiquetado con la letra  $m$ . Esta información es representada como datos  $g$  y generalmente se presenta hacia el transmisor en la forma de una función en el tiempo,  $g(t)$ .

Los términos datos e información son definidos en la tabla 3.1. Para que la señal  $g(t)$  pueda ser transmitida, necesita ser convertida en una señal que se adapte a las características del medio de transmisión (en este caso es el espacio libre). Esta señal representada ahora por  $s(t)$  es, entonces transmitida a través del medio. Al otro lado del sistema de comunicación, una señal  $r(t)$ , aunque un poco diferente en la forma de la señal  $s(t)$ , es recibida. Esta señal es convertida dentro del receptor a una forma apropiada de salida, la cual se representa por  $g'(t)$  o datos  $g'$ , siendo una aproximación de la entrada de datos  $g$  o de la señal  $g(t)$ . Finalmente, el dispositivo de salida presenta el mensaje estimado,  $m'$ , hacia el punto destino.

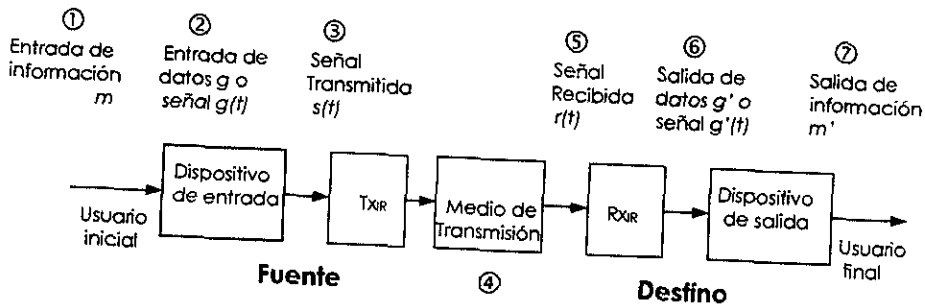


Fig. 3.1. Diagrama a bloques de un sistema de comunicación IR típico.

La forma de aplicar el ejemplo anterior al sistema de comunicación inalámbrico IR que se pretende realizar, es mediante ha analogías que podemos considerar, como:

1. - El usuario inicial en nuestro sistema sería el operador, el cual estaría mandando una serie de comandos o datos de control, con la ayuda de un computador, hacia el vehículo robotizado o a cualquier otro dispositivo que se desee manipular a distancia. Es aquí donde se le dice al dispositivo remoto (que puede ser un vehículo, o un brazo robot) la tarea a realizar.
2. - El dispositivo de entrada sería el teclado del computador, el cual permite al operador mandar la lista de tareas hacia el dispositivo remoto para que éste las realice, por lo tanto del teclado se pasan (las ordenes al transmisor y posteriormente las recibe el microcontrolador).
3. - El transmisor es la parte de diseño que permitirá mandar los datos modulados y amplificados a una velocidad de 2,400 bps por rayos de luz infrarrojos y a una distancia de 6 metros.

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

4. - El medio de transmisión será el espacio libre, donde la radiación infrarroja se propagará.

5. - El receptor es la parte de diseño que permitirá recibir los datos, demodulándolos y regenerándolos a una velocidad de 2,400 bps; percibiendo rayos de luz infrarrojos con un ángulo de recepción de 30°.

6. - El dispositivo de salida sería el microcontrolador o un computador, el cual al momento de recibir los datos se encargara de arrancar los programas residentes en su memoria para realizar algún proceso.

7. - El usuario final sería lo que tenga que controlar el microcontrolador o el computador, que puede ser el vehículo robotizado, el brazo robot, u otro dispositivo, donde cualesquiera de estos realizarán las tareas encomendadas del operador que se encuentra a una distancia lejana. En la tabla que se presenta a continuación se mencionan algunas definiciones importantes de los términos Datos e Información.

Tabla 3.1. Datos e información

<b>Datos</b>	<b>Son la representación de hechos, conceptos, o instrucciones presentados de manera conveniente para comunicar, interpretar, o procesar pensamientos e ideas. Expresión general englobando los números, letras, y símbolos que constituyen la entrada de un proceso de computador.</b> <b>Analógicos.- Datos representados en forma continua.</b> <b>Digitales.- Datos representados por una sucesión de caracteres codificados.</b>
<b>Información</b>	Es la intención de circular un conjunto de estos datos asignados por pensamientos e ideas. Colección de hechos u otros datos. <b>Inteligente.- Datos, información, o mensajes destinados a ser transmitidos.</b>

Finalmente, en general la **fente** comprendería al operador manipulando al computador y al transmisor, utilizando el espacio libre para llegar al **destino**, el cual comprendería al receptor, al microcontrolador o al computador y al dispositivo remoto. Lo anterior será la forma en que el sistema de comunicación inalámbrico IR se verá presentado en la práctica, aunque en dos vías.

## 3.2. Fundamentos

Antes de pasar a los puntos 3 y 5 del ejemplo anterior, mencionaremos los fundamentos básicos de un sistema de comunicación IR. En la figura 3.2 se muestra un esquema generalizado de un sistema IR. Todo lo que se necesita es enviar el haz IR con la suficiente potencia y a una frecuencia adecuada para que pueda ser recibido. Esto se ve muy fácil, pero se necesitan diseños muy elaborados y además circuitos específicos, los cuales se mencionaran más adelante.



Fig. 3.2. Esquema generalizado de un sistema de transmisión de datos IR.

El reto de la parte de recepción es tratar de bloquear los distintos factores a que esta expuesto para lograr un buen desempeño. El primer factor al que se enfrenta es a la luz que existe en el ambiente. Como la señal IR transmitida es muy débil (considerada como una fuente de luz de relativamente alta potencia), con respecto a fuentes locales incandescentes, fluorescentes o a la misma luz solar, es muy posible que ocurran interferencias, lo cual es un factor de mucha importancia.

Existen dos elementos que contribuyen en este problema. El primero, es cuando el nivel de estimulación que existe en el ambiente afecta al detector causando baja sensibilidad, y en el peor de los casos, saturación. El segundo elemento proporciona un nivel frecuente de ruido de aproximadamente 60 dB afectando directamente a la señal, pero especialmente a frecuencias bajas como de 50 ó 60 Hz. También, la sensibilidad de un fotodetector se extiende hasta el rango de la luz visible, ocasionando demasiada interferencia producida por fuentes con significativa potencia como lámparas incandescentes y luz solar. Estas fuentes producen grandes cantidades de energía infrarroja, especialmente el Sol.

Para lograr un adecuado enlace entre la fuente de radiación IR y el fotodetector, debe de existir una buena excitación de corriente directa en el emisor, si no se cumple lo anterior se tendrá un enlace de datos lento a una distancia muy corta.

Pero la mayoría de los enlaces IR necesitan distancias mayores que 30 cm, y velocidades de transmisión mayores que 300 baudios, y además inmunidad a elementos que produzcan interferencia. Para esto se necesitan altas frecuencias, manejar corriente alterna para facilitar la amplificación de la señal detectada y filtrar las componentes de baja frecuencia.

Con respecto a la distancia de operación que necesita un enlace IR, el presente proyecto cuenta con una distancia de aproximadamente seis metros, encontrándose dentro de los sistemas de comunicación IR que operan a distancias moderadas, es decir, entre los 30 centímetros hasta los 10 metros. Cabe mencionar, que existen otros sistemas que operan a distancias grandes (de 10 a 100 metros), los cuales se apoyan de instrumentos ópticos, como lentes, para aumentar su distancia aunque disminuyendo la calidad en la transmisión de la información, puesto que necesitan de una alineación muy precisa entre emisor y receptor.

En la mayoría de las aplicaciones para enlaces IR que operan a distancias moderadas, siendo la distancia el parámetro principal en que se diseño el presente proyecto de tesis, se emplean frecuencias portadoras entre los 20 y 50 KHz proporcionando velocidades de transmisión hasta 5,000 baudios, y además a un costo modesto. El transmisor diseñado para este tipo de distancias de operación, emplea una fuente de radiación IR capaz de realizar transmisiones a 40 KHz, ver la figura 3.3. Como se puede observar, se emplea una compuerta lógica AND y un transistor operando en corte y saturación para que el diodo emisor infrarrojo (Infrared Emitting Diode, IRED) conmute de acuerdo a los cambios que proporcionan los datos a ser enviados. Estos dispositivos proporcionan buenos resultados debido a que la frecuencia portadora no es muy grande. Este transmisor IR emplea una técnica conocida como OC (onda continua), la cual manipula a la frecuencia portadora de relativamente alta frecuencia en on y en off.

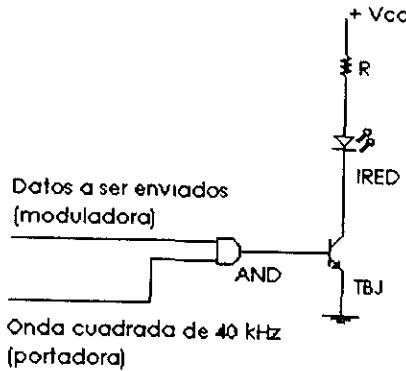


Fig. 3.3. Circuito transmisor IR operando en onda continua.

Hoy día en muchas áreas los IREDs son capaces de manejar corrientes pico de un ampere, pero en la práctica trabajan a la mitad de este valor. Algunos diseños que usan el 50 % de la señal cuadrada, presentan corrientes en el diodo entre 100 y 500 mA. Esto es muy importante puesto que los IREDs manejan a la salida este tipo de corrientes.

Ya que se menciona la parte óptica del transmisor IR, ahora comentaremos sobre el dispositivo que maneja al IRED, el transistor bipolar, que desempeña un buen trabajo requiriendo corrientes base entre 10 y 20 mA o más logrando con esto que se asegure su saturación. Pero en algunos diseños se necesita asegurar una corriente de colector suficiente para que el IRED pueda mandar los datos ópticos con toda seguridad de que llegarán al lado de recepción, para solucionar este problema se emplea una configuración Darlington, la cual es una conexión de dos transistores bipolares para funcionar como un solo transistor "superbeta", donde su principal característica es que estos dos transistores actúan como una sola unidad con una ganancia de corriente que es el producto de las ganancias de corriente de los transistores por separado, ver la pagina 56-figura 3.9.

En la base del transistor bipolar, de igual forma si se emplea una configuración Darlington, esta la salida de la compuerta lógica AND, presentando en la entrada uno la señal moduladora y en la entrada dos la señal portadora de 40 KHz. Esta compuerta hace la función de un modulador, donde se realiza un proceso de multiplicación de dos frecuencias digitales.

La resistencia que acompaña al IRED en algunas ocasiones se maneja su valor entre 10 y 50  $\Omega$  proporcionando la corriente máxima a la salida del transmisor IR.

En el otro extremo del enlace IR se encuentra el proceso de recepción, que en principio cuenta con un dispositivo detector de radiación IR, llamado fototransistor, ver la figura 3.4. Este dispositivo produce huecos de electrones en respuesta a los fotones que chocan en su base. Por lo regular estos dispositivos son muy sensibles, trabajan con corrientes muy pequeñas del orden de microamperes y nanoamperes proporcionando además un balance respecto a la velocidad de respuesta, la cual es del rango de nanosegundos. Esto permite frecuencias de transmisión entre 50 y 100 KHz, contribuyendo a manejar amplias velocidades de transmisión, a lograr una buena amplificación, y también facilita el filtrado del ruido existente.

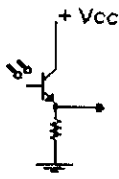


Fig. 3.4. Fototransistor detector de radiación IR.

En muchas aplicaciones de recepción de datos se sugiere colocar en la base del fototransistor un filtro óptico, con la finalidad de atenuar la porción visible del espectro electromagnético mientras permite que la radiación IR pase intacta. Este filtro óptico puede ser de un material especial, pero por lo regular se utiliza acrílico plástico color rojo opaco. Este filtro es colocado dentro del sistema como una sola pieza de material enfrente, a escasos milímetros de la base del fototransistor detector de radiación IR. En la figura 3.5 se puede observar lo explicado anteriormente.

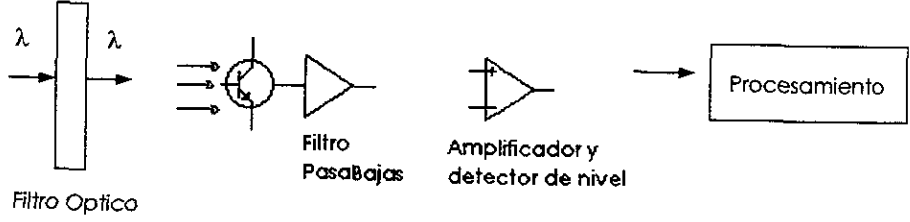


Fig. 3.5. Circuito receptor IR.



El filtro óptico está diseñado para proporcionarte una mayor área de captación de energía IR al fototransistor que se encuentra detrás de él. Usualmente el fototransistor y el filtro se montan de tal manera que parece como un sólo encapsulado.

Con respecto al filtrado de la luz visible, que se mencionó anteriormente, se aplica una atenuación a las frecuencias bajas que ocasionan la interferencia entre la región visible y la región IR. Esto se puede realizar mediante un acoplamiento de capacitancia cuyos valores son justamente elegidos por debajo de la frecuencia de transmisión.

Ya que la señal es detectada se eleva a un nivel suficiente de voltaje para posteriormente extraer los datos y procesarlos. Pero antes con la presencia de los pulsos de alta frecuencia provocan que se cargue un capacitor arriba del umbral de voltaje en unos cuantos ciclos, donde un comparador proporciona la señal con un nuevo nivel de voltaje. Cuando la señal (portadora) se encuentra ausente, el capacitor se descarga hasta que el umbral de voltaje del comparador alcanza significativamente el nivel lógico opuesto.

Con los cambios de señal ya detectados, es decir, presencia y ausencia de portadora, entran a la etapa de extracción de datos, donde se permite solamente el paso de la señal de baja frecuencia (moduladora). Ahora con esta señal moduladora, la cual es una aproximación de la señal que se tiene en el otro extremo, se amplifica y se asegura su nivel de voltaje para posteriormente introducirse en la etapa de procesamiento. En esta última etapa se regenera la señal recibida aproximándola a un más a la señal moduladora original. Con esto a la salida del receptor IR se tiene una copia aproximada de los datos originales, donde se acondicionan ya sea para entender un protocolo físico de comunicación (+/- 12 V) o para manejar niveles TTL (0 y 5 V).

Finalmente, empleando los fundamentos anteriores es posible desarrollar un enlace de comunicación IR entre dos dispositivos que necesiten comunicarse entre sí, ya sea para intercambiar datos o para ejercer una manipulación teniendo como resultado la realización de un trabajo predeterminado. Con las bases anteriores presentaremos a continuación a detalle el diseño de los circuitos de transmisión y recepción, así como las características generales de operación del sistema.

### 3.3. Diseño de los circuitos de transmisión y recepción

El ejemplo del punto 3.1 se refiere a un enlace IR con una sola vía de comunicación. En este punto mencionaremos un enlace IR completo, es decir, en dos vías o bidireccional, el cual permitirá que el computador y el dispositivo remoto se comuniquen con facilidad en ambas direcciones.

Tomando como punto de partida el circuito básico del transmisor IR y el circuito básico del receptor IR que se mencionaron en el punto anterior y además apoyándonos de los puntos de diseño 3 y 5 de la figura 3.1, podemos combinar todas estas ideas para desarrollar el circuito **TRANSMISOR-RECEPTOR IR** y el circuito **RECEPTOR-TRANSMISOR IR**, los cuales se presentan en la figura 3.6 y 3.7, respectivamente. Estos circuitos finales proporcionan el hardware del enlace IR considerados como un sistema de comunicación inalámbrico bidireccional. Más adelante mencionaremos el software empleado que complementa al enlace IR.

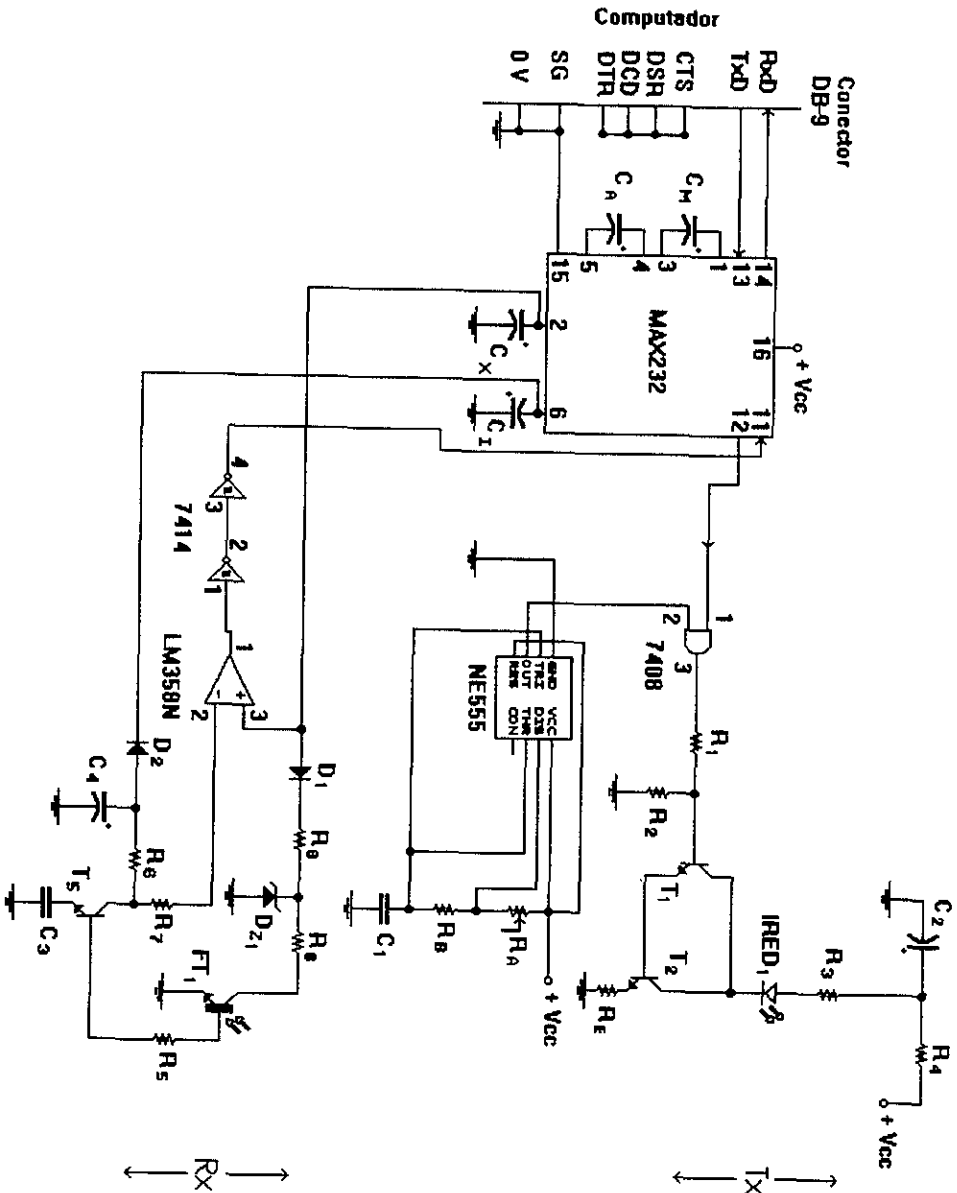


Fig. 3.6. Circuito Transmisor-Receptor IR conectado en el computador.

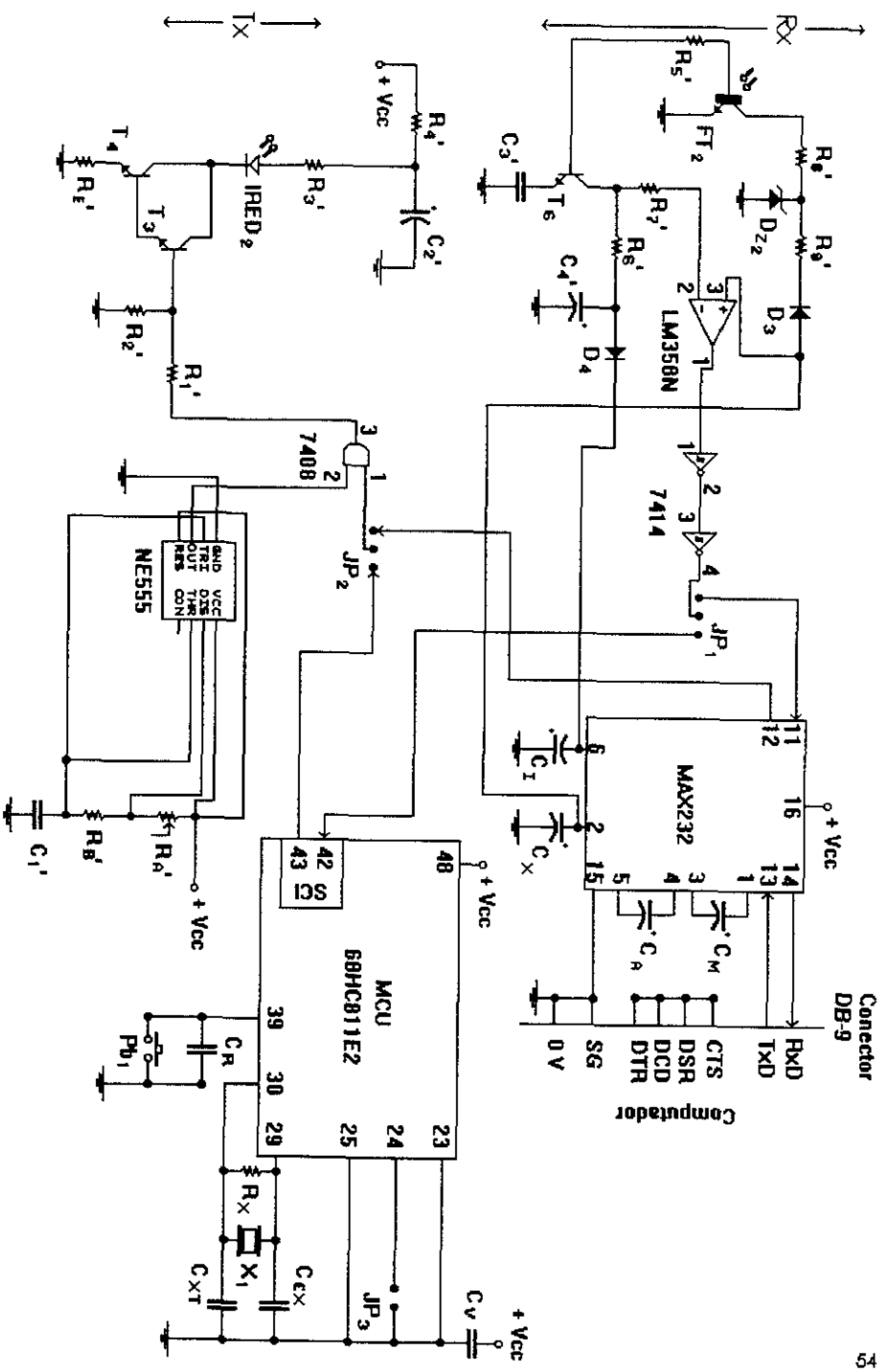


Fig. 3.7. Circuito Receptor-Transmisor IR conectado en el microcontrolador (MCU) con la opción de conexión hacia otro computador.

Como se puede observar en la figura 3.6 se cuenta con un dispositivo temporizador 555 que funciona como un multivibrador astable que proporciona una frecuencia portadora ( $f_p$ ) de 40 KHz según el arreglo que presenta en sus terminales. Mediante las resistencias  $R_A$  y  $R_B$  y el capacitor  $C_1$  se proporciona el valor de  $f_p$  según la ecuación 3.1. Como  $R_A$  es variable, proporciona a la salida del CI 555, pin OUT, una variación de frecuencia entre 36 y 40 KHz.

$$f_p = \frac{1.44}{(R_A + 2R_B) C_1} \text{ [Hz]} \text{ ----- (3.1)}$$

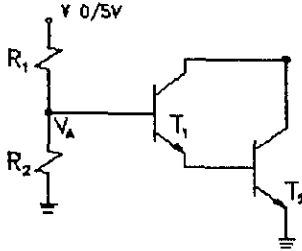
Por ejemplo, si  $R_A$  vale 2 K $\Omega$ , la frecuencia portadora mínima será de 36 KHz, y si  $R_A$  vale 1.6 K $\Omega$  se tendrá una frecuencia portadora de operación de 40 KHz. Si  $R_A$  llega a caer a bajo de 1.6 K $\Omega$  se proporcionarán frecuencias arriba de 40 KHz, las cuales no podrán ser utilizadas para realizar el enlace, puesto que el filtro del lado de recepción permite solo una  $f_p$  de aproximadamente 40 KHz. Regresando nuevamente al circuito de la figura 3.6, la frecuencia  $f_p$  proporcionada por el 555 es conectada en la entrada 2 de la compuerta AND (CI 7408), donde será multiplicada con respecto a lo que exista en la entrada 1. En esta entrada, pin 1, se conecta la frecuencia moduladora ( $f_m$ ) que viene del computador. Esta frecuencia tiene niveles de voltaje de +/- 12 Volts, debido a que maneja la norma RS-232, los cuales son convertidos a niveles de voltaje TTL por medio del CI MAX232, pudiendo con lo anterior manejar niveles de 0 a 5 Volts tanto para la  $f_m$  como para la  $f_p$ . Teniendo los mismos niveles de voltaje podemos multiplicar los pulsos que vienen del multivibrador astable con los pulsos provenientes del computador ya convertidos, produciendo una modulación de las dos señales por medio de la compuerta AND (CI 7408). Claro que la frecuencia del CI 555 es de mínimo el doble que la frecuencia en que vienen los datos del computador.

Cabe señalar que el circuito de la figura 3.6 y el circuito de la figura 3.7, están conectados hacia el computador, por medio de un conector DB-9 respetando la norma RS-232 y además las señales CTS, DSR, DCD y DTR se presentan configuradas en módem nulo. Ya teniendo una sola señal a la salida de la compuerta AND (que es la multiplicación de las dos señales anteriores) se introduce a un circuito Darlington (etapa de amplificación) que cuenta con un divisor de voltaje formado por  $R_1$  y  $R_2$  en la base del transistor  $T_1$ . Las resistencias  $R_1$  y  $R_2$  permiten dividir el voltaje según los cambios de nivel que presente la señal modulada, por ejemplo, si la señal

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

presenta un nivel de 5 Volts se tendrá en la base de T<sub>1</sub> un voltaje de aproximadamente 4.098 V, es decir, un uno lógico, y para cuando el nivel de la señal presente un nivel de 0 Volts se tendrá en la base de T<sub>1</sub> un cero lógico. Ver la siguiente figura.

SEÑAL MODULADA  
PROVENIENTE DE LA  
COMPUERTA AND.



$$V_A = \frac{V R_2}{R_1 + R_2} \quad (3.2)$$

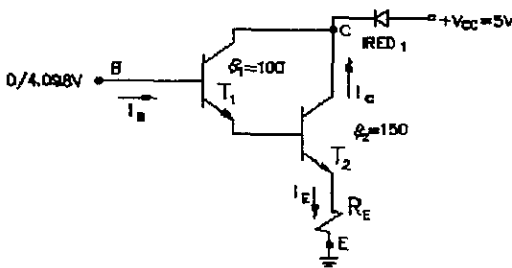
Si V=0V ⇒ V<sub>A</sub>=0V.

Si V=5V, SE TIENE:

$$V_A = \frac{(5)(10K\Omega)}{(2.2K\Omega + 10K\Omega)} = 4.0984 \text{ V}$$

Fig. 3.8. Divisor de voltaje en la base de T<sub>1</sub>.

El circuito Darlington formado por T<sub>1</sub> y T<sub>2</sub> permite aumentar la ganancia de corriente, la cual esta compuesta por el producto de las ganancias de corriente de cada uno de los transistores. Esto es, que el circuito Darlington amplificará la intensidad de la señal modulada ha aproximadamente 15,000 puesto que T<sub>1</sub> tiene una beta de 100 y T<sub>2</sub> tiene una beta de 150. Ver la siguiente figura.



β<sub>1</sub> Y β<sub>2</sub> SON BETAS MEDIDAS

$$\beta_T = \beta_1 \beta_2 \quad (3.3)$$

$$\beta_T = (100)(150) = 15,000$$

Fig. 3.9. Ganancia de corriente total proporcionada por el circuito Darlington.

La ganancia de corriente proporcionada por el circuito Darlington, la cual es la corriente de colector (I<sub>c</sub>), es suministrada hacia el cátodo del diodo emisor infrarrojo (IRED<sub>1</sub>). Esta corriente que recibe el IRED<sub>1</sub> es de aproximadamente 100 mA, como se muestra a continuación:

Si V<sub>cc</sub> = V<sub>c</sub> = 5 Volts y

Si V<sub>B</sub> = 4.098 V, ver la figura anterior, se tiene;

$$V_B = V_E + V_{BE} \text{ ----- (3.4)}$$

Despejando  $V_E$  de la ecuación anterior;

$$V_E = V_B - V_{BE} \text{ ----- (3.5)}$$

Sustituyendo valores en la anterior, se tiene;

$$V_E = 4.098 \text{ V} - 1.6 \text{ V} = 2.498 \text{ V}$$

El valor anterior de  $V_E$  es cuando se tiene un uno lógico en  $V_B$ . Cuando se presenta un cero lógico,  $V_E$  es igual a -1.6 Volts.

Apoyándonos de la ley de ohm, se obtiene;

$$V_E = I_E R_E \text{ ----- (3.6)}$$

Despejando  $I_E$  de la ecuación anterior;

$$I_E = V_E / R_E \text{ ----- (3.7)}$$

Si  $R_E = 25\Omega$  y  $V_E = 2.498 \text{ V}$  (uno lógico),  $I_E$  valdrá:

$$I_E = \frac{2.498 \text{ V}}{25\Omega} = 99.92 \text{ mA}, \text{ como la } I_E \text{ es aproximadamente igual a la } I_C, \text{ por lo tanto;}$$

$$I_E \approx I_C \approx 100 \text{ mA}$$

Cuando existe un cero lógico, se tiene;

$$I_C = \frac{-1.6 \text{ V}}{25\Omega} = -64 \text{ mA}, \text{ es una corriente inversa que polariza directamente al IRED}_1$$

En el otro extremo, es decir, en el ánodo del IRED<sub>1</sub> se cuenta con una resistencia muy pequeña ( $R_3$ ) que permite pasar la mayor parte de la corriente que suministra la fuente de alimentación ( $V_{CC} = +5 \text{ Vcd}$ ) y además un tiempo proporcionado por el capacitor  $C_2$  y la resistencia  $R_4$  de 2.2 mseg permitiendo tener una polarización directa en el IRED<sub>1</sub> durante este periodo de tiempo. El capacitor  $C_2$  desempeña una función muy importante en el circuito, la cual consiste en corregir el retardo originado por el arreglo Darlington.

Cuando el diodo emisor infrarrojo 1 (IRED<sub>1</sub>) presenta un voltaje positivo permanente en el ánodo, el cual es menor a otro voltaje positivo que se presenta en el cátodo, un uno lógico, éste se encuentra inversamente polarizado ocasionando que no emita radiación IR, es decir, no hay transmisión de información. Pero si en el cátodo del IRED<sub>1</sub>, se cambia rápidamente su voltaje a un valor negativo, un cero lógico, éste radiará luz IR estableciendo una transmisión de información. Ver la figura 3.10.

El encendido y apagado del IRED<sub>1</sub> por así llamarlo, está relacionado de acuerdo a la conmutación del circuito Darlington, cuando se le presenten bits en estado bajo o bits en estado alto de manera sucesiva, proporcionados a la salida de la compuerta AND.

En general, el transmisor IR opera en onda continua (CW, Continuous Wave), esto significa, que se mantiene un enlace constante entre emisor y receptor mediante la señal portadora. La manera de conmutar a esta señal portadora es en ON y OFF, como ya se explicó anteriormente.

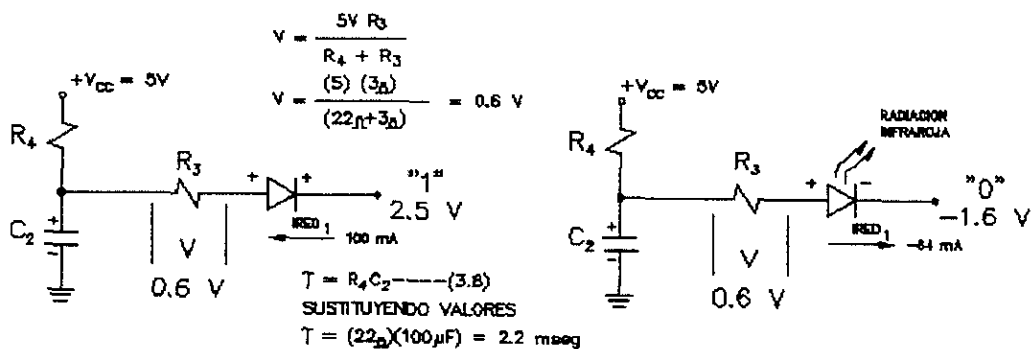


Fig. 3.10. Operación del diodo emisor infrarrojo (Transmisión de información).

Teniendo ya traducidos los datos que deseamos enviar y además montados en una frecuencia que nos sirve como camino para llegar al destino (lado de recepción) en un tipo de luz invisible para el ojo humano, damos paso a hacer uso del espacio libre donde esta señal IR se propagará a una frecuencia de 40 KHz sin que le afecten parásitos electromagnéticos, hasta el detector. Hay que tomar en cuenta que este haz de luz se atenúa conforme cruza el medio y más si pasa la distancia a la que está diseñado (aproximadamente seis metros). Durante el recorrido



ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

de este haz de datos, puede encontrarse con superficies que logren rebotarlo o desviarlo de su destino, haciendo que se degrade aun más o de lo contrario que llegue más directivo, por este motivo las pruebas que se le realizaron a este sistema, las cuales se presentan en la pagina 99, arrojaron un ángulo de transmisión de aproximadamente 32°, permitiéndonos basar en este dato para asegurar un enlace de comunicación limpio.

Al momento que el haz IR es detectado por el fototransistor FT2 del circuito de la figura 3.7, este elimina la frecuencia portadora, dejando solamente los datos, los cuales entran a una etapa de amplificación. FT2 no es un simple detector de radiación IR, sino que además internamente cuenta con un filtro pasa bajas. FT2 desempeña dos funciones; la primera es detectar la señal modulada o señal de información, y la segunda es cuando ya se tiene detectada esta señal, atenúa la frecuencia portadora y la frecuencia moduladora la deja intacta a la salida.

Para desempeñar la primer tarea, FT2 funciona como un simple detector de infrarrojos, esta tarea consiste en que al momento de detectar la presencia de luz por medio del elemento sensor (E), éste se va a saturación permitiendo que se tenga coherencia para los niveles de voltaje que se están manejando en todo el sistema (0 y 5 V), ver la figura 3.11. Si el elemento sensor (E) de FT2 no percibe luz, éste se va a corte presentando un voltaje Colector-Emisor (VCE ó VAR) igual a 6.18 V, y cuando percibe luz el VAR es igual a 0 V. Estos cambios de voltaje de 0 a 6.18 V se presentan en la salida (S) de FT2.

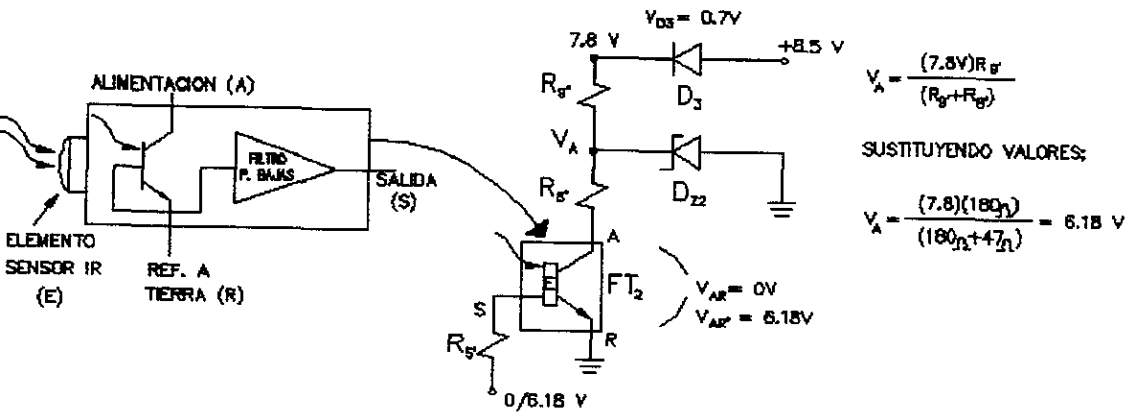


Fig. 3.11. Funcionamiento doble del fototransistor FT2.

Para desempeñar la segunda tarea, FT<sub>2</sub> funciona como un filtro pasa bajas, presentando en la salida de éste los datos que le fueron enviados por el dispositivo remoto. Enseguida estos datos entran a una etapa de amplificación mediante el transistor T<sub>6</sub>, el cual es un amplificador emisor común. La configuración de emisor común permite amplificar la corriente y el voltaje. En esta configuración la base de T<sub>6</sub> es la entrada, el emisor es la referencia a tierra por medio del capacitor C<sub>3</sub>' y el colector es la salida. Ver la figura 3.12.

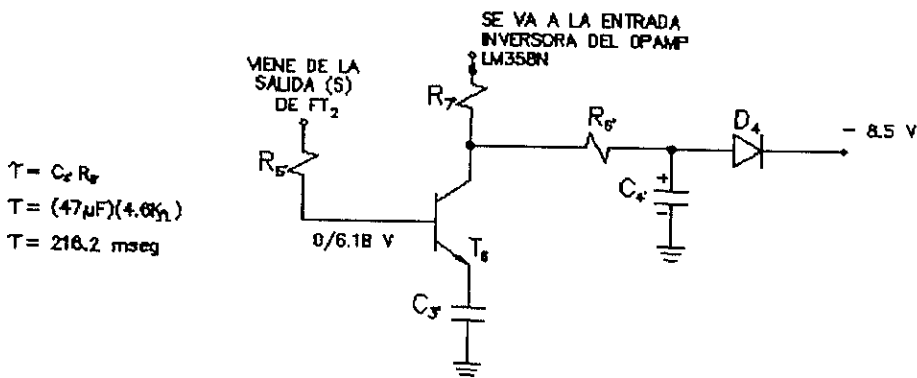


Fig. 3.12. Configuración de emisor común.

Como se puede observar en la figura anterior, el diodo D<sub>4</sub> permite solamente la polarización negativa de -8.5 V provenientes del pin 6 del CI MAX232, C<sub>4</sub>' es un capacitor electrolítico que corrige junto con la resistencia R<sub>6</sub>' el retardo originado por el amplificador T<sub>6</sub>, y finalmente, R<sub>7</sub>' ajusta el voltaje de salida de T<sub>6</sub> hacia la entrada inversora del amplificador operacional LM358N.

Cuando se presentan los cambios de estado de la señal moduladora en la base del transistor T<sub>6</sub>, provenientes de la resistencia R<sub>5</sub>', ver la figura 3.13. Estos cambios de estado producen que T<sub>6</sub> opere en corte o en saturación, por ejemplo, si se presenta un uno lógico en la base de T<sub>6</sub>, éste opera en la región de corte puesto que el colector presenta un voltaje negativo de -8.5 V y por consecuencia el V<sub>CE</sub> es igual a -8.5 V, cuando se presenta un cero lógico T<sub>6</sub> pasa a la región de saturación ocasionando que V<sub>CE</sub> caiga a cero Volts. Estos nuevos niveles de voltaje que adquiere la señal de información deben ser ajustados a niveles lógicos TTL, mediante una etapa de detección de nivel. Esta etapa emplea un detector de nivel de voltaje inversor (OPAMP LM358N), configurado para trabajar en +V<sub>sat</sub> y -V<sub>sat</sub>. Cabe señalar que este amplificador operacional además de ajustar

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

los niveles de voltaje, invertirá la señal proveniente de  $T_6$  ocasionando que la señal presente su forma "casi" original, puesto que como sabemos  $T_6$  aparte de ayudar a su amplificación, paso a invertir la señal de información.

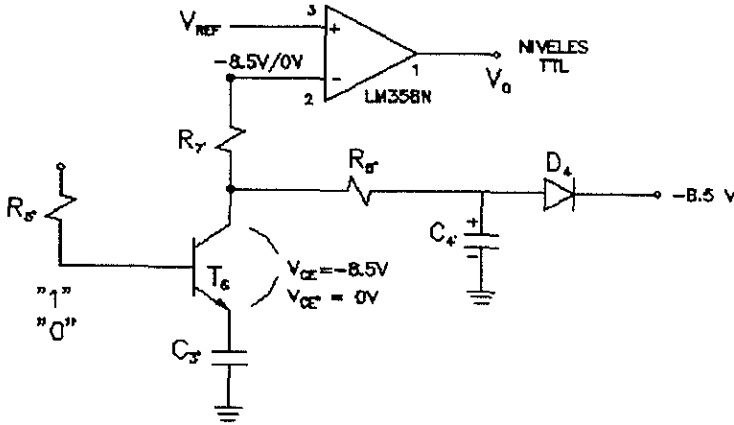


Fig. 3.13. Operación del transistor  $T_6$ .

Con los nuevos niveles de voltaje presentes después de la resistencia  $R_7$ , es decir en el pin 2 del amplificador operacional (OPAMP) LM358N, y un voltaje de referencia positivo ( $V_{REF}$ ) de 8.5 V en el pin 3, se obtiene a la salida del OPAMP, pin 1, un voltaje  $V_o$  igual a cambios de nivel de voltaje netamente TTL, ver la figura 3.13. Cabe mencionar que el voltaje de referencia ( $V_{REF}$ ) proviene del pin 2 del CIMAX232. En la figura 3.14 se muestra la polarización del OPAMP, la cual consiste de +5 V para +Vcc y 0 V para GND. Ahora apoyándonos en esta misma figura tenemos lo siguiente; Cuando se tiene un voltaje de referencia ( $V_{REF}$ ) de 8.5 V en la entrada no inversora, y un voltaje  $V_i$  de -8.5 V en la entrada inversora, se recurre a que si  $V_i$  es menor al voltaje de referencia ( $V_{REF}$ ), se tiene en la salida ( $V_o$ ) un voltaje + $V_{sat}$  igual a 5 V o un "1". Ahora, cuando se tiene el mismo voltaje de referencia ( $V_{REF}$ ) pero el voltaje  $V_i$  cambia a cero Volts, se recurre a que si  $V_i$  es mayor al voltaje de referencia ( $V_{REF}$ ), se tiene en la salida ( $V_o$ ) un voltaje - $V_{sat}$  igual a 0 V o un "0". Pero como nos podemos dar cuenta nunca  $V_i$  es mayor al voltaje de referencia, al contrario se presenta menos negativo, aun así nos proporciona un voltaje - $V_{sat}$  igual a cero, puesto que la polarización del OPAMP esta de +Vcc = 5 V a GND = 0 V, lo cual ocasiona que - $V_{sat}$  no pueda ir más halla de cero volts, o lo que es lo mismo + $V_{sat}$  no puede ir más halla de 5 V puesto que esta alimentado a 5 V y no a 9 ó a 12 ó a 24 Volts. Ver la figura 3.14.

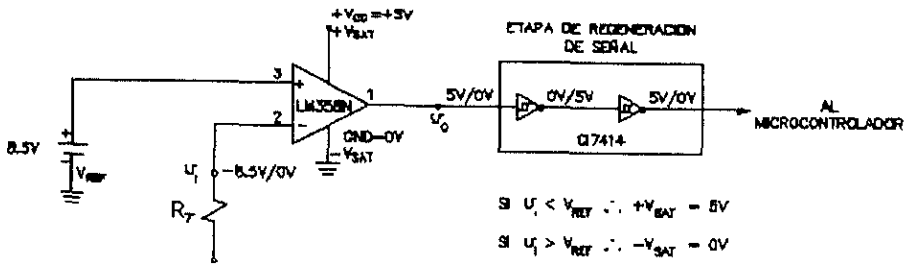


Fig. 3.14. Detector de nivel de voltaje inversor (LM358N).

Como ya tenemos la señal moduladora con niveles lógicos TTL (0/5V) y los datos o bits que la componen son casi iguales a los transmitidos, ahora éstos pasarán a la última etapa que consiste en regenerar la señal para que esta se parezca aun más a la señal original. Esta etapa emplea, o mejor dicho, esta formada por dos inversores de histéresis Schmitt-Trigger contenidos en el CI 7414. Estos inversores tienen la tarea de regenerar la señal, pero la pueden invertir, debido a esto utilizamos dos, por ejemplo, el primer inversor regenera la señal totalmente pero además la invierte, al pasar esta señal por el segundo inversor, también regenera la señal invirtiéndola, regresándola a su estado original. En general, estos dos inversores funcionan como si solamente hubieran regenerado los pulsos de la señal, sin modificar su fase.

Finalmente, ya en la salida de esta última etapa se logra obtener una señal de recepción casi idéntica a la señal que fue transmitida, aunque todavía existe un proceso opcional que solamente se emplea cuando se requiere que estos datos recibidos se introduzcan hacia un computador. Este proceso opcional consta de una etapa de conversión de niveles lógicos, es decir, convertir los niveles TTL de la señal recibida a niveles RS-232, para que la computadora remota pueda entender los datos que esta recibiendo. Para lograr lo anterior utilizamos el CI MAX232 como se muestra en la figura 3.7.

La forma de utilizar esta etapa opcional es colocando el conector JP1 entre el pin 4 del CI 7414 y el pin 11 del CI MAX232 para que la señal recibida con niveles lógicos TTL pase a ser convertida por el MAX232 a niveles RS-232 y pueda ser entendida por cualquier dispositivo remoto que maneje la norma RS-232, es decir, esta etapa opcional permite enlazar dos computadores remotos o dos dispositivos que manejen este tipo de protocolo de nivel físico. Para el caso de no ser necesario utilizar esta última etapa, el conector JP1 se coloca entre el pin 4 del CI 7414 y el pin 42 (Rx/D)

del microcontrolador MC68HC811E2, puesto que este maneja niveles TTL, sin necesidad de utilizar etapas de conversión de nivel extra.

Hasta este momento se ha logrado transmitir una señal desde un computador, esta señal ha utilizado el espacio libre para propagarse y ser recibida por un microcontrolador, teniendo la opción de ser recibida en su lugar por otro computador. Este proceso solamente utiliza una vía de comunicación. Para que en este caso el microcontrolador (o el computador, según el enlace que se desee realizar) pueda enviar una señal, que pueda ser un mensaje o una respuesta, al computador que se encuentra en el otro extremo, se requiere utilizar una segunda vía de comunicación. Cabe señalar que la primer vía de comunicación contiene al TxIR<sub>1</sub> y al RxIR<sub>2</sub> a una frecuencia de 40 KHz y la segunda vía de comunicación contiene al TxIR<sub>2</sub> y al RxIR<sub>1</sub> también a 40 KHz, como se muestra en la siguiente figura.

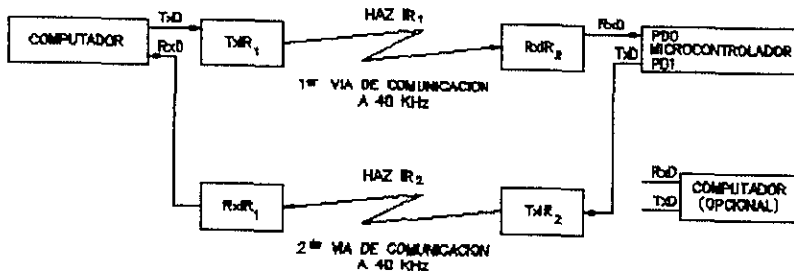


Fig. 3.15. Comunicación bidireccional infrarroja completa (Dos vías de comunicación).

Como se puede observar contamos con un sistema de comunicación bidireccional, logrando que el computador y el microcontrolador se comuniquen en ambas direcciones, es decir, en la primer vía el computador habla y el microcontrolador escucha, y en la segunda vía el microcontrolador habla y el computador escucha.

Pero estas dos vías de comunicación operan a la misma frecuencia, dando pie a ocasionar entre ellas interferencia, lo cual nunca ocurre debido a que es una comunicación bidireccional en modo Half-Dúplex, ver el capítulo II. Este modo de operación permite utilizar la misma frecuencia para transmitir y para recibir, solo que los tiempos de transmisión son distintos, es decir, mientras transmite la primer vía, la segunda no transmite y viceversa, pero nunca las dos vías al mismo tiempo.

La manera de utilizar la segunda vía de comunicación, apoyándonos del modo Half-Dúplex, es cuando el microcontrolador requiere enviar un mensaje al computador que se encuentra en el otro extremo. Para lograr lo anterior, en la figura 3.7, el pin 43 del microcontrolador se debe de conectar con el pin 1 del CI 7408 por medio del conector JP2 (si se necesita que en lugar del microcontrolador, transmita el computador, entonces el conector JP2 se debe de colocar entre el pin 12 del CI MAX232 y el pin 1 del CI 7408), con esto la información que desee enviar el microcontrolador será multiplicada con la frecuencia portadora de 40 KHz que proporciona el CI 555. La multiplicación la realiza la compuerta AND.

A la salida de la compuerta AND, pin 3, se tiene una modulación en frecuencia que es la multiplicación de la señal portadora que viene del CI 555 y la señal moduladora que viene del microcontrolador o del computador, según sea el caso. A continuación esta señal modulada se introduce a una etapa de amplificación por medio del arreglo Darlington formado por T3 y T4. Este arreglo presenta en la base del transistor T3 un divisor de voltaje mediante R1' y R2'. Posteriormente los cambios de señal amplificados polarizan directamente e inversamente al IRED2 ocasionando que esos cambios lo enciendan y lo apaguen produciendo radiación infrarroja. Es muy importante señalar que este transmisor (TxIR2) funciona exactamente igual que TxIR1, ocurre lo mismo para el receptor que se encuentra del lado del computador (RxIR1), el cual funciona igual que RxIR2, es por este motivo que no se profundiza en su explicación.

Ya teniendo la señal modulada convertida en radiación IR, esta se propaga por el espacio libre hasta ser detectada por el fototransistor FT1. Cuando FT1 detecta los cambios que fueron transmitidos por el IRED2, los filtra dejando solamente la señal moduladora, es decir, los datos que envió el microcontrolador.

Con la señal moduladora sin la componente de alta frecuencia, se introduce a una etapa de amplificación mediante el transistor T5 configurado como colector común, proporcionando a la salida una baja impedancia así como los datos amplificados y listos para entrar al detector de nivel LM358N, el cual tiene la función de que la señal recibida tome valores de voltaje entre 5 y 0 Volts, y además la invierte, dejándola en su fase original puesto que el transistor T5 además de amplificarla, la invirtió.

A la salida del amplificador operacional LM358N se tiene la señal de información con niveles lógicos TTL, ahora esta señal pasa a ser regenerada por medio de dos inversores contenidos en el CI 7414, los cuales ayudan a que la señal recibida sea casi idéntica a la señal transmitida. Posteriormente, esta señal pasa por una etapa de conversión de nivel, aquí los niveles TTL de la señal recibida se cambian por niveles RS-232 (esto lo realiza, como ya se explicó anteriormente, el CI MAX232), puesto que el computador al momento de recibir los datos, estos los entiende con niveles de +/- 12 Volts y no con niveles de 0 y 5 Volts. Cabe señalar o mejor dicho recordar que la conexión física entre este receptor y el computador es por medio de un conector DB-9 configurado en módem nulo.

Para complementar la explicación sobre el funcionamiento del sistema inalámbrico bidireccional IR, a continuación se presenta la lista de los componentes que se emplearon para el Transmisor-Receptor IR (Fig. 3.6) y para el Receptor-Transmisor IR (Fig. 3.7). Estos componentes son:

**PASIVOS**

$C_M = C_A = C_X = C_I = 1 \mu F \text{ 25 V}$

$C_1 = C_1' = 10 \text{ nF (Cerámico)}$

$C_2 = C_2' = 100 \mu F \text{ 25 V}$

$C_3 = C_3' = 100 \text{ nF (Cerámico)}$

$C_4 = C_4' = 47 \mu F \text{ 25 V}$

$C_V = 0.1 \mu F \text{ (Cerámico)}$

$C_{EX} = C_{XT} = 22 \text{ pF (Cerámico)}$

$C_R = 1 \mu F \text{ (Cerámico)}$

$R_1 = R_1' = 2.2 \text{ K}\Omega$

$R_2 = R_2' = 10 \text{ K}\Omega$

$R_3 = R_3' = 3\Omega$

$R_4 = R_4' = 22\Omega$

$R_5 = R_5' = 9.9 \text{ K}\Omega$

$R_6 = R_6' = 4.6 \text{ K}\Omega$

$R_7 = R_7' = 80\Omega$

$R_8 = R_8' = 47\Omega$

$R_9 = R_9' = 180\Omega$

$R_A = R_A' = 2 \text{ K}\Omega \text{ (Potenciometro)}$

$R_B = R_B' = 1 \text{ K}\Omega$

$R_E = R_E' = 25 \Omega$

$R_X = 10 \text{ M}\Omega$

### SEMICONDUCTORES

X<sub>1</sub> = 8 MHz

T<sub>1</sub> = T<sub>2</sub> = T<sub>3</sub> = T<sub>4</sub> = TBJ BC547

IRED<sub>1</sub> = IRED<sub>2</sub> = ECG 3027

Compuerta AND 7408

Circuito MAX232 (MAXIM)

Circuito temporizador NE555

FT<sub>1</sub> = FT<sub>2</sub> = F533

T<sub>5</sub> = T<sub>6</sub> = SRTA 933

D<sub>1</sub> = D<sub>2</sub> = D<sub>3</sub> = D<sub>4</sub> = 2N4006

Dz<sub>1</sub> = Dz<sub>2</sub> a 6 V

Amplificador Operacional (OPAMP) LM358N

Inversor de histéresis 7414 (SCHMITT-TRIGGER)

Microcontrolador MC68HC811E2 (MCU)

### VARIOS

Conectores DB-9 con concha y cable

Conectores (JP<sub>1</sub>, JP<sub>2</sub> y JP<sub>3</sub>)

Push Botton (Pb<sub>1</sub>)

Tablillas Master Circuit

Fuentes de poder (a 5 V)

Estructuras de aluminio

Soldadura

Ya vimos como trabajan los componentes discretos tanto del Transmisor-Receptor IR como del Receptor-Transmisor IR para lograr establecer una comunicación bidireccional, ahora veremos este mismo funcionamiento pero a nivel de señales.

Para poder entender mejor el funcionamiento de este sistema de comunicación, nos apoyaremos del siguiente ejemplo. Este ejemplo consiste en enviar por medio del computador la letra "R" hacia el microcontrolador, al momento que éste reciba el mensaje inmediatamente enviara una serie de letras como "S" "X" "S" "W" "S" "V" hasta la última letra "S" (esta transmisión tiene una duración de 9 minutos). Estas letras las recibirá el computador mostrándolas en pantalla, ver la figura 3.42 de la pagina 105. El desarrollo del presente ejemplo se muestra a continuación:

Considerando que el sistema se encuentra funcionando al 100% y que el enlace IR establecido es entre un computador y un microcontrolador, se procede a enviar el comando "R". Ver la figura 3.16.

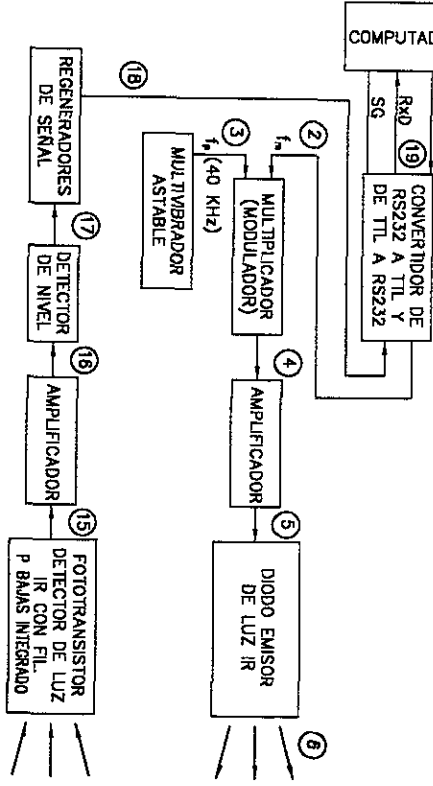
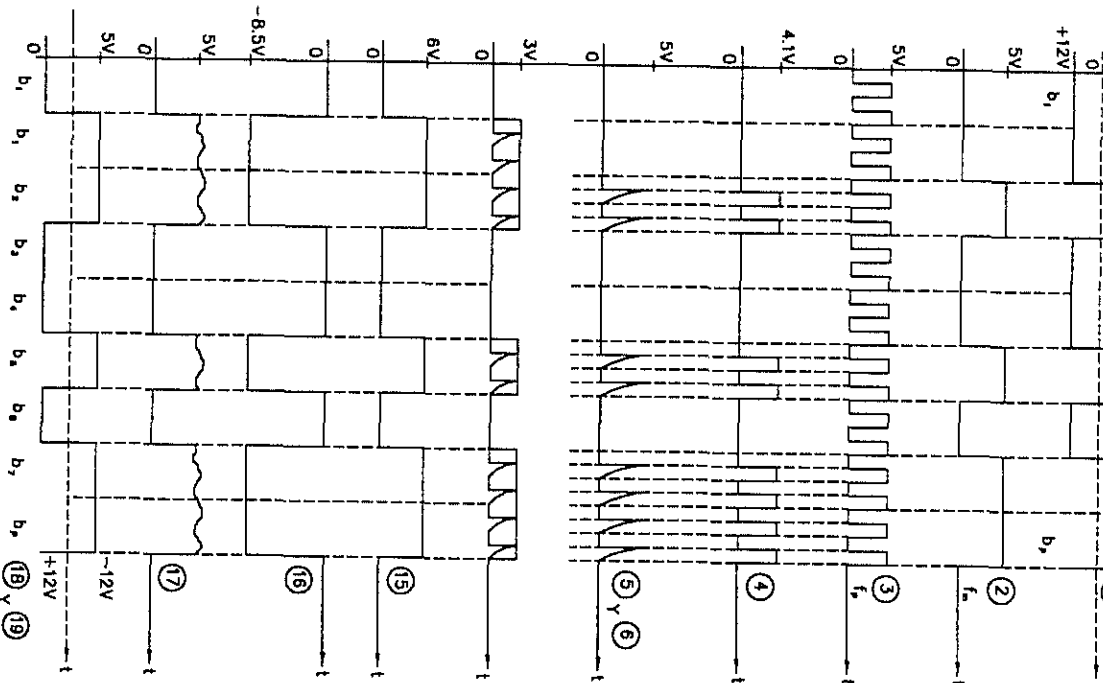


Esta letra "R" es proporcionada por el computador en código ASCII, es decir, 0100101 con niveles de voltaje de +/- 12 V y además presentando un formato de datos en transmisión asíncrona, esto es, con un bit de inicio y con un bit de paro (0 0100101 1). En total el computador nos entrega 9 bits, de los cuales 7 bits corresponden a la letra "R", según el código ASCII, y los dos restantes corresponden al inicio y al fin de la trama de datos, a esto se le conoce como transmisión asíncrona. Ver el punto 1 de la figura 3.16. Al entrar estos bits o datos al CI MAX232 (en este caso convertidor RS-232 a TTL), solamente cambian sus niveles de voltaje a 0 y 5 V como se observa en la gráfica del punto 2.

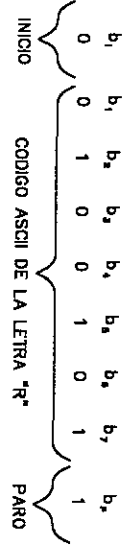
El punto 3 presenta la forma de la señal portadora generada por el CI 555 (Multivibrador Astable), la cual entra al multiplicador (CI 7408) junto con la señal moduladora (fm) proveniente del computador con niveles de voltaje TTL.

A la salida del multiplicador, se obtiene una modulación en frecuencia de las dos señales, esto es, la portadora (fp) fue multiplicada con respecto a los cambios de la moduladora (fm) como se observa en el punto 4. Aquí los 9 bits que forman la letra "R", solo los de estado alto, presentan en su interior trenes de pulsos a 40 KHz.

En el punto 5 se puede observar esta misma señal pero amplificada al igual que en el punto 6, solo que aquí esta señal va en forma de fotones o luz. Cabe señalar que este circuito transmisor mantiene el enlace con el receptor por medio de la portadora aunque no exista transmisión de información, ya que opera en onda continua (CW). También hay que mencionar, que mientras se está realizando la transmisión (envío de la letra "R"), el receptor (RxIR1) está enlazado con el transmisor del otro extremo (TxIR2) mediante su portadora (2da vía de comunicación).



LA TRANSMISION ASINCRONA ES:



LA RECEPCION ASINCRONA ES:

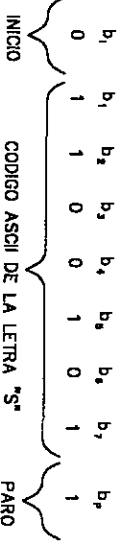


Fig. 3.16 Analisis de señal del transmisor-receptor IR

En este momento la letra "R" viaja por el espacio libre, propagándose constantemente hasta que es detectada por el receptor del microcontrolador. Al ser detectada esta señal inmediatamente se retira la portadora, quedando solamente la envolvente, es decir los datos o bits de la señal. Lo anterior se muestra en el punto 7 de la figura 3.17.

En el punto 8 de esta misma figura se obtienen los bits recibidos pero *amplificados*, presentando niveles de voltaje de 0 y -8.5 V, los cuales al salir del detector de nivel presentan niveles de 0 y 5 V como se observa en el punto 9. Hasta aquí ya se tiene el código ASCII de la letra "R" junto con su bit de inicio y su bit de paro.

En el punto 10, ya se le ve con más forma a la señal recibida siendo una copia (exacta) aproximada de la señal que se envió. En este momento los 9 bits recibidos se introducen al receptor de datos (PD0) del puerto SCI del microcontrolador, el cual se encargará por medio del programa de recepción de datos (programa PROTOCON, ver el punto 3.4) que contiene internamente, de detectar el bit de inicio y el bit de paro para posteriormente procesar sólo los 7 bits que conforman la letra "R".

Cuando el microcontrolador reconoce o compara a la letra "R" que recibió con la letra "R" que tiene en memoria, se arranca una subrutina que se encarga de *transmitir durante 9 minutos ininterrumpidamente* una serie de letras como "S", "X", "S", "W", "S", "V", etc. Desde este momento el microcontrolador transmite estas letras en código ASCII y con sus respectivos bits de inicio y de paro. En el caso que el microcontrolador no reconozca la "R" que se le envió, debido a cualquier clase de error, este no transmitirá.

Inicialmente el microcontrolador envía primero la letra "S", es decir, 0 1100101 1, como se muestra en el punto 11. Esta señal se introduce al multiplicador donde se modula de acuerdo a la frecuencia portadora que proporciona el *multivibrador astable*, ver los puntos 12 y 13. Posteriormente esta señal ya modulada, se amplifica (punto 14) y se convierte a luz IR por medio del diodo emisor infrarrojo.

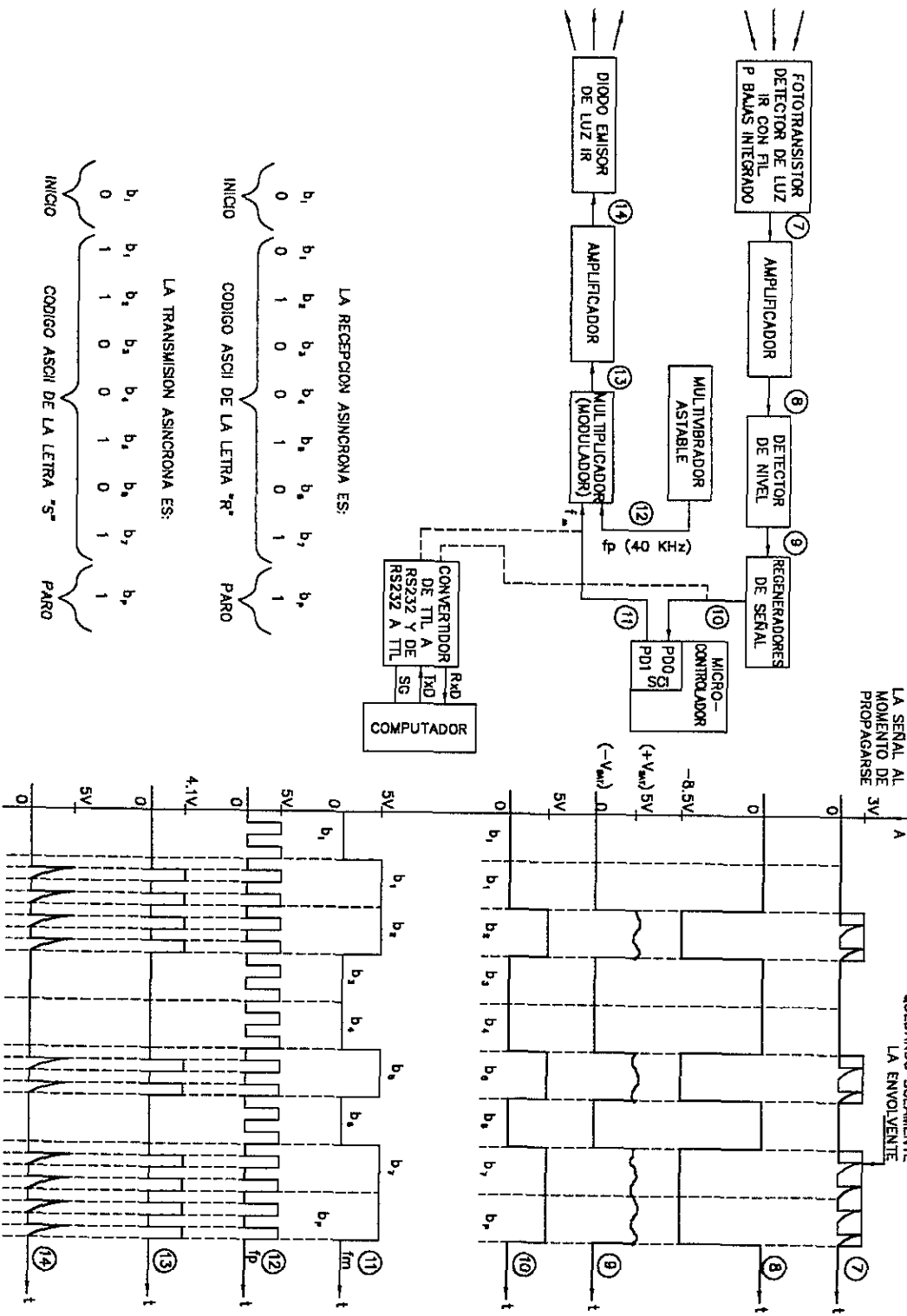


Fig. 3.17 Analisis de señal del receptor-transmisor IR

Conforme los cambios de luz IR son detectados en el otro extremo, se retira la señal portadora quedando solamente los bits de la letra "S" junto con los bits de inicio y paro, ver el punto 15 de la figura 3.16. Enseguida, como se puede observar en el punto 16, estos bits se amplifican adquiriendo niveles de voltaje un tanto inusuales. Estos nuevos niveles de voltaje que presenta la señal son ajustados mediante el detector de nivel, obteniendo una señal muy semejante a la original, lo anterior se muestra en el punto 17.

Teniendo los 9 bits que conforman a la letra "S" casi idénticos a los transmitidos, solamente falta regenerar la señal completamente para asegurar su forma casi idéntica, ver el punto 18.

Al recibir en total 9 bits completamente regenerados, solo falta convertir sus niveles de voltaje de TTL a RS232, es decir, de 0 y 5 V a +/- 12 V, para que el computador los pueda entender, ver el punto 19.

El programa de comunicación con que cuenta el computador, detecta primero el bit de inicio, posteriormente cuenta 7 bits y después detecta el bit de paro. Este programa permite desplegar el código ASCII de la letra correspondiente en pantalla, es decir, despliega o muestra en pantalla la letra "S" que nos envió el microcontrolador sin ningún error. Cuando se termina este proceso, inmediatamente el programa despliega la siguiente letra que fue recibida, en este caso es la "X", hasta terminar de recibir todas las letras que le fueron enviadas durante 9 minutos.

En este momento el que transmite es el microcontrolador y el que recibe es el computador, pero si este le envía un comando de paro (letra "P") al microcontrolador, este dejara de transmitir ocasionando que en el monitor del computador se vea interrumpida la recepción, observando solamente el parpadeo del cursor. Si en caso contrario se le envía una letra diferente a "P", el microcontrolador no responderá y seguirá con la transmisión.

Cuando el microcontrolador termina de transmitir, se mantiene vivo el enlace, es decir, el TxIR<sub>1</sub> sigue enlazado por portadora

con el RxIR2 y el TxIR2 (el cual ya termino de transmitir) sigue enlazado por portadora con el RxIR1.

Finalmente, como hemos visto a lo largo de esta explicación mencionamos como transmitir y recibir una señal de un lado a otro y también de manera inversa, explicamos el modo de operar de las dos vías de comunicación, además de la frecuencia de operación que utilizan para enlazarse, mencionamos los cálculos de diseño que se emplearon para desarrollar los circuitos que conforman el sistema inalámbrico IR. En general se explicó todo el hardware del sistema, aunque para el ejemplo anterior se mencionaron algunas cuestiones sobre el software que se utilizó, siendo el elemento fundamental que complementa el sistema, y además es buen momento para profundizar en el tema.

### 3.4. Software de comunicación

El software que se empleo para este sistema de comunicación, tanto para ponerlo en operación así como para realizar sus respectivas pruebas, consiste de dos partes. La primera parte es el software para el Transmisor-Receptor IR (computador), el cual es un programa terminal llamado **TTY**, la segunda parte es el software para el Receptor-Transmisor IR (microcontrolador), el cual es un programa desarrollado en lenguaje ensamblador llamado **PROTOCON**. Cabe mencionar que se desarrollaron otros dos programas en lenguaje ensamblador llamados **PROTCOR** Y **PROTCOT** muy similares pero con distinta forma de detectar errores. Lo anterior se explicará más adelante, por ahora nos enfocaremos en primera instancia al programa terminal TTY. Hay que tomar en cuenta que en el computador se carga el programa TTY, y en el microcontrolador se carga el programa PROTOCON, si se requiere establecer un enlace computador-computador, estos dos dispositivos deberán tener cargado el TTY, pero si se necesita establecer un enlace computador-brazo robot, en este caso el controlador del brazo robot deberá tener cargado un programa en lenguaje C.

El programa terminal TTY se instala en cualquier computadora personal desde Windows 3.1 hasta Windows 95, este programa permite configurar los puertos de comunicación disponibles en el computador, la velocidad de transmisión, el tipo de formato de datos, el control de flujo, etc. A continuación se muestra la presentación de este programa terminal de comunicaciones.

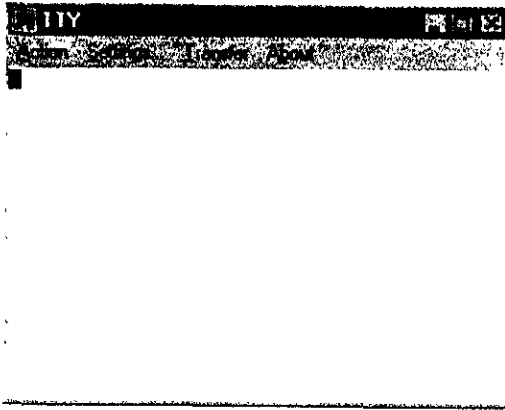


Fig. 3.18. Programa terminal TTY listo para transmitir o recibir información.

Cuando se ejecuta este programa (desde el administrador de archivos de Windows se hace un doble click a tty.exe), aparece su ventana de presentación como se observa en la figura anterior. Este programa cuenta con cuatro letreros, los cuales desempeñan las siguientes funciones:

Action.- Este menú permite conectar o desconectar el enlace de comunicación, así como salirse del programa, ver la figura 3.19. Para cerrar este programa, se puede hacer uso de los botones que se encuentran en el ángulo superior derecho.

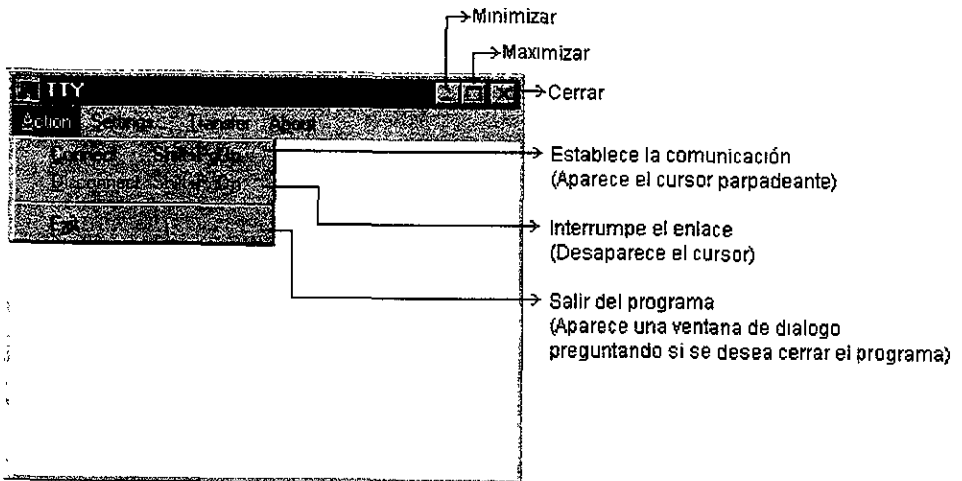


Fig. 3.19. Funciones del menú Action.

Settings.- Al hacer un click en este letrero, se despliega una ventana de configuración, llamada TTY Settings..., ver la figura 3.20, donde se muestran los siguientes parámetros de comunicaciones (caja Com Options):

Port : Aquí se selecciona el puerto disponible en el computador, el cual puede ser COM1, o COM2, o hasta el COM9, todo esto dependiendo de que dirección del puerto este configurada o habilitada desde windows. Este parámetro toma como valor por default COM1.

Baud Rate: Esta opción nos permite seleccionar la velocidad de transmisión a la cual queremos transmitir la información, que puede ser desde 110 bps hasta 256 Kbps. El valor por default que toma es de 9600 bps.

Data Bits: Con esta opción podemos elegir el tamaño de bits de datos que se deseen utilizar, como pueden ser 8 bits o menos (7, 6 y 5). Este número de bits de datos se elige de acuerdo al código de la información, como en este caso se utiliza el código ASCII, este emplea solo 7 bits. El valor que toma por default es 8.

Parity: Aquí nosotros podemos configurar un tipo de chequeo de errores de acuerdo al tipo de paridad que usemos como puede ser par, impar, marca, espacio, o ninguna. El valor por default que toma esta opción es ninguna (None).

Stop Bits: Los bits de paro que podemos elegir en esta opción son 1, 1.5 ó 2, los cuales indicarán el termino de la trama de datos que se transmite. Como hemos visto la transmisión asíncrona necesita de un bit de inicio, de 7 u 8 bits de datos, y de un bit de paro, de los cuales los dos últimos se seleccionan. El bit de inicio no se selecciona debido a que el programa TTY al momento de transmitir envía un cero lógico por default, indicando al otro extremo que se inicia la comunicación. Esta opción toma el valor de 1.

Flow: Aquí se permite habilitar el tipo de control de flujo que se requiera para realizar el enlace, estos pueden ser: Data Terminal Ready/Data Set Ready (DTR/DSR), Request To Send/Clear To Send (RTS/CTS), y XON/XOFF. Para una explicación más amplia sobre estos tipos de control de flujo, se puede consultar el



punto 2.3 del capítulo II. El tipo de control de flujo que se habilita por default es RTS/CTS.

Como podemos ver en la figura 3.20, la ventana TTY Settings... contiene tres cajas. La primera es COM Options, la segunda es ITY Options, y la última es Control Options, también contiene tres botones, los cuales son: Font, Ok y Cancel. Anteriormente explicamos la función de la caja COM Options, ahora se procede a explicar la caja ITY Options. En esta caja contamos con tres opciones, Autowrap, New Line y Local Echo. La opción Autowrap se puede habilitar o deshabilitar, si esta es habilitada no muestra en pantalla los datos que se reciben, ni los que se transmiten sin perturbar el enlace, solo es una opción de visualización. Cabe mencionar que estas tres opciones son solo de visualización.

La segunda opción, New Line, se puede habilitar o deshabilitar. Si esta opción es habilitada al momento de transmitir una línea completa, en pantalla no se ocupa la siguiente línea o renglón, sino que se deja un espacio entre renglones. Para recibir es la misma cuestión, es decir recibimos varias líneas de información consecutivas pero espaciadas una de otra.

La tercer y última opción de la caja ITY Options, Local Echo, permite visualizar en pantalla los datos que serán transmitidos, por ejemplo; si esta opción no se habilita y queremos establecer el enlace al enviar una "R", esta letra o carácter no lo veremos en pantalla pero si veremos los datos recibidos.

La caja Control Options, permite seleccionar si deseamos que el programa terminal TTY nos proporcione notificaciones en la recepción y además despliegue en pantalla esos errores en caso de que existan. Estas opciones la mayoría de las veces se emplean ambas al mismo tiempo, puesto que si se reciben notificaciones las podemos ver como <CE-24> ó <CE-9> dependiendo del tipo de error, si al contrario no tenemos habilitada la opción Display Errors, no podemos observar en pantalla esas notificaciones.

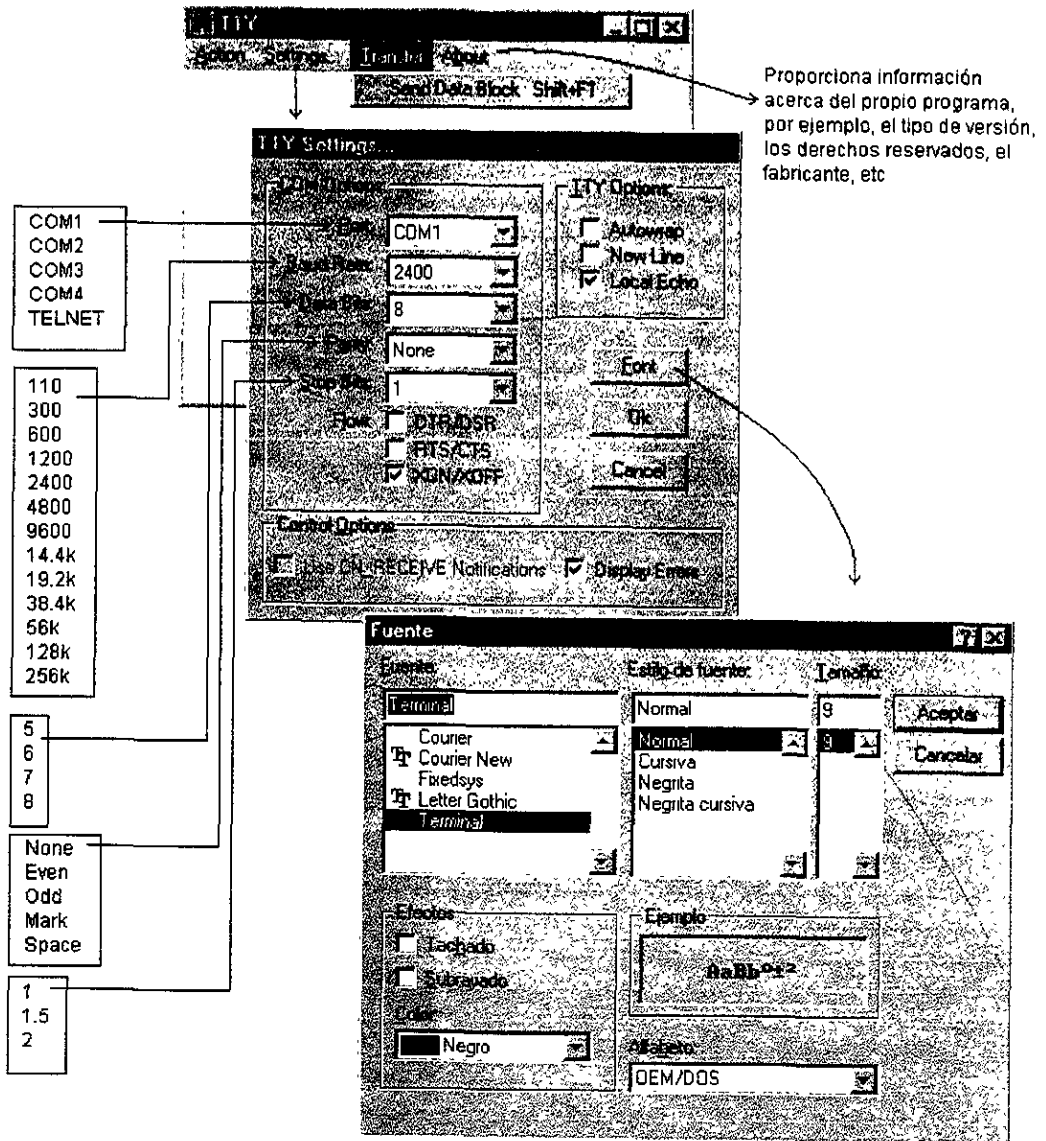


Fig. 3.20. Funciones del menú Settings, del submenú Font y de los letteros Transfer y About.

Si se requiere ocupar algún tipo de letra diferente al que proporciona el programa TTY, podemos hacerlo por medio del botón Font. Al hacer un click en este botón, se despliega otra ventana que se llama Fuente, como

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

se puede ver en la parte inferior de la figura 3.20. En esta ventana podemos elegir diferentes tipos de fuentes como terminal, Letter Gothic, Fixedsys, Courier, etc. También podemos elegir el estilo, por ejemplo; Normal, Cursiva, Negrita, así como el tamaño de la letra, además se le pueden agregar efectos y diferentes tipos de colores a las letras. Si se realizan cambios en esta ventana se guardan con el botón Aceptar, y si no es necesario guardar las modificaciones se cancelan con el botón Cancelar. Sucede lo mismo para la ventana TTY Settings..., si se realizaron cambios en los parámetros de comunicación y de visualización se guardan con el botón Ok, si no son importantes esos cambios se cancelan con el botón Cancel.

Ya para terminar con las funciones del programa terminal TTY, nos falta por mencionar los letreros Transfer y About, los cuales se presentan a continuación:

Transfer.- Permite enviar un bloque de datos predeterminado hacia el otro extremo, el cual no se puede visualizar en pantalla, pero si ocurre algún error en la transmisión de este bloque de datos, se despliega en pantalla la notificación <CE-6552636>.

About.- Este letrero presenta una ventana que proporciona información acerca o sobre este programa terminal como puede ser el tipo de versión, el nombre del fabricante, el registro de derechos reservados, etc.

Hasta ahora ya conocemos las funciones del programa TTY, siendo este el momento para dar inicio a su configuración, la cual consiste de los siguientes pasos:

1.- En principio, consideramos que acabamos de acceder al programa terminal TTY, ver nuevamente la figura 3.18. Enseguida damos un click al letrero Settings..., para que aparezca la ventana TTY Settings donde realizaremos las siguientes modificaciones:

### COM Options:

Port = COM1  
Baud Rate = 2,400 bps  
Data Bits = 7  
Parity = Ninguna  
Stop Bits = 1  
Flow = XON/XOF

### TTY Options:

✓ Local Echo

### Control Options:

✓ Usar Notificaciones  
✓ Desplegar Errores

2.- Ya teniendo realizadas las modificaciones, damos un click en el botón Ok. Si se requiere cambiar el tipo de letra o algún otro

parámetro, necesitamos acceder nuevamente a la ventana TTY Settings, puesto que al momento de oprimir el botón Ok se cerró. Teniendo otra vez abierta la ventana TTY Settings, oprimimos el botón Font para que enseguida aparezca su respectiva ventana, e introducir lo siguiente:

Fuente:

Fuente = Terminal
Estilo de fuente = Normal
Tamaño = 9
Color = Negro

Para guardar esta nueva configuración damos un click en Aceptar. Posteriormente regresamos a la ventana TTY Settings en donde damos un click en Ok puesto que ya tenemos los parámetros guardados y no realizaremos ninguna modificación. En el caso de que se necesite modificar algún parámetro, antes de oprimir el botón Ok realizamos los ajustes necesarios y ahora si damos un click en Ok.

3.- Hasta este momento ya tenemos configurados todos los parámetros de comunicación, sin ningún otra ventana abierta, solamente la ventana principal del TTY. Posteriormente damos un click en el letrero Action, y enseguida damos un click en Connect para establecer la comunicación, inmediatamente el cursor comienza a parpadear indicando que esta listo y en espera tanto para transmitir como para recibir.

Con las condiciones anteriores podemos operar el sistema para que envíe la letra "R" y esperar las respuestas del microcontrolador, pero falta el programa PROTOCON que empleará el microcontrolador para "cuchar", por ejemplo, la letra "R" y enseguida transmitir "M", "8", "S", "X", "S", "W", "S", "V", etc.

El programa PROTOCON de comunicación, esta desarrollado, como ya mencionamos, en lenguaje ensamblador con comandos propios de la familia 68HC11 de motorola. Este programa habilita la transmisión y recepción de datos por medio de los registros del SCI (Serial Communications Interface), así como detectar errores de ruido. A continuación se presenta el listado del programa PROTOCON:

```
*****
**** Programa desarrollado en lenguaje ensamblador para el microcontrolador ****
**** MC68HC811E2 y para realizar las respectivas pruebas del enlace IR. ****
*****
```

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

\*\*\*\* Este programa transmite y recibe datos por el puerto serie durante 5 \*\*\*\*  
 \*\*\*\* minutos y posteriormente si recibe un comando (R) aumenta la transmisión \*\*\*\*  
 \*\*\*\* 9 minutos. Mediante otro comando (P) es posible parar la transmisión. \*\*\*\*  
 \*\*\*\* Si se requiere detectar errores por medio de otro comando (N) se \*\*\*\*  
 \*\*\*\* detectan errores de ruido en la línea. \*\*\*\*

***** Registros *****			⇒ SECCIÓN
REGBAS	EQU	\$1000	Inicio de direcciones (bloque de registros)
PORTB	EQU	\$04	Puerto B (despliegue de minutos)
TOC2	EQU	\$18	Registro Output Compare 2
TCTL1	EQU	\$20	
TMSK1	EQU	\$22	
TFLG1	EQU	\$23	
TMSK2	EQU	\$24	
TFLG2	EQU	\$25	
PACTL	EQU	\$26	Interrupción en tiempo real
BAUD	EQU	\$2B	***Registros para la comunicación serie***
SCCR1	EQU	\$2C	
SCCR2	EQU	\$2D	
SCSR	EQU	\$2E	
SCDR	EQU	\$2F	

***** Constantes *****			⇒ SECCIÓN
TDRE	EQU	\$80	Transmitter Data Register Empty
RDRF	EQU	\$20	Recive Data Register Full

***** Variables *****			⇒ SECCIÓN
MINS	EQU	\$00	Minutos
SEGS	EQU	\$01	Segundos
DECSGS	EQU	\$02	Décimas de segundo
NUMDISP	EQU	\$03	Contador de display a encender
TEMPO	EQU	\$04	# interrupciones para decremento de tiempo
PERIOD	EQU	\$05	(ORIGINAL 200000) 200ms con el xtal de 8Mhz
VARTEM	EQU	\$06	Variable temporal para lanzar el CRONOMETRO
TIME	EQU	\$09	Variable para el número de décimas de seg.
BASEDIG	EQU	\$0000	Base de los dígitos a desplegar

***** Main Program *****			⇒ SECCIÓN
ORG	\$F800		Dirección origen del programa
START	LDS	#\$0047	Empleo del stack
LDX	#REGBAS		Referencia del bloque de registros

***** Instalación de la rutina de servicio para el OC2 ****			⇒ RUTINA
LDAA	##%01000000		OM2:OL2 = 0:1
STAA	TCTL1,X		Pone el OC2 en comparación
STAA	TFLG1,X		Limpia el OC2F
STAA	TMSK1,X		Habilita la interrupción del OC2

***** Instalación de la rutina de servicio para el RTI ****			⇒ RUTINA
LDAA	##%01000000		RTI1(TMSK2) = 1
STAA	TMSK2,X		Habilita interrupciones RTI
STAA	TFLG2,X		Limpia el RTI
LDAA	##%00000011		Taza Nominal RTI = 32.77 ms
STAA	PACTL,X		

***** Inicialización de la comunicación serie *****			⇒ RUTINA
CLR	SCCR1,X		Inicializa el SCI a 1 bit de inicio, 8 bits

ESTA TESIS NO SALE DE LA BIBLIOTECA

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

```

LDD      #322C      de datos. 1 bit de paro y a 2400 baud
STAA     BAUD,X     Habilita la transmisión y recepción (TE-RE)
STAB     SCCR2,X    y la interrupción a la recepción (RIE=1)
***** inicialización ***** => RUTINA

LDAA     #$1A
STAA     PERIOD
LDAA     #$0A
STAA     TIME
LDAA     #5         Carga de MINS
STAA     MINS
CLRA
STAA     SEGS      Segundos a 00
STAA     DECSGS    Décimas de segundo a 00
LDAA     #3        ***** Primer display ... encendido *****
STAA     NUMDISP
STAA     VARTEM    ***** Inicialización del Crono *****
LDAA     #1
STAA     DECSGS    Primer descuento de MINS y SEGS
CLI
BRA      *         Habilita interrupciones
                    Interrupción driven
*****
**** Envío de un carácter cargado en el registro B por el puerto serie **
*****
WRITEC   BRCLR  SCSR,X #TDRE * ***** Esperar hasta que** => SUBROUTINA
STAB     SCDR,X ***** el puerto este libre**
RTS

***** Rutina de actualización del cronometro ***** => RUTINA

CRONO    DEC  DECSGS
         BEQ  REDUCIR
         RTS

REDUCIR  LDAA  $09
         STAA DECSGS
         LDD  MINS
         BNE CONTINU
         STOP

CONTINU  LDAB  #$F0
         ANDB SEGS
         LDAA #$0F
         ANDA SEGS
         BEQ  DECENSG
         DECA
         ABA
         STAA SEGS
BSR      SNDSEGS ***** Actualización de segundos *****
         RTS

DECENSG  TSTB
         BEQ  DECRMIN
         LDAA #9
         DECB
         ANDB #$F0
         ABA
         STAA SEGS
BSR      SNDSEGS ***** Actualización de segundos *****

```

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

```

RTS
DECRMIN  LDAA  %#01011001    (Carga SEGS con 59)
          STAA  SEGS
          DEC   MINS
          LDAB  #'M'          ***** Envío de minutos por el puerto *****
          BSR  WRITEC
          LDAB  MINS
          ADDB  #48
          BSR  WRITEC
          RTS
SNDSEGS  LDAB  #'S'          ***** Envío de segundos por el puerto *****
          BSR  WRITEC
          LDAB  SEGS
          BSR  WRITEC
AFFICHDEC NUMDISP
          BPL  POSITIV
          LDAA  #03          ***** Recomenzar el despliegue *****
          STAA  NUMDISP
POSITIVLDY #BASEDIG          ***** Base de los displays *****
          LDAB  NUMDISP
          LSRB
          ABY
          LDAA  $00,Y
          LDAB  NUMDISP
          LSRB
          BCS  NODESP          ***** Salto a no NODESPLAZAR *****
          LSLA
          LSLA
          LSLA
          LSLA
NODESP   ANDA  #$F0          *** Eliminación del nibble no necesario ***
          ORAA  NUMDISP
          STAA  PORTB,X
          RTS
*   *****
*   * SVRTI.- Rutina de servicio RTI => RUTINA
*   * Retoma a la interrupción RTI
*   *****
SVRTI   BCLR  TFLG2,X $BF    Limpia RTIF
          LDD  MINS          ***** Se acabo el tiempo???.? *****
          BEQ  NODESC
          DEC  VARTEM
          BNE  NODESC
          LDAA # $03
          STAA VARTEM
          BSR  CRONO          ***** Rutina del CRONOMETRO *****
          RTS                ***** Regresa al servicio RTI *****
*****
*   Solicitud de cambio de conteo a 9 minutos con "R" y *****
*   solicitud de paro de conteo con "P" desde el puerto serie **
*****
CHANGE  LDAA  #9            => SUBROUTINA
          STAA  MINS
    
```

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

```

CLRA
STAA SEGS
BRA EXITSCI
PARO CLR MINS          => SUBROUTINA
      CLR SEGS
      BRA EXITSCI

```

\*\*\*\*\*  
 \*\*\*\*\* Detección de error de ruido (NF) con "N" desde el puerto serie \*\*\*\*\*  
 \*\*\*\*\*

```

NEF   LDAA  #$04          => SUBROUTINA
      BITA  SCSR,X
      BEQ  LISUA
      *   LDAB #'0'
      *   STAB SCDR,X
      BRA  EXITSCI
LISUA LDAB  #'0'
      STAB SCDR,X
      BRA  EXITSCI

```

```

* *****
* * SVSCI.- Rutina de servicio SCI y de detección de errores => RUTINA
* * Retorna a la interrupción RTI
* *****

```

```

SVSCI LDAA  SCSR,X      Lectura para borrar la bandera RDRF
      LDAA  SCDR,X      Lee el dato en el acumulador A (ACCA)
      SEI                    Deshabilita interrupciones
      CMPA #'R            Reinicializar conteo
      BEQ  CHANGE
      CMPA #'P            Parar conteo
      BEQ  PARO
      CMPA #'N            Detección de error de ruido (NF)
      BEQ  NEF

```

```

EXITSCI CLI
RTI          ***** Regresa al servicio SCI *****

```

```

* *****
* * SV5OC2.- Rutina de servicio Output Compare 2 => RUTINA
* * Llama a la interrupción OC2
* *****

```

```

SV5OC2  LDD  PERIOD      Tiempo del delay para 1/2 ciclo
        ADDD TOC2,X      Adición al último valor compare
        STD  TOC2,X      Resultado de la adición en OC2
        BSR  AFFICH      ***** Rutina affich *****
        BCLR TFLG1,X $BF Limpia el OC2F
        RTI          ***** Retoma al servicio OC2 *****

```

\*\*\*\*\* Vectores de interrupción y Reset \*\*\*\*\* => SECCIÓN

```

ORG  $FFD6
FDB  SVSCI      Sistema serial SCI
ORG  $FFE6
FDB  SV5OC2     Timer Output Compare 2
ORG  $FFF0
FDB  SVRTI     Interrupción tiempo Real
ORG  $FFFE
FDB  START     Vector de reset
END

```



PROTOCON está formado por 5 secciones, las cuales son:

**Registros.-** En esta sección se dan de alta las direcciones de los registros a utilizar. Por ejemplo, el registro BAUD presenta la dirección en hexadecimal \$102B (en este registro se proporciona la velocidad de transmisión, ver el punto 2.5 del capítulo II). Los registros que se manejan en este programa se muestran con sus respectivas direcciones en el anexo A.

**Constantes.-** Se definen las direcciones de las constantes, en este caso TDRE es igual a \$1080 y RDRF es igual a \$1020. Estas constantes son banderas que indican si el registro transmisor de datos se encuentra vacío, y si el registro receptor de datos está lleno. Estas banderas, como se mencionó en el punto 2.5 del capítulo II, están contenidas en el registro SCCR2.

**Variables.-** Aquí, se definen las variables minutos, segundos, décimas de segundo, etc. Es decir, se definen las variables de tiempo para 5 y 9 minutos de forma decreciente. Las direcciones de estas variables van desde la \$1000 hasta la \$0000.

**Main Program.-** Inicialmente se presenta el origen en memoria del programa, el empleo del stack pointer y la referencia de direcciones de todos los registros empleados. Esta sección se compone de las siguientes rutinas: Instalación de la rutina de servicio para el OC2, Instalación de la rutina de servicio para el RTI, Inicialización de la comunicación serie, Inicialización, Rutina de actualización del cronometro, Rutina de servicio RTI, Rutina de servicio SCI y de detección de errores, y por último la Rutina de servicio Output Compare 2. A continuación se explica el funcionamiento de cada una de estas rutinas.

a) La rutina de servicio para el Output Compare 2 (OC2), y la rutina de servicio para el Real-Time Interrupt (RTI), permiten activar los registros de tiempo así como los vectores OC2 y RTI proporcionando el conteo, la comparación y las interrupciones periódicas necesarias para la ejecución de las demás rutinas. En este caso la tasa de interrupción entre rutinas es de 32.77 mseg (milisegundos) debido a que el cristal que se maneja es de 8 MHz y los bits RTR1 y RTR0 del registro PACTL están en "uno".

b) La rutina de inicialización de la comunicación serie, no es más que la configuración del SCI, la cual permite por medio del registro SCCR1 (Serial Communication Control Register 1) definir el formato

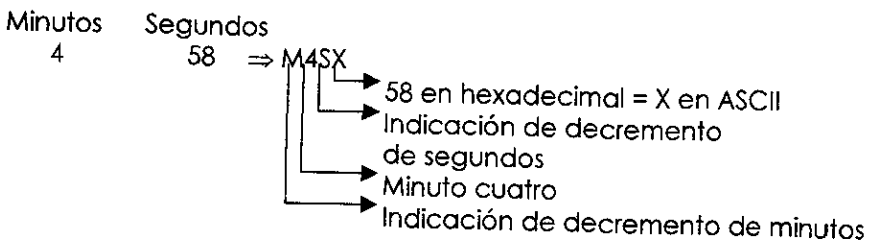
## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

de datos, en este caso con el comando CLR (clear) se limpia el registro SCCR1 quedando todos sus bits en cero, dando como resultado que el formato de datos elegido sea un bit de inicio, 8 bits de datos, y un bit de paro (si M en el SCCR1 es igual a cero, se tiene; Start bit, 8 Data bits, 1 Stop bit). Para configurar la velocidad de transmisión se carga de manera inmediata el número 32 (0x32) hexadecimal en el registro BAUD, con la cual se obtiene una velocidad de 2,400 bps (bits por segundo), ver el anexo B. Como en nuestro caso se requiere habilitar el transmisor y el receptor internos en el SCI y además habilitar la interrupción al momento de recibir información, se carga de manera inmediata el número 2C (0x2C) en el registro SCCR2 (Serial Communication Control Register 2). Como nos podemos dar cuenta esta configuración es muy similar a la realizada en el programa TTY (ventana TTY Settings...).

La configuración del SCI mostrada en esta rutina es más que suficiente para realizar el enlace, solo que para transmitir caracteres utilizamos la subrutina **WRITEC**, y para recibir caracteres utilizamos la rutina **SVSCI**, que en este caso, se detectan errores de ruido en la línea.

c) La rutina de inicialización, permite inicializar las variables, esto es; la variable TIME se carga con 10 décimas de segundo, la variable MINS se carga con 5 minutos de transmisión, se limpia a cero la variable SEGS para iniciar en 0 segundos, también la variable DECSGS se limpia para iniciar en 0 décimas de segundo. En general, esta rutina permite configurar el reloj de conteo.

d) La rutina de actualización del cronometro realiza el decremento de décimas de segundo, segundos y minutos, por ejemplo; 5:00:00 a 4:59:09 a 4:58:07 a 4:55:05. Al momento que va decrementando los segundos envía por medio de la subrutina WRITEC la letra "S" junto con otra letra que es equivalente a ese número hexadecimal en código ASCII. Cuando decremента 59, es decir un minuto, envía por medio de la subrutina WRITEC la letra "M" junto con el minuto decrementado. Lo anterior se muestra a continuación.



4	57	⇒ SW, esto se interpreta como segundo 57
4	56	⇒ SV, esto se interpreta como segundo 56
4	55	⇒ SU, esto se interpreta como segundo 55
3	58	⇒ M3SX, esto se interpreta como minuto tres, segundo 58

e) La rutina de servicio RTI (SVRTI) permite regresar a la interrupción cuando se terminan de decrementar los 5 minutos, además limpia los registros de tiempo para comenzar otra vez con el conteo desde cero, en caso de que se reinicie el programa. Cabe mencionar que si el programa se reinicia faltándole dos minutos para terminar, este comenzará desde el principio, puesto que el *stack pointer* marca el origen del programa, es decir en la localidad 0047.

f) La rutina de servicio SCI y de detección de errores (SVSCI) establece la recepción de los datos. En el caso de que se reciba la letra "R", esta rutina la compara con otra letra "R" que tiene guardada en un acumulador (acumulador A), posteriormente, si estos dos datos son iguales el programa salta a la subrutina **CHANGE**, donde se carga la variable MINS con 9 y se limpia a ceros la variable SEGS. Con este proceso la transmisión cambia a 9 minutos comenzando desde M8SX hasta M0SX.

Si se recibe la letra "P", la rutina SVSCI la compara con la "R" que tiene en el acumulador, como son diferentes no salta a CHANGE, después la compara con la "P" que tiene guardada en el acumulador A, como ahora estos dos datos son iguales salta a la subrutina **PARO**, la cual se encarga de limpiar a cero la variable MINS y la variable SEGS, ocasionando que la rutina de actualización del cronometro deje de decrementar, es decir, se detiene el conteo.

Si se recibe la letra "N", la rutina SVSCI la compara con la "R" que tiene en el acumulador, como son diferentes no salta a CHANGE, después la compara con la "P" que tiene en el acumulador, como también son diferentes no salta a PARO y por último la compara con la "N" que tiene guardada en el acumulador A, como ahora estos dos datos son iguales salta a la subrutina **NEF**. Esta subrutina se encarga de detectar errores de ruido mediante el registro SCSR (Serial Communication Status Register). Si la bandera NF (Noise Error Flag) del registro SCSR se prende, envía o transmite el carácter "©" considerando que existe ruido en la línea, pero si esta bandera no sé

prende no se envía ningún carácter, indicando que en ese instante no existe ruido.

Cabe recordar que el microcontrolador recibe los datos por medio de la rutina SVSCI y además transmite información mediante la subrutina WRITEC, siendo estos dos elementos del programa PROTOCON fundamentales para que exista comunicación entre computador y microcontrolador.

g) La última rutina que contiene la sección Main Program, es la rutina de servicio Output Compare 2 (SV5OC2), la cual permite llamar a la interrupción OC2 cuando los registros de tiempo han adicionado el último valor a comparar, proporcionando el resultado y regresando a la rutina de servicio para el OC2.

Vectores de interrupción y reset.- Esta sección es la última del programa PROTOCON, donde aquí se definen las direcciones en memoria (origen) de las rutinas SVSCI, SV5OC2, SVRTI, y el inicio del programa (Start), es decir, el vector Stack Pointer.

De acuerdo a la explicación anterior, el programa PROTOCON le permite al microcontrolador transmitir una serie de letras durante 5 minutos, le permite recibir comandos para aumentar el tiempo de transmisión, para detener en cualquier instante la transmisión y para detectar errores de ruido. Para lograr lo anterior el programa PROTOCON configura primeramente el SCI, con lo cual se puede realizar el enlace hacia el TTY que tiene los mismos parámetros como velocidad de transmisión y formato de datos. Con cualquiera de estos parámetros que sea diferente no se podrá realizar el enlace.

Para ejecutar el programa de comunicación PROTOCON, es necesario que este cargado en la memoria EEPROM (Electrical Erasable Programmable Read Only Memory) del microcontrolador. Este procedimiento se realiza por medio de un programa llamado "PCBUG", el cual corre en ambiente MS-DOS. Este programa permite editar los listados de instrucciones en ensamblador, posteriormente compila estas instrucciones y después las baja o carga hacia el microcontrolador. En la figura 3.21 se muestra el programa PROTOCON compilado (PROTOCON.s19) listo para cargarse y ser ejecutado.

```
S123F8008E0047CE10008640A720A723A7228640A724  
A7258603A7266F2CCC322CA72BE7E0  
S123F8202D861A9705860A9709860597004F970197028  
60397039706860197020E20FE1F5E  
S123F8402E80FCE72F397A000227013996099702DC00  
2601CFC6F0D401860F940127074A97  
S123F8601B97018D21395D270B86095AC4F01B97018D  
1339865997017A0000C64D8DC0D60B  
S123F88000CB308DBA39C6538DB5D6018DB17A00032  
A048603970318CE0000D60354183A46  
S123F8A018A600D603542504484848484F09A03A7043  
91D25BFD00270B7A0006260686DA  
S123F8C00397068D813B860997004F970120277F00007  
F0001201F8604A5E27022017C626  
S123F8E0A9E72F2011A62EA62F0F815227D8815027DD  
814E27E10E3BDC05E318ED188D8E99  
S107F9001D23BF3BC5  
S105FFD6F8E548  
S105FFE6F8F825  
S105FFF0F8B360  
S105FFFEF80005  
S9030000FC
```

Fig. 3.21. Programa PROTOCON compilado.

Al momento que el programa PROTOCON es cargado en la EEPROM (algunas veces se maneja el término E<sup>2</sup>) del microcontrolador, se procede a restablecer todo el sistema interno por medio del push boton Pb1, ver la figura 3.7 del punto 3.3. En esta figura también se presentan las principales conexiones para el adecuado funcionamiento del microcontrolador. Con este restablecimiento se ocasiona que se ejecute el programa PROTOCON y comience a transmitir una serie de caracteres durante 5 minutos. Estos caracteres o letras, las podemos observar en el TTY que se encuentra en el otro extremo, ver la siguiente figura.

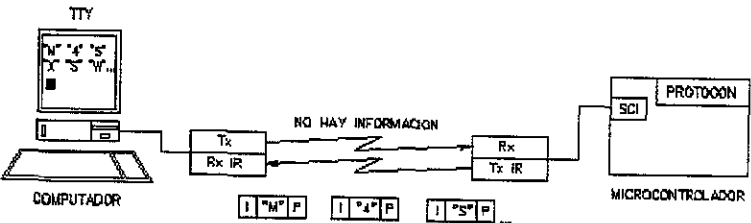


Fig. 3.22. El microcontrolador transmitiendo durante 5 minutos, y el computador recibiendo esa transmisión también durante 5 minutos.

A partir de este momento ya se ha establecido el enlace computador-microcontrolador empleando tanto hardware como software.

Después de transcurridos cinco minutos, el microcontrolador deja de transmitir y el computador por medio del programa terminal TTY termina de desplegar la información recibida, presentando solamente el parpadeo del cursor. Si el computador envía la letra "R" por medio del teclado, la cual visualizamos en pantalla, hacia el microcontrolador este comenzará a transmitir nuevamente pero ahora cuatro minutos más que la transmisión anterior, es decir, 9 minutos. Lo anterior se muestra en la siguiente figura.

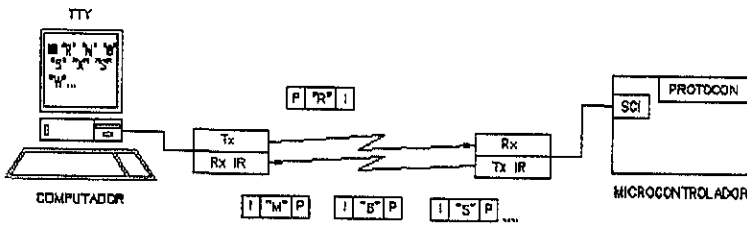


Fig. 3.23. El computador enviando un comando (carácter "R"), y el microcontrolador recibiendo esa transmisión para posteriormente transmitir durante 9 minutos.

Si no se desea esperar 9 minutos de transmisión del microcontrolador, se puede detener la transmisión al enviarle la letra "P" desde el computador. Lo anterior se muestra en la siguiente figura.

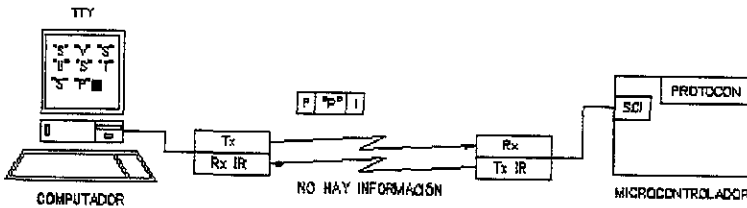


Fig. 3.24. El computador envía un comando de paro al microcontrolador, este recibe la petición para posteriormente interrumpir la transmisión.

Después de interrumpir la transmisión, el computador y el microcontrolador solamente están enlazados por portadora sin transmitir información, como se observa en la siguiente figura.

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

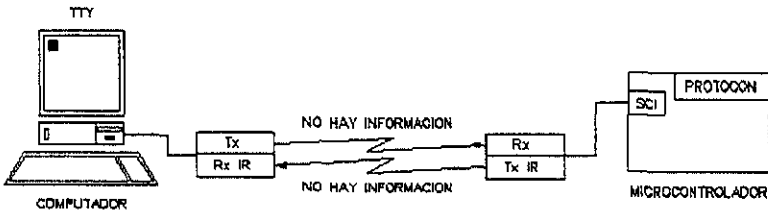
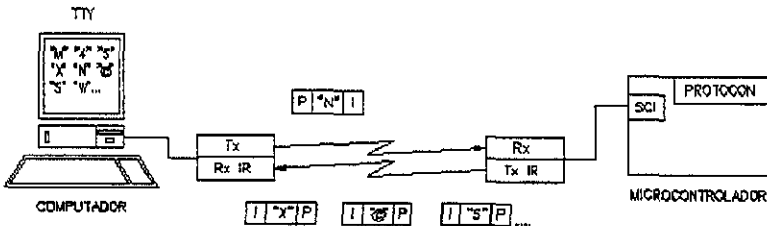
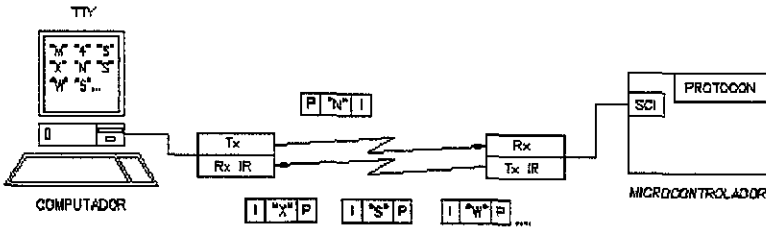


Fig. 3.25. Enlace vivo entre computador y microcontrolador.

Si se desean detectar errores de ruido en el enlace, se recurre a reiniciar el programa PROTOCON. En este momento se comienza a transmitir información durante 5 minutos, posteriormente se envía la letra "N" al microcontrolador. Si existe algún error, el microcontrolador contestará con el carácter "©", si no existen errores simplemente no contestará. Lo anterior se muestra en la siguiente figura.



a) Error de ruido detectado.



b) No hay errores de ruido.

Fig. 3.26. Detección de errores de ruido por medio del comando "N".

En el caso que se necesiten detectar errores de re arranque se emplea el programa **PROTCOR**, el cual es muy similar a PROTOCON. El programa PROTCOR solamente presenta diferencias en la rutina SVSCI y además en lugar de la subrutina NEF, se coloca la subrutina **OREF** la cual detecta, en el caso de que existan, errores de re arranque. Lo anterior se muestra a continuación:

```

*****
*** Detección de error de Rearranque (OR) con "O" desde el puerto serie ***
*****
OREF      LDAA  #$08
          BITA  SCSR,X
          BEQ  VISUAL
*
*        LDAB  #'D'
          STAB  SCDR,X
          BRA  EXITSCI
VISUAL    LDAB  #'D'
          STAB  SCDR,X
          BRA  EXITSCI

*
* *****
* * SVSCI- Rutina de servicio SCI y de detección de errores
* * Retoma a la interrupción RTI
* *****

SVSCI     LDAA  SCSR,X      Lectura para borrar la bandera RDRF
          LDAA  SCDR,X      Lee el dato en el acumulador A (ACCA)
          SEI                Deshabilita interrupciones
          CMPA  #'R          Reinicializar conteo
          BEQ  CHANGE
          CMPA  #'P          Parar conteo
          BEQ  PARO
          CMPA  #'O          Detección de error de re arranque (OR)
          BEQ  OREF

EXITSCI   CLI
          RTI                ***** Regresa al servicio SCI *****

```

La rutina SVSCI compara la letra "O" recibida con la que se tiene en el acumulador A, si estos dos datos son iguales, el programa salta a la subrutina OREF. Esta subrutina se encarga de detectar errores de re arranque por medio de la bandera OR (Overun Error Flag) contenida en el registro SCSR. Si la bandera OR se prende envía o transmite el carácter "D" indicando que existe un error de re arranque, pero si no se prende no se envía ningún carácter.



Por último, si se necesitan detectar errores de trama se utiliza el programa **PROTCOT**. Este programa es muy similar a **PROTOCON**, el cual también presenta diferencias en la rutina **SVSCI**, así como la incorporación de la subrutina **FEF**, en lugar de la subrutina **NEF**. Lo anterior se muestra a continuación:

```

*****
**** Detección de error de trama (FE) con "T" desde el puerto serie ****
*****
FEF          LDAA  #$02
             BITA  SCSR,X
             BEQ   JISUA
*           LDAB  #'f'
*           STAB  SCDR,X
             BRA   EXITSCI
JISUA        LDAB  #'f'
             STAB  SCDR,X
             BRA   EXITSCI

*           *****
*           * SVSCI.- Rutina de servicio SCI y de detección de errores
*           * Retoma a la interrupción RTI
*           *****

SVSCI        LDAA  SCSR,X      Lectura para borrar la bandera RDRF
             LDAA  SCDR,X      Lee el dato en el acumulador A (ACCA)
             SEI                    Deshabilita interrupciones
             CMPA  #'R          Reinicializar conteo
             BEQ  CHANGE
             CMPA  #'P          Parar conteo
             BEQ  PARO
             CMPA  #'T          Detección de error de trama (FE), esto es, se
             BEQ  FEF          detecta un cero en lugar de un bit de paro

EXITSCI CLI
             RTI                ***** Regresa al servicio SCI *****

```

La rutina **SVSCI**, para este programa, compara la letra "T" recibida con la que se tiene en el acumulador A, si estos dos datos son iguales, el programa salta a la subrutina **FEF**. Esta subrutina se encarga de detectar errores de trama, esto es, se detecta un cero en lugar de un bit de paro (un "uno").

La subrutina **FEF** utiliza la bandera **FE** (Framing Error) contenida en el registro **SCSR**. Si esta bandera se prende envía el carácter "f" indicando que se detectó un cero, pero si no se prende no se envía ningún carácter, indicando con esto que se detectó un bit de paro, lo cual es normal debido al formato de datos que empleamos.

Tanto el programa terminal TTY como el programa PROTOCON de comunicación se emplearon para realizar una serie de pruebas, las cuales se presentan a continuación así como el montaje físico del hardware utilizado.

### 3.5. Implementación y pruebas

Los dispositivos electrónicos de las unidades transmisoras y receptoras se montaron en tablillas prototipo de manera individual, las cuales dan la flexibilidad de ocuparlas como tarjetas desmontables, y además permiten distribuir sin ningún problema todos los componentes. Lo anterior se muestra en las figuras 3.27 y 3.28.

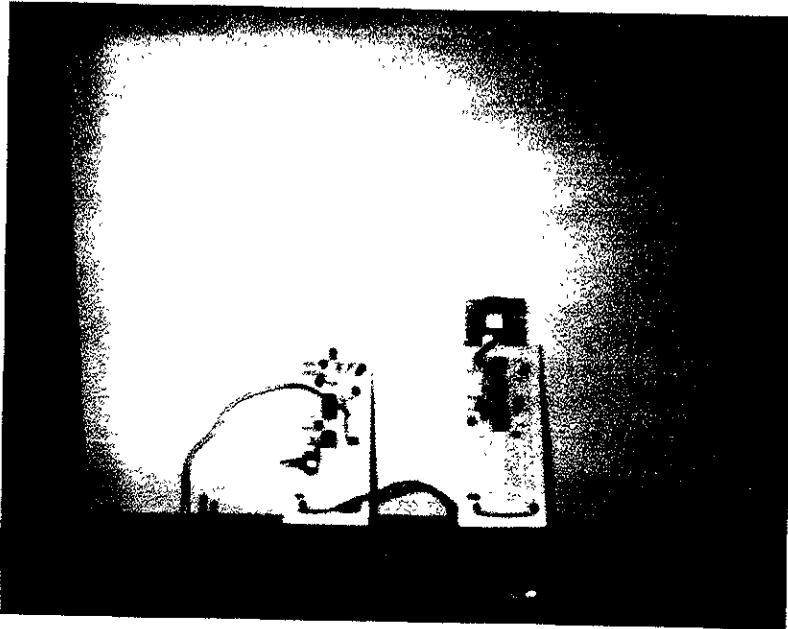


Fig. 3.27. Se presenta la tarjeta transmisora (izquierda) y la tarjeta receptora (derecha), así como un conector DB-9 que se conecta al puerto serie del computador.

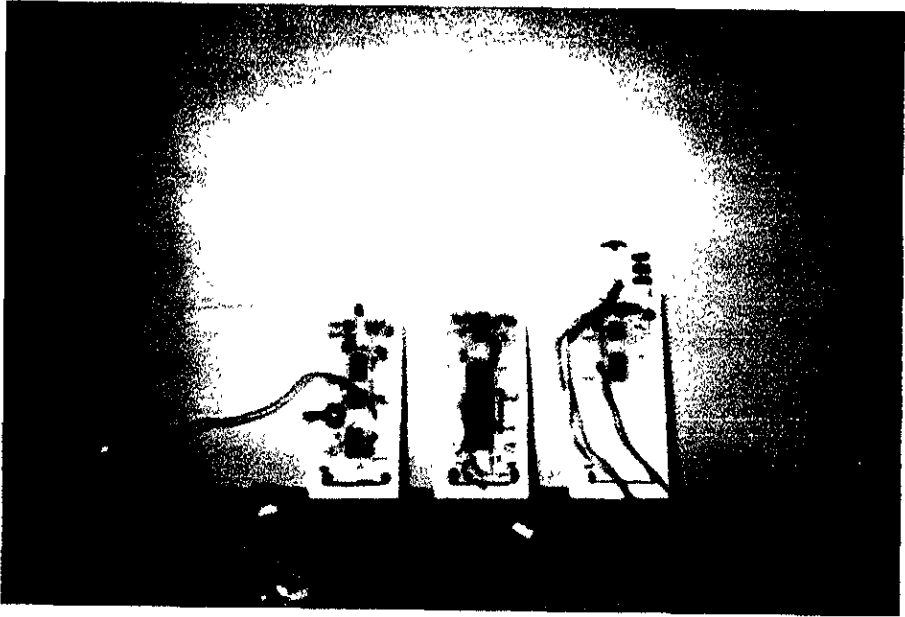


Fig. 3.28. Se presenta la tarjeta transmisora (izquierda), la tarjeta del microcontrolador (en medio) y la tarjeta receptora (derecha), así como un conector DB-9 por si se requiere enlazar un computador.

Como se puede observar en las figuras anteriores, los componentes electrónicos se encuentran muy bien distribuidos y las tarjetas se observan muy manejables y prácticas. En resumen todo el sistema es desmontable considerando a cada tarjeta electrónica como una unidad individual (módulo). Cabe señalar que las tarjetas transmisoras y la tarjeta del microcontrolador miden aproximadamente 14 X 7 cm cada una y las tarjetas receptoras miden 18 X 7 cm cada una. También cabe mencionar que en la parte superior de las tarjetas receptoras se colocó un pequeño filtro óptico de acrílico color rojo opaco.

El montaje de estas tarjetas se realizó sobre estructuras de aluminio como se muestra a continuación.

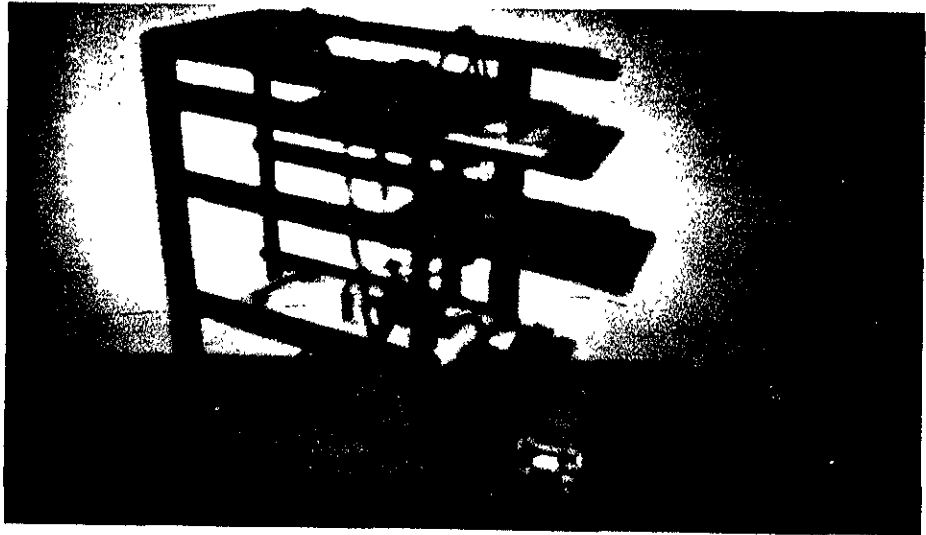


Fig. 3.29. Estructura prototipo del Transmisor-Receptor IR. Arriba se encuentra la tarjeta transmisora y abajo la tarjeta receptora.

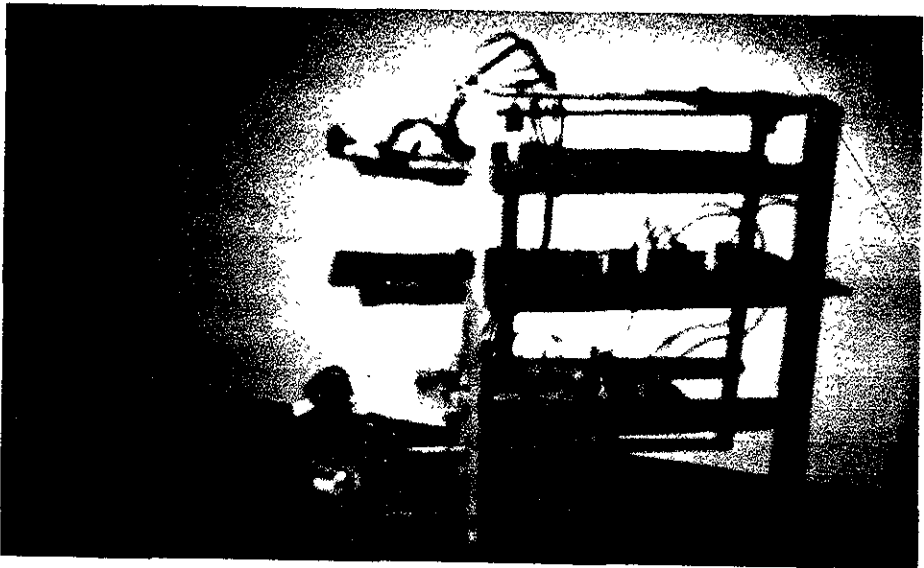


Fig. 3.30. Estructura prototipo del Receptor-Transmisor IR. Arriba se encuentra la tarjeta receptora, en medio la tarjeta del microcontrolador y abajo la tarjeta transmisora.

Para estas estructuras se empleo material aluminio, delgado, ligero, y muy manejable. A estas estructuras se les acondicionaron rieles en tres niveles para que las tarjetas puedan montarse, desmontarse y colocarse en cualquier posición. Cada estructura cuenta con su propio conector DB-9.

Cabe señalar que en el nivel intermedio de ambas estructuras se colocó una placa que permite evitar rebotes del haz IR, evitando con esto que se pudiera originar algún tipo de interferencia entre emisor y receptor durante el desarrollo de las pruebas. Para la estructura del Receptor-Transmisor IR esta placa se colocó de manera consecutiva a la tarjeta del microcontrolador, ver figura 3.30.

Estas estructuras dan la posibilidad de colocar, por ejemplo, el Transmisor-Receptor IR en cualquier superficie plana dando libertad de alinear el sistema sin ningún problema con respecto al Receptor-Transmisor IR, que también puede colocarse sobre cualquier superficie. Con respecto a la alineación del sistema, lo veremos más adelante.

Otro factor que permite que los circuitos tengan libertad de colocarse en cualquier lugar, son las fuentes de alimentación, las cuales se presentan a continuación.



Fig. 3.31. Alimentación del sistema. Dos fuentes de voltaje individuales que proporcionan +5 Vcd cada una.

El diseño de estas fuentes de alimentación es propio, así como el montaje, el cual se realizó en cajas pequeñas de lamina. El tamaño de las fuentes, como se observa en la figura, es muy práctico permitiendo con esto que en general todo el sistema sea muy manejable, desmontable y consecuentemente modular. Lo anterior se muestra en la siguiente figura.

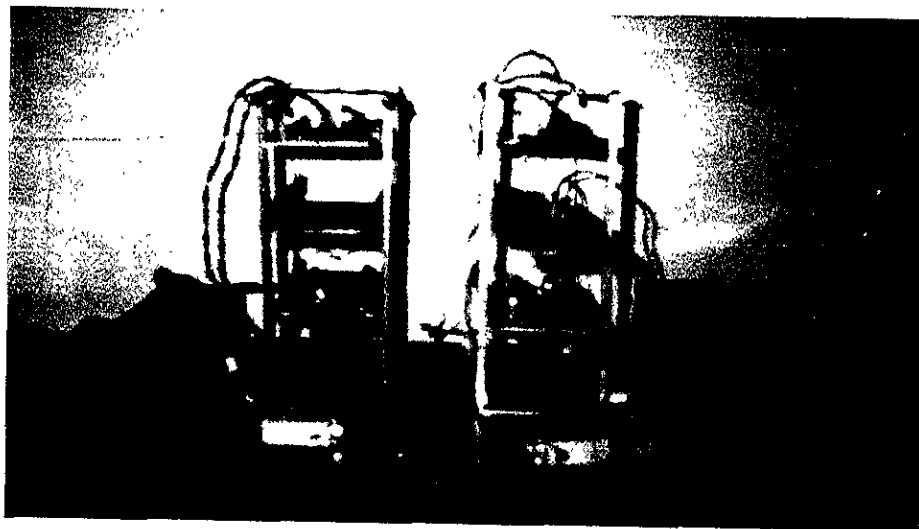


Fig. 3.32. Montaje completo de todo el sistema con aditamentos (Fuentes de poder y conectores).

La figura anterior presenta el montaje completo de todo el sistema de comunicación inalámbrico IR, el cual se empleo para realizar las pruebas operativas. Antes de pasar al desarrollo de las pruebas se presenta, ver la figura 3.33, un panorama general de todo el sistema IR ya implementado, junto con conexiones entre tarjetas, conectores y alimentación, así como parte del computador utilizado para realizar las respectivas pruebas de operación del sistema.

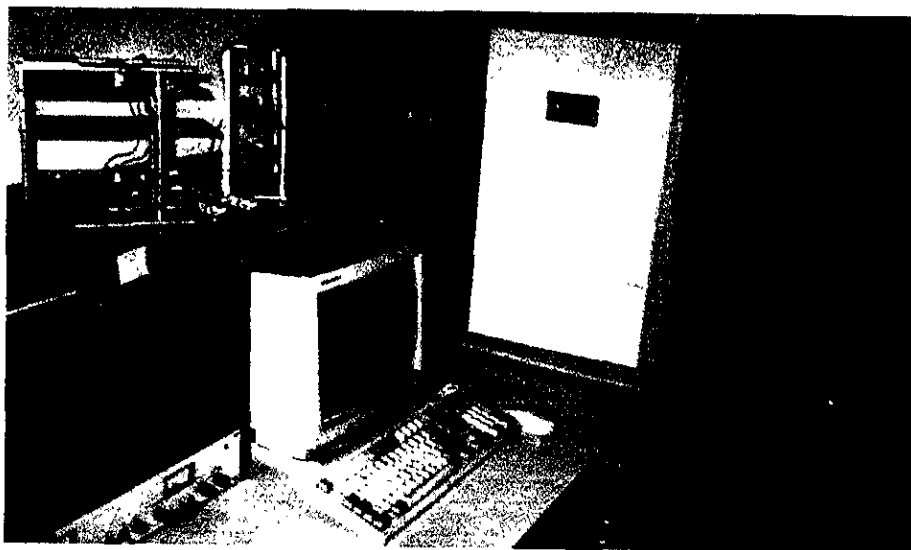


Fig. 3.33. Vista general del sistema de comunicación inalámbrico infrarrojo (IR).

Teniendo todo el sistema IR montado, se dio paso a la realización de las pruebas operativas considerando los siguientes parámetros:

- Dimensiones del laboratorio donde se realizaron las pruebas (16.2 x 8.4 m).
- Intensidad de luz en el laboratorio (8 lámparas fluorescentes).
- Aparatos que emiten radiación infrarroja (2 monitores y las 8 lámparas).
- Temperatura ambiente del laboratorio (18°C).

Conforme a los parámetros anteriores se realizó lo siguiente:

### Prueba #1 "Velocidad de transmisión máxima"

El enlace que se estableció para realizar esta prueba fue entre dos computadores, ver la figura 3.34, a los cuales se les instaló el programa terminal TTY. Antes de desarrollar la prueba se configuraron los parámetros de comunicación de ambos computadores como se muestra a continuación:

	Computador A	Computador B
Puerto	COM1	COM1
Velocidad de transmisión	300 bps	300 bps
Bits de datos	8	8
Paridad	None	None
Bits de paro	1	1
Control de flujo	XON/XOFF	XON/XOFF

Para obtener la velocidad máxima que soporta el equipo para transmitir información, se hizo variar el parámetro de la velocidad. Primero, se realizó a 300 bps transmitiéndose la palabra "hola" del computador A hacia el computador B y viceversa, con claridad. Posteriormente, se cambió la velocidad a 1,200 bps, en ambos computadores, transmitiéndose de igual forma la palabra "hola" del computador A hacia el computador B y viceversa, con claridad. Enseguida, se realizó el mismo experimento a 2,400 bps sin presentar problemas, pero cuando se varió la velocidad a 4,800 bps se presentaron errores, por ejemplo, en lugar de recibir la palabra "hola", se recibió "bola". Debido a esto se concluyó que la velocidad de transmisión máxima del sistema es de 2,400 bps garantizando que a esta velocidad se puede establecer un enlace confiable.

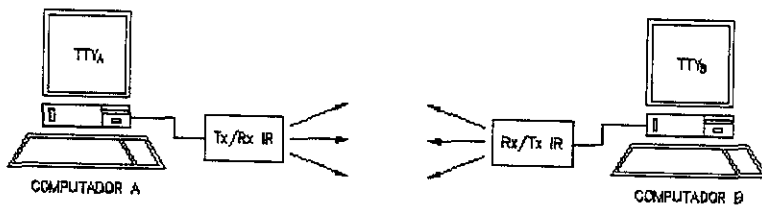


Fig. 3.34. Enlace computador-computador, para desarrollar la prueba #1.

La velocidad de 2,400 bps es suficiente, puesto que el sistema solo requiere enviar comandos, y no transferir grandes cantidades de información que estén en función del tiempo. En los sistemas de control supervisorio que emplean Unidades Terminales Remotas (UTRs) para mover equis cantidad de datos, a menudo transmiten a 300 bps. Existen otros sistemas de control supervisorio que trabajan arriba de 2,400 bps.



## Prueba #2 "Distancia de operación máxima, ángulos de transmisión y recepción"

Se estableció un enlace IR entre un computador y un microcontrolador, como se observa en la figura 3.35, para obtener la distancia de operación máxima del sistema así como los ángulos de transmisión y recepción. Antes de realizar esta prueba se instaló el programa terminal TTY en el computador y se cargo el programa PROTOCON de comunicación en el microcontrolador.

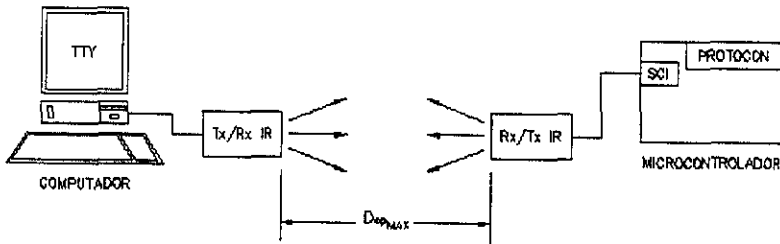


Fig. 3.35. Enlace computador-microcontrolador, para desarrollar la prueba #2.

Esta prueba se desarrolló colocando el computador en una posición fija, donde el microcontrolador se retiró a 4, 4.5, 5, 5.5, y 6 m. Cuando el microcontrolador se colocó a 4 m del computador, el enlace no presentó síntomas de atenuación de la señal, lo cual pudimos observar en el TTY, puesto que si ocurría algún problema este desplegaría en pantalla un error, por ejemplo; si nosotros interrumpíamos la comunicación de manera intencional, el TTY desplegaría en pantalla el error <CE-24>, y si no interrumpíamos la comunicación, el TTY mantenía el cursor parpadeando. Por lo tanto a 4 m, el cursor seguía parpadeando indicando que el enlace no presentaba problemas.

Estas mismas características se presentaron hasta más de 5.5 m, exactamente 5.6 m, donde el enlace permaneció "vivo". Cuando esta distancia se sobrepasó, por ejemplo, a 6 m el TTY desplegó errores <CE-24> indicándonos que a esta distancia se rompía el enlace de comunicación. Por lo tanto, el enlace se garantiza hasta 5.6 m, pero si se pasa esta distancia el enlace no es confiable.

Para obtener los ángulos de transmisión y recepción, se hizo girar tanto a la derecha como a la izquierda la estructura del Transmisor-Receptor IR. Al momento de realizar el giro a la derecha, el transmisor envió el haz en otra dirección y el receptor no recibió el haz que le envió el otro extremo,

en ese instante el TTY desplegó en pantalla el error <CE-24>, indicando que en ese ángulo se rompió el enlace, es decir, más de  $13^\circ$  para el transmisor y más de  $64^\circ$  para el receptor. Cuando se realizó el giro a la izquierda, se rompió el enlace en más de  $19^\circ$  para el transmisor y más de  $62^\circ$  para el receptor. Lo anterior se muestra en las figuras 3.36 y 3.37. Cabe señalar que estos ángulos corresponden para ambos transmisores y para ambos receptores, puesto que los transmisores tienen los mismos componentes electrónicos y los receptores también tienen los mismos componentes electrónicos.

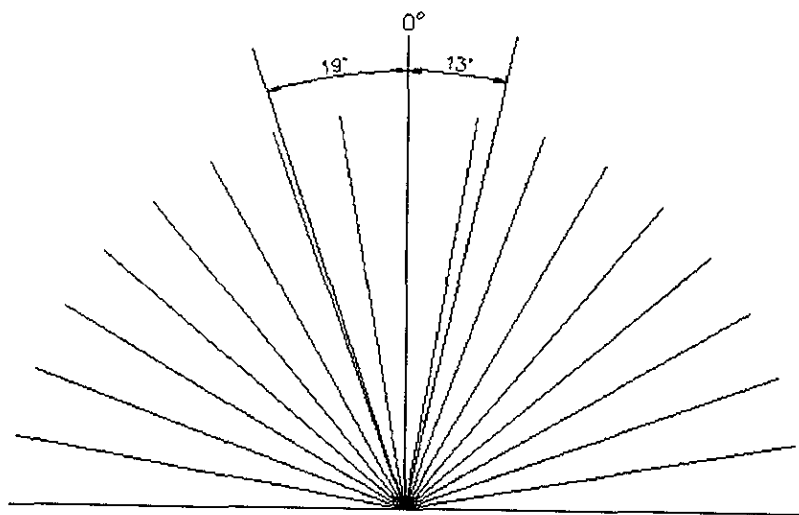


Fig. 3.36. Ángulo de transmisión del sistema IR.

Como se puede observar en la figura 3.36, se transmite con un ángulo de  $32^\circ$  y según la figura 3.37, se recibe con un ángulo de  $126^\circ$ . Esto nos indica que transmitimos un haz IR un poco directivo, pero contamos con una área bastante grande para detectar la información, lo cual demuestra en primera instancia que el enlace de comunicación es confiable.

Estos ángulos permiten alinear sin ningún problema al Transmisor-Receptor IR y al Receptor-Transmisor IR, ocasionando con esto que no sea necesario que se "vean" directamente, sino que pueden estar un poco desviados uno del otro.

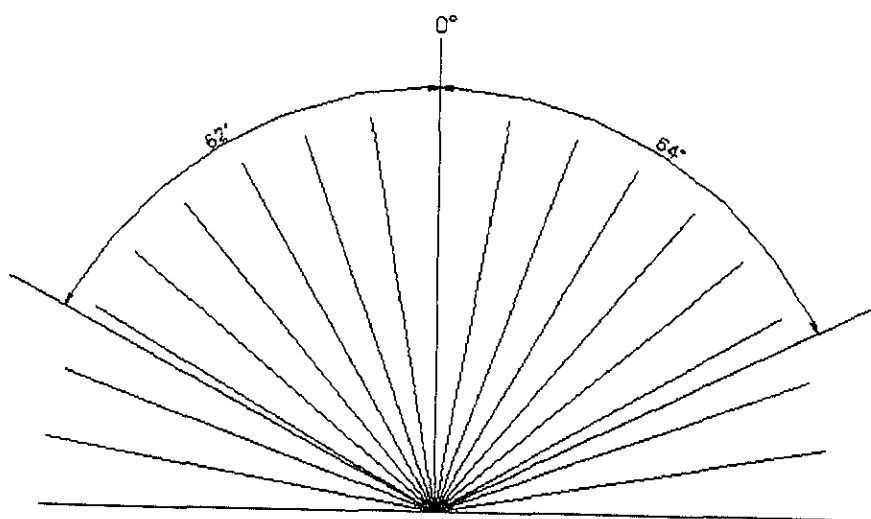


Fig. 3.37. Ángulo de recepción del sistema IR.

Lo que hace a este sistema IR utilizable es al hecho de que el haz se extiende a medida que aumenta el alcance del transmisor y que aumente igualmente el área de visión del receptor. De hecho, el alcance máximo de operación, como se vio al principio de esta prueba, es de 5.6 m, de acuerdo a esta distancia se garantiza la confiabilidad y la calidad del enlace, pero si se sobrepasa esta distancia y además los ángulos de transmisión y recepción no se garantizan buenos resultados.

### Prueba #3 "Tiempo de transmisión libre de errores"

Esta prueba se desarrolló estableciendo un enlace entre un computador y un microcontrolador, como se observa en la figura 3.38. La presente prueba consiste en arrancar el programa PROTOCON contenido en el microcontrolador para comenzar una transmisión de 5 minutos, la cual recibe el computador mediante el programa terminal TTY mostrándola en pantalla como se observa en la parte superior de la figura 3.39.

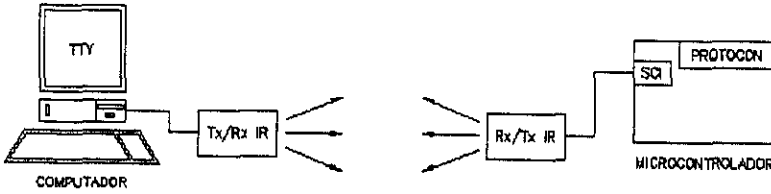


Fig. 3.38. Enlace computador-microcontrolador, para desarrollar la prueba #3.

Posteriormente se envía la letra "R" desde el computador hacia el microcontrolador, donde el programa TTY la muestra en pantalla. Cuando este comando es recibido por el microcontrolador, se arranca la subrutina CHANGE del programa PROTOCON para comenzar una transmisión de 9 minutos, la cual recibe el computador como se puede observar en la parte inferior de la figura 3.39.

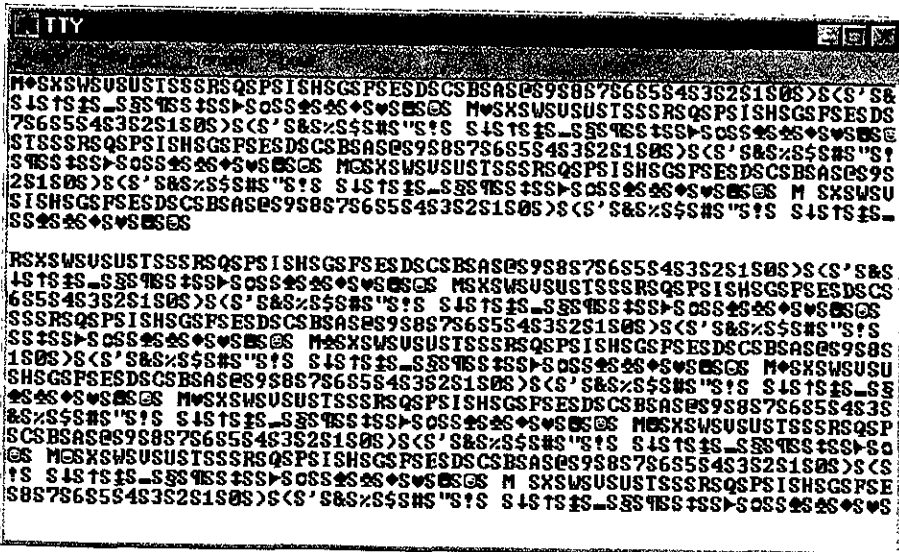


Fig. 3.39. Transmisión durante 14 minutos libre de errores. La parte superior presenta los caracteres transmitidos durante 5 minutos y la parte inferior presenta los caracteres transmitidos durante 9 minutos.

Para comprobar que estos caracteres transmitidos no presentan errores, se comparo la transmisión de la figura 3.39 con una transmisión realizada de manera alámbrica, es decir sin utilizar el sistema IR. Esta transmisión se presenta en la figura 3.40.

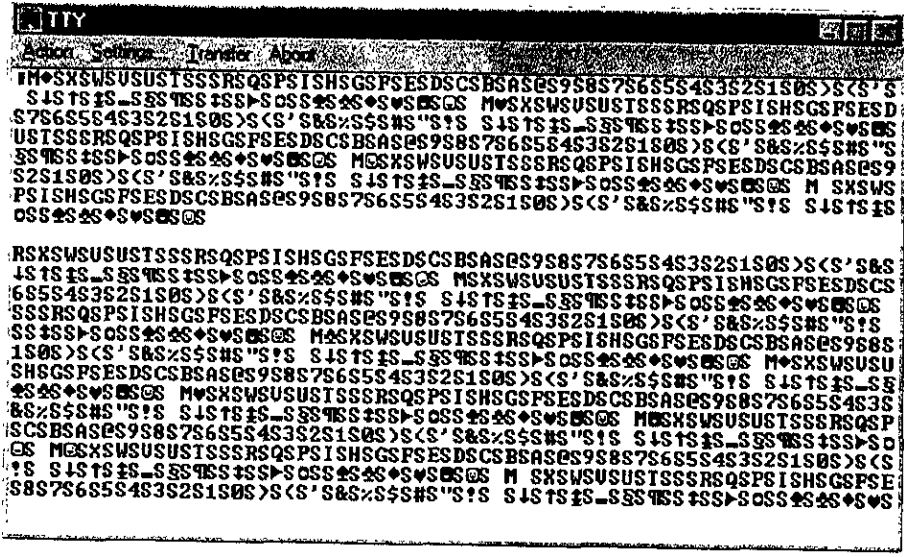


Fig. 3.40. Transmisión durante 14 minutos conectando el microcontrolador directamente al computador sin utilizar el sistema inalámbrico.

La transmisión de la figura 3.40 se obtuvo conectando el microcontrolador directamente al puerto serie del computador mediante el conector DB-9 que se muestra en la figura 3.28.

Como se puede observar, la figura 3.39 y la figura 3.40 son exactamente iguales, aunque en la parte superior de la figura 3.40 se presenta el carácter "□" antes de la letra "M", lo cual es un error de tecleo por parte del operador al momento de realizar la prueba. La comparación anterior nos permite probar que el sistema es confiable, puesto que los datos no son erróneos. En total, se obtuvo un tiempo de transmisión libre de errores durante 14 minutos, entre un computador y un microcontrolador enlazados vía luz IR.

En el caso de que se hubieran obtenido errores durante la transmisión se verían como en la figura 3.41. Si la figura 3.40 se toma como patrón y se compara con la figura anterior, se pueden observar claramente los errores de transmisión, puesto que se recibieron otros caracteres diferentes a los

ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR  
que transmitió el microcontrolador.

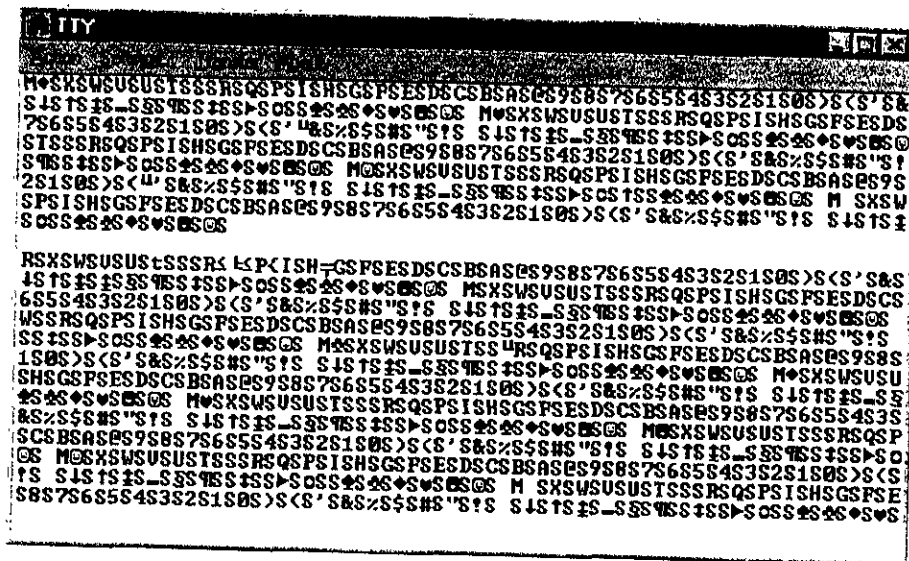


Fig. 3.41. Errores en la transmisión.

Posteriormente se realizó una transmisión inversa para comprobar la confiabilidad del sistema, esto consistió en enviar primero la letra "R" hacia el microcontrolador y transmitir durante 9 minutos, después de terminado este periodo se reancho el programa PROTOCON contenido en el microcontrolador para que este transmitiera durante 5 minutos, ver la figura 3.42.

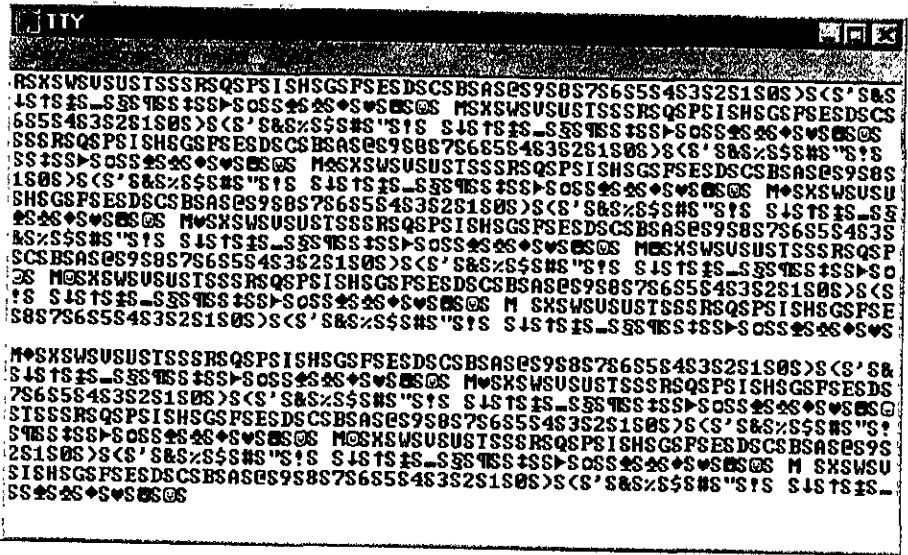


Fig. 3.42. Transmisión inversa durante 14 minutos. La parte superior presenta los caracteres transmitidos durante 9 minutos y la parte inferior presenta los caracteres transmitidos durante 5 minutos.

Esta prueba termino rearrancando el programa PROTOCON, posteriormente se envió la letra "R" para aumentar la transmisión a 9 minutos. Concluidos esos 9 minutos de transmisión, se envió nuevamente la letra "R" pero enseguida se paro la transmisión enviando la letra "P". Lo anterior se muestra en la figura 3.43. Cabe señalar que esta transmisión se comparo con otra transmisión semejante pero realizada de manera alambrica, la cual se presenta en la figura 3.44. Como se puede observar en ambas figuras los caracteres son idénticos.

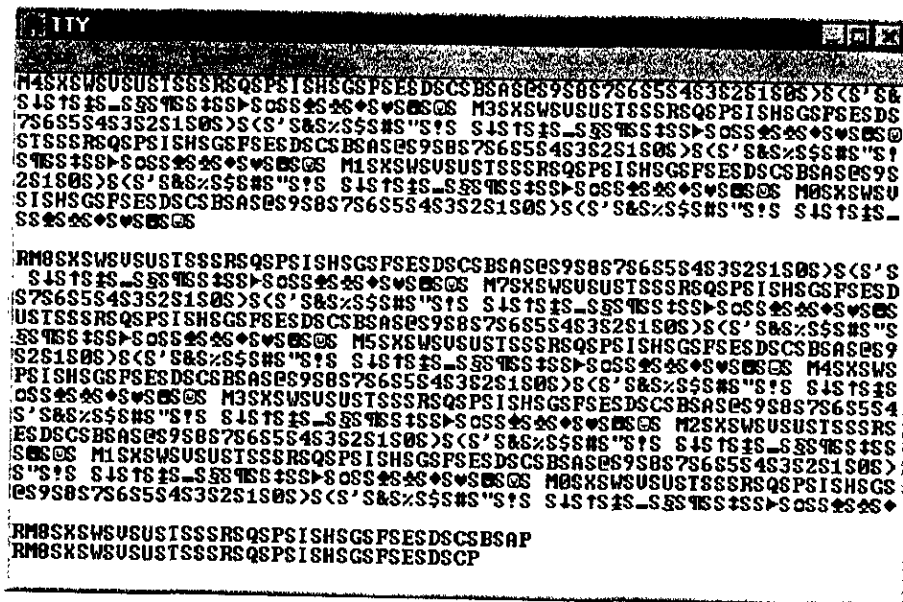


Fig. 3.43. Transmisión durante 14 minutos con paro. La parte inferior muestra el reinicio de la transmisión y enseguida el paro o interrupción de los caracteres recibidos.



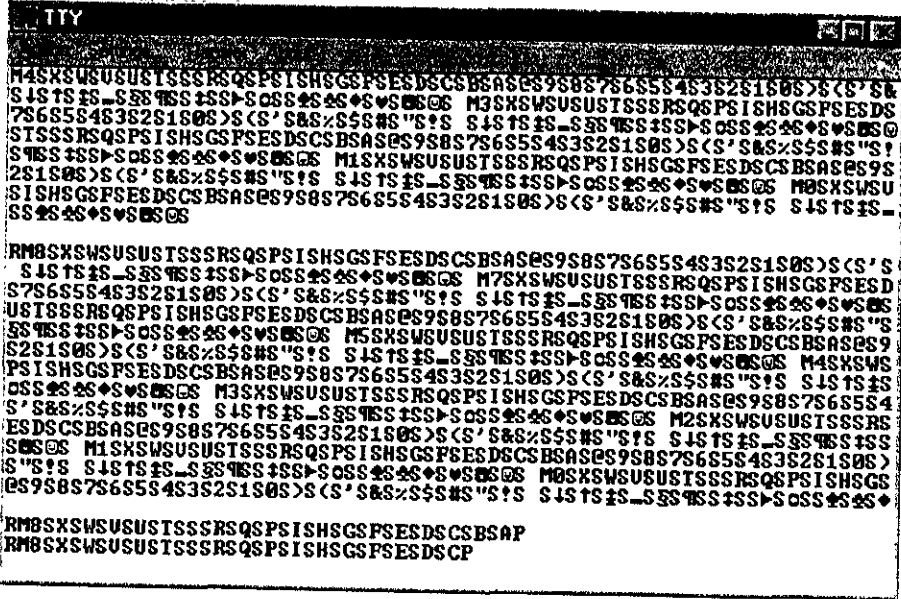


Fig. 3.44. Transmisión durante 14 minutos con paro, conectando el microcontrolador directamente al computador sin utilizar el sistema inalámbrico.

Finalmente, se comprobó que si se realiza una transmisión en un sistema alámbrico, esa transmisión debe de ser la misma si se realiza en un sistema inalámbrico. Esto nos indica que el sistema de comunicación inalámbrico IR desarrollado es confiable, permitiendo realizar transmisiones libres de errores durante varios minutos.

### Prueba #4 "Manipulación del robot Scorbot"

Para desarrollar esta prueba se estableció un enlace computador-brazo robot como se muestra en la siguiente figura.

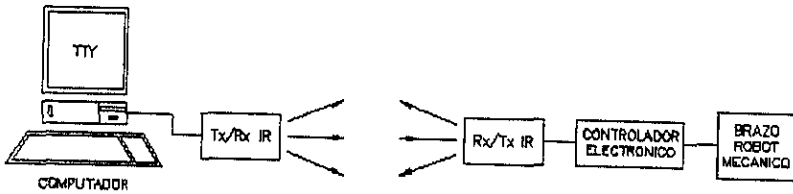


Fig. 3.45. Enlace computador-brazo robot, para desarrollar la prueba #4.

El robot Scorbot-ER III que se emplea para realizar esta prueba, consta de dos elementos principales, los cuales son; El brazo mecánico, ver la figura 3.46, y El controlador electrónico. En el anexo C se presenta una descripción general sobre este robot.



Fig. 3.46. Presentación general del robot Scorbot-ER III (Brazo mecánico).

Este robot, que se encuentra en el laboratorio de automatización del ININ, realizó dos movimientos diferentes obteniéndose muy buenos resultados. Cabe mencionar que para realizar lo anterior, al computador remoto se le instaló el programa TTY para enviar la instrucción de

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

movimiento, y al controlador electrónico del brazo robot se le cargo el programa MOVIR, desarrollado en lenguaje C, para que al momento de recibir la instrucción el controlador ordene al brazo mecánico a realizar el movimiento elegido. A continuación se presenta el listado del programa MOVIR.

```
*****
* Programa en lenguaje C desarrollado para realizar pruebas de transmisión y      *
* recepción de datos, el cual permite establecer un enlace IR desde un computador *
* a un brazo robot (SCORBOT-ER III).                                           *
* *****                                                                    *
* Este programa permite recibir datos desde un computador en código ASCII para  *
* posteriormente procesarlos y entregarlos al controlador del brazo robot. Ya   *
* con los datos recibidos, que son una serie de instrucciones o comandos, el    *
* controlador ordena al robot a realizar algún movimiento predeterminado.     *
*****
```

```
#include<dos.h>
#include <stdio.h>
#include <conio.h>
```

```
union REGS rin,rout;
int *apun;char K;
```

```
/*VARIABLES*/
```

```
void START_PORT1(int P);
void START_PORT2(int P);
char READ_PORT(int P);
void INTERRUPTION(int k, int P);
void CHECK1(char FR,char ER,int P)
```

```
/* PROTOTIPOS DE FUNCIONES*/
```

```
/*INICIALIZAR PUERTO*/
```

```
void START_PORT1(int P)
{
    rin.h.ah=0;
    rin.x.dx=0;
    rin.h.al=0xE7;
    int86(0x14,&rin,&rout);
    /*LIMPIAR BANDERAS*/
    rin.h.ah=3;
    rin.x.dx=0;
    int86(0x14,&rin,&rout);
}
void START_PORT2(int P)
{
    rin.h.ah=0;
```

```

rin.x.dx=1;
nn.h.al=0xA3;
int86(0x14,&rin,&rout);
    /*LIMPIAR BANDERAS*/
rin.h.ah=3;
rin.x.dx=1;
int86(0x14,&rin,&rout);
}

```

```

void INTERRUPTION(int k, int P)
{
    if (k==1)
    {
        /*ACTIVAR INTERRUPCIONES*/
        rin.h.ah=1;
        rin.x.dx=P;
        rin.h.al='W';
        int86(0x14,&rin,&rout);
    }
    else if (k==0)
    {
        /*DESACTIVAR INTERRUPCIONES*/
        rin.h.ah=1;
        rin.x.dx=P;
        rin.h.al='X';
        int86(0x14,&rin,&rout);
    }
    else printf ("Error de opcion\n");
}

```

```

char READ_PORT(int P)
{
    /* checar la respuesta del controlador*/
    /*Estado de la línea*/
    rin.h.ah=3;
    rin.x.dx=P;
    int86(0x14,&rin,&rout);
    while(rout.h.ah & 0x01)
    {
        rin.h.ah=2;
        rin.x.dx=P;
        int86(0x14,&rin,&rout);
        K=rout.h.al;
        /* printf("%c ",rout.h.al);*/
        rin.h.ah=3;
    }
}

```

## ENLACE INFRARROJO ENTRE UN COMPUTADOR Y UN MICROCONTROLADOR

```
rin.x.dx=P;
int86(0x14,&rin,&rout);
}
return(K);
}
void CHECK1(char FR,int P)
{
rin.h.ah=1;
rin.x.dx=P;
rin.h.al=FR;
int86(0x14,&rin,&rout);
/*rin.h.ah=1;
rin.x.dx=P;
rin.h.al=ER;
int86(0x14,&rin,&rout);*/
}
```

```
main( )
{
char ter;
clrscr();
START_PORT1(0);
START_PORT2(0);

do
{
ter=READ_PORT(1);
CHECK1(ter,0);
printf("\t%c",ter);
}while(ter!=27);
}
```

MOVIR permite inicializar los puertos de comunicaciones del controlador electrónico a la recepción, logrando con esto, recibir datos en código ASCII. Al momento que el controlador electrónico recibe los datos, los procesa y le ordena al brazo mecánico a ejecutar algún movimiento.

Esta prueba consiste en enviar desde el computador la instrucción 2M+5, por medio del TTY, hacia el controlador electrónico para que al momento que este reciba la instrucción ordene al brazo mecánico a moverse hacia la derecha.

Esta instrucción en código ASCII le indica al controlador a mover el motor dos en dirección positiva cinco impulsos, esto provoco que se moviera hacia la derecha el hombro del brazo mecánico. Para obtener más detalles acerca de los comandos de movimiento consultar el anexo C.

Posteriormente se le envió el comando 2M-5 desde el computador remoto, ocasionando que el controlador electrónico moviera otra vez el motor dos pero en dirección negativa o contraria cinco impulsos, lo cual provoco que se moviera hacia la izquierda el hombro del brazo mecánico, es decir el brazo regreso a su posición inicial.

Finalmente, esta prueba nos permitió reafirmar la calidad y confiabilidad del enlace de comunicación inalámbrico infrarrojo.

Como se pudo observar, para realizar las cuatro pruebas anteriores se establecieron varios enlaces IR, donde el enlace principal es entre un computador y un microcontrolador, y los enlaces opcionales son entre dos computadores y entre un computador y un brazo robot. También cabe señalar que las pruebas 3 y 4 se realizaron a una distancia entre dispositivos de aproximadamente 5.5 metros.

Por último, el desarrollo de estas pruebas nos permitieron obtener las especificaciones generales del sistema IR, las cuales se presentan en el siguiente capítulo.

# Capítulo IV

## 4. RESULTADOS

Los resultados que arrojaron las pruebas anteriores, simplemente, son las especificaciones del sistema IR, las cuales se presentan a continuación.

PARÁMETRO	DESCRIPCIÓN
INTERFACE	Puerto serie RS-232C, Conector DB-9
CONTROL DE FLUJO	XON/XOFF
CANALES IR	2
ÁNGULO DE TRANSMISIÓN	13° Derecha, 19° Izquierda (32°)
ÁNGULO DE RECEPCIÓN	64° Derecha, 62° Izquierda (126°)
FRECUENCIA DE OPERACIÓN	40 KHz
TIPO DE MODULACIÓN	En frecuencia, Onda Continua (CW)
VELOCIDAD DE TRANSMISIÓN	2,400 bps (bits por segundo)
FORMATO DE DATOS	NRZ (No Retorno a Cero)
DISTANCIA DE OPERACIÓN MÁXIMA	5 metros con 60 centímetros
DIMENSIONES DE LOS TRANSMISORES	14 X 7 centímetros c/u
DIMENSIONES DE LOS RECEPTORES	18 X 7 centímetros c/u
ALIMENTACIÓN DE CA	127 V, 60 Hz
ALIMENTACIÓN DE CD	5 V
TEMPERATURA DE OPERACIÓN	0° C a 25° C
CONSUMO DE CORRIENTE	150 mA
CORRIENTE DE SALIDA DEL TRANSMISOR	100 mA

Tabla 4.1. Especificaciones del sistema IR.

## 5. CONCLUSIONES

Los circuitos electrónicos presentados en este trabajo se diseñaron para una alimentación de 5 Vcd y a una frecuencia de transmisión de 40 KHz. Otras frecuencias portadoras pueden ser utilizadas, con las cuales se pueden obtener muy buenos resultados, como por ejemplo, incrementar la velocidad de transmisión. La distancia que se logró con estos circuitos fue de 5.6 metros considerando un solo diodo emisor infrarrojo a una corriente de 100 mA. Pero esta distancia puede incrementarse a varios metros utilizando dos diodos emisores, con lo cual aparte de aumentar la distancia se aumentan las posibilidades de interferencia ya que para nuestro caso, éste es un sistema de comunicación bidireccional. Cabe señalar, que para distancias grandes a parte de necesitar varios diodos emisores se necesitan múltiples fotoreceptores y además un sistema adicional de lentes de Fresnel, así como un sentido de orientación preciso, ya que se necesitaría alinear los dispositivos exactamente uno frente al otro para que se lleve a cabo el enlace con éxito.

Con el diseño de los circuitos electrónicos y apoyándose de programas de cómputo, se logró desarrollar un sistema de comunicación inalámbrico bidireccional capaz de transmitir datos de control a una velocidad de transmisión moderada y a una distancia de operación entre dispositivos aceptable, haciendo uso de señales infrarrojas y además proporcionando buenos resultados de funcionalidad y confiabilidad.

Se realizaron pruebas operativas del sistema IR no solo entre un computador y un microcontrolador, sino entre otros dispositivos electrónicos, permitiéndose establecer enlaces de comunicación de un computador a otro y de un computador a un controlador de brazo robot. De esta manera se puede garantizar la calidad del sistema IR y además mostrar su flexibilidad.

Finalmente, cabe mencionar que la implementación de los circuitos se realizó con componentes electrónicos disponibles en el mercado, así como también el material utilizado para montaje.



## 6. BIBLIOGRAFÍA

- Elements of infrared technology  
Generation, Transmission, and Detection  
Kruse, Mc Glauchlin, Mc Quistan
  
- Infrared Radiation  
A Handbook for applications  
Bramson
  
- Télécommandes technique et réalisation  
P. Guevle
  
- Modéms technique et réalisation  
C. Tavernier
  
- M68HC11 Reference manual  
Motorola
  
- Computer Networks: Protocols,  
Standards and Interfaces  
Uyless Black
  
- Data and computer communications, fourth edition  
William Stallings, PH. D.

➤ Motorola semiconductor

Application note

➤ Electronic devices and circuit theory, fifth edition

Robert Boylestad, Louis Nashelsky

➤ Apuntes de Costos y Evaluación Económica,  
agosto de 1996

# Anexo A

## El microcontrolador MC68HC811E2

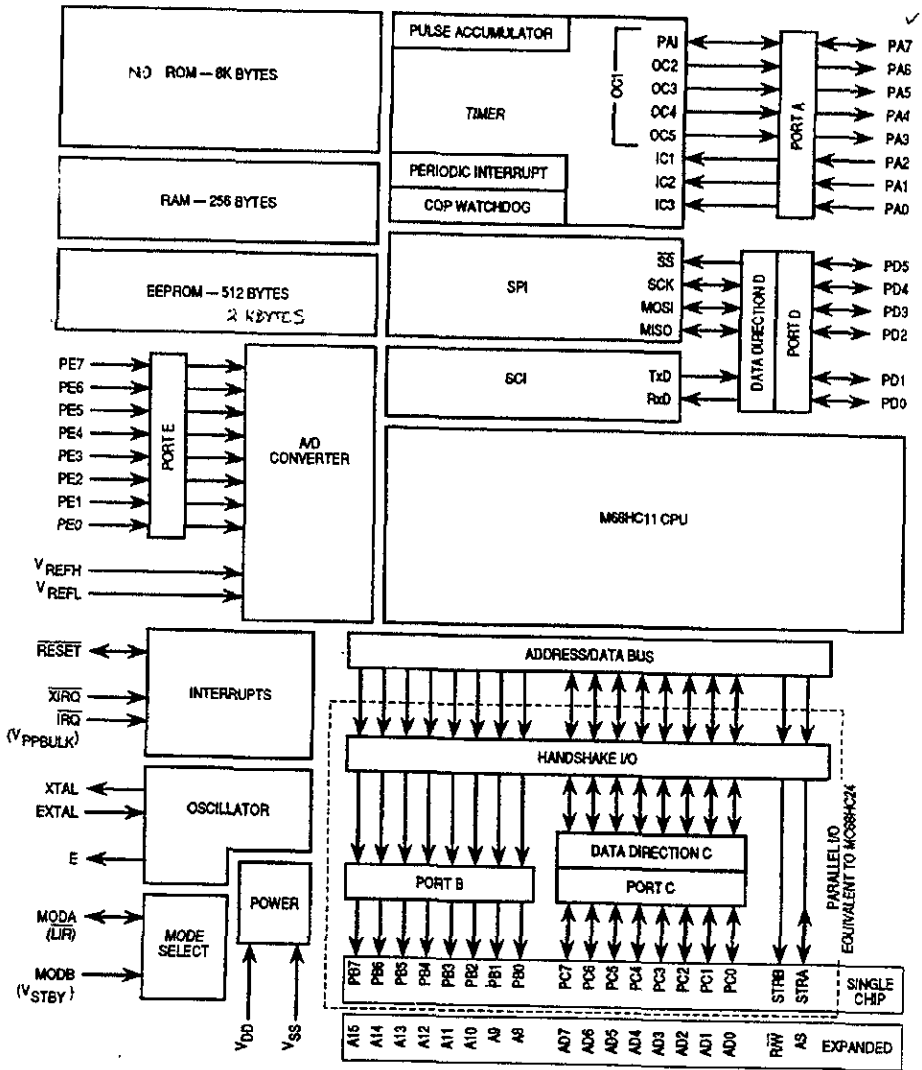
El circuito de alta densidad metal-oxido (HCMOS) es un microcontrolador (MCU) de 8 bits de alta integración con todos los periféricos dentro del mismo encapsulado.

La tecnología HCMOS usada en el MC68HC11 combina un tamaño pequeño y alta velocidad con una baja potencia y alta inmunidad al ruido en los CMOS. Dentro del mismo encapsulado se incluye 8Kbytes de memoria de solo lectura Random Only Memory (ROM), 2Kbytes de memoria programable ROM electricamente borrable (Electrically Erasable Programmable ROM( EEPROM), y 256 bytes de Random Access Memory (RAM).

Dentro del encapsulado son incluidos también 8 canales convertidores analógico-digital (A/D) con 8 bits de resolución. Un puerto de comunicaciones serie asincrono (asynchronous Serial Communications Interface SCI) y un canal sincrono separado (Synchronous Serial Peripheral SPI) son incluidos. El reloj principal activa a un sistema de conteo libre de 16 bits que tiene tres pines de entrada de captura (Input -Capture), cinco salidas comparadoras (Output-compare) y una función de interrupción en tiempo real. Un acumulador de pulsos de Ocho bits subsistema que puede contar eventos externos o medir periodos de tiempo externos.

Un circuito de monitoreo es incluido en el encapsulado para protección contra errores del sistema. Un computer Operating Properly (COP) sistema watchdog protege contra fallas del software. Un sistema de monitoreo del reloj genera un reseteo en caso de que el reloj se pierda o corra muy bajo. Un circuito de detección de códigos ilegales produce una interrupción no enmascarable si un código ilegal es detectado.

La figura siguiente es un diagrama a bloques del MCU. Este diagrama exhibe la mayoría de los subsistemas y la relación con los pines del MCU. En la esquina inferior derecha de este diagrama se observa el subsistema de Entrada salida (I/O) paralelo, la función de este subsistema se pierde cuando el MCU es operado en modo expandido (opción que convierte al MCU en un MCU de 16 Bits).



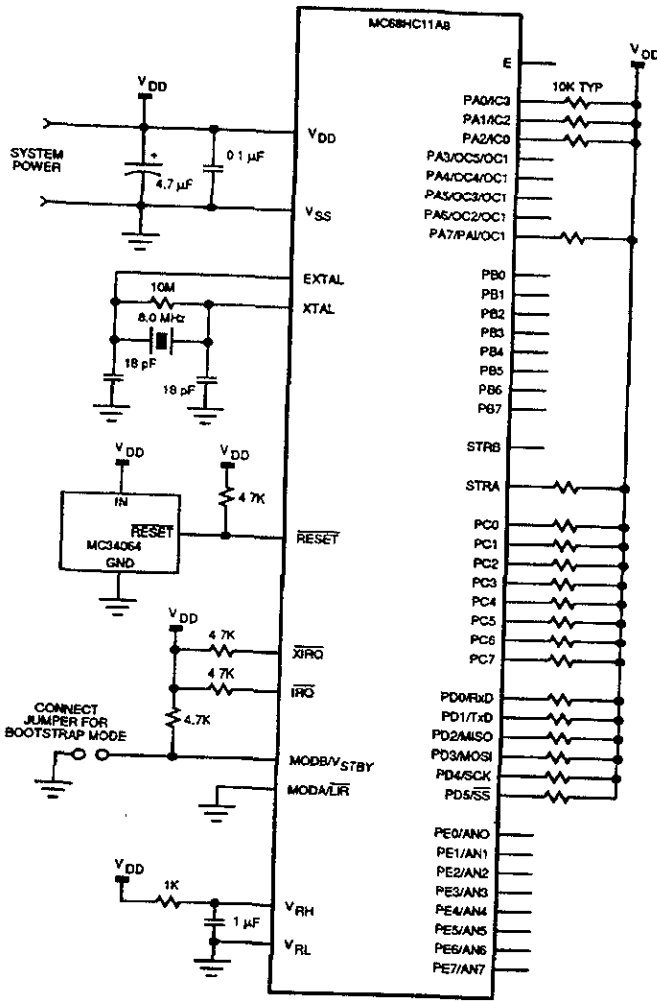
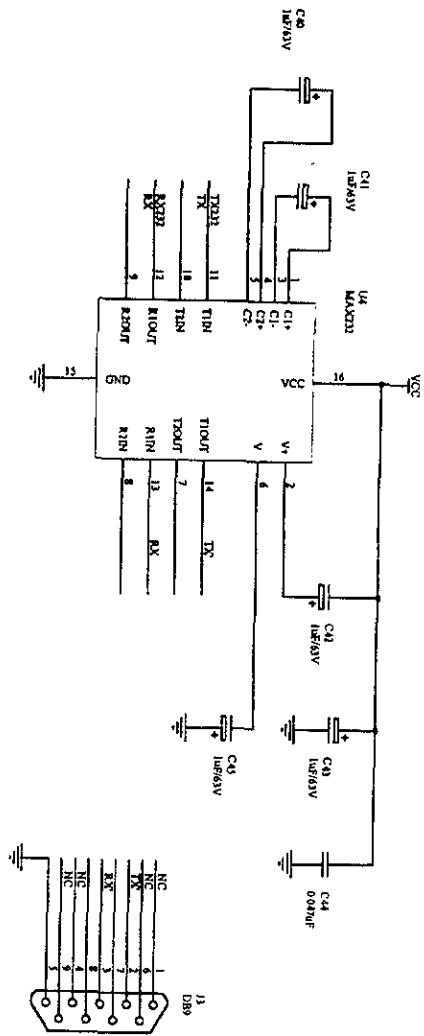


Figure 2-22. Basic Single-Chip-Mode Connections

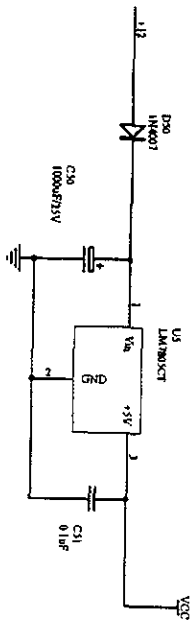
### INTERFACE RS232



### CONECTOR ALIMENTACION



### FUENTE DE ALIMENTACION



La alimentación de este MCU se realiza a través de los pines VDD (Positivo) y VSS (Tierra) este, usa una fuente de poder simple de 5 Vcc. Pero en algunas aplicaciones puede ser necesaria una fuente opcional de referencia cuando son utilizados los convertidores Analógicos –Digitales (A/D) o por una batería de respaldo que mantiene energizada la memoria RAM cuando se desactiva la alimentación principal, esta fuente adicional opcional opera también a un voltaje de 5V.

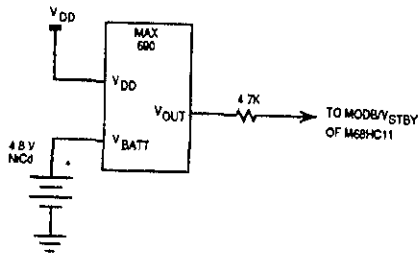


Figure 2-8. RAM Standby MODB/VSTBY Connections

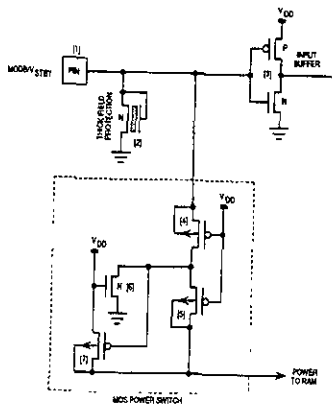


Figure 2-20 Internal Circuitry — MODB/VSTBY Pin

Este MCU posee 145 códigos de operación que lo hacen más versátil y fácil de programar.

**ABA**

Add Accumulator B to Accumulator A

**ABA**

**Operation:**  $ACCA \leftarrow (ACCA) + (ACCB)$

**Description:** Adds the contents of accumulator B to the contents of accumulator A and places the result in accumulator A. Accumulator B is not changed. This instruction affects the H condition code bit so it is suitable for use in BCD arithmetic operations (see DAA instruction for additional information).

**Condition Codes and Boolean Formulae:**

S	X	H	I	N	Z	V	C
—	—	*	—	*	*	*	*

**H**  $A3 \cdot B3 + B3 \cdot \overline{R3} + \overline{R3} \cdot A3$

Set if there was a carry from bit 3; cleared otherwise.

**N** R7

Set if MSB of result is set; cleared otherwise.

**Z**  $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$

Set if result is \$00; cleared otherwise.

**V**  $A7 \cdot B7 \cdot \overline{R7} + \overline{A7} \cdot \overline{B7} \cdot R7$

Set if a twos complement overflow resulted from the operation; cleared otherwise.

**C**  $A7 \cdot B7 + B7 \cdot \overline{R7} + \overline{R7} \cdot A7$

Set if there was a carry from the MSB of the result; cleared otherwise.

**Source Forms:** ABA

**Addressing Modes, Machine Code, and Cycle-by-Cycle Execution:**

Cycle	ABA (INH)		
	Addr	Data	R/W
1	OP	1B	1
2	OP+1	—	1



# ABX

Add Accumulator B to Index Register X

# ABX

**Operation:**  $IX \leftarrow (IX) + (ACCB)$

**Description:** Adds the 8-bit unsigned contents of accumulator B to the contents of index register X (IX) considering the possible carry out of the low-order byte of the index register X; places the result in index register X (IX). Accumulator B is not changed. There is no equivalent instruction to add accumulator A to an index register.

**Condition Codes and Boolean Formulae:**

S	X	H	I	N	Z	V	C
—	—	—	—	—	—	—	—

None affected

**Source Forms:** ABX

**Addressing Modes, Machine Code, and Cycle-by-Cycle Execution:**

Cycle	ABX (INH)		
	Addr	Data	R/W
1	OP	SA	1
2	OP+1	—	1
3	FFFF	—	1

# ABY

Add Accumulator B to Index Register Y

# ABY

**Operation:**  $Y \leftarrow (Y) + (ACCB)$

**Description:** Adds the 8-bit unsigned contents of accumulator B to the contents of index register Y (IY) considering the possible carry out of the low-order byte of index register Y; places the result in index register Y (IY). Accumulator B is not changed. There is no equivalent instruction to add accumulator A to an index register.

**Condition Codes and Boolean Formulae:**

S	X	H	I	N	Z	V	C
—	—	—	—	—	—	—	—

None affected

**Source Forms:** ABY

**Addressing Modes, Machine Code, and Cycle-by-Cycle Execution:**

Cycle	ABY (INH)		
	Addr	Data	R/W
1	OP	18	1
2	OP+1	3A	1
3	OP+2	—	1
4	FFFF	—	1

Table 3-2. Instruction Set (1 of 7)

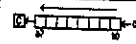
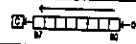
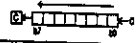
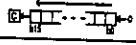
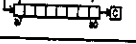
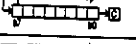
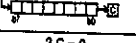
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes									
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C		
ABA	Add Accumulators	$A + B \Rightarrow A$	INH	1B	—	2	—	—	Δ	—	—	Δ	Δ	Δ	Δ	
ABX	Add B to X	$IX + (00 : B) \Rightarrow IX$	INH	3A	—	3	—	—	—	—	—	—	—	—	—	
ABY	Add B to Y	$IY + (00 : B) \Rightarrow IY$	INH	1B 3A	—	4	—	—	—	—	—	—	—	—	—	
ADCA (opr)	Add with Carry to A	$A + M + C \Rightarrow A$	A A A A A	IMM DIR EXT IND,X IND,Y	89 99 89 A9 A9	 dd hh    ff ff ff	2 3 4 4 5	—	—	Δ	—	—	Δ	Δ	Δ	Δ
ADCB (opr)	Add with Carry to B	$B + M + C \Rightarrow B$	B B B B B	IMM DIR EXT IND,X IND,Y	C9 D9 F9 E9 E9	 dd hh    ff ff ff	2 3 4 4 5	—	—	Δ	—	—	Δ	Δ	Δ	Δ
ADDA (opr)	Add Memory to A	$A + M \Rightarrow A$	A A A A A	IMM DIR EXT IND,X IND,Y	8B 9B 8B AB AB	 dd hh    ff ff ff	2 3 4 4 5	—	—	Δ	—	—	Δ	Δ	Δ	Δ
ADDB (opr)	Add Memory to B	$B + M \Rightarrow B$	B B B B B	IMM DIR EXT IND,X IND,Y	C8 D8 F8 E8 E8	 dd hh    ff ff ff	2 3 4 4 5	—	—	Δ	—	—	Δ	Δ	Δ	Δ
ADDD (opr)	Add 16-Bit to D	$D + (M, M + 1) \Rightarrow D$		IMM DIR EXT IND,X IND,Y	C3 D3 F3 E3 E3	 dd hh    ff ff ff	4 5 6 6 7	—	—	—	—	—	Δ	Δ	Δ	Δ
ANDA (opr)	AND A with Memory	$A * M \Rightarrow A$	A A A A A	IMM DIR EXT IND,X IND,Y	84 94 B4 A4 A4	 dd hh    ff ff ff	2 3 4 4 5	—	—	—	—	—	Δ	Δ	0	—
ANDB (opr)	AND B with Memory	$B * M \Rightarrow B$	B B B B B	IMM DIR EXT IND,X IND,Y	C4 D4 F4 E4 E4	 dd hh    ff ff ff	2 3 4 4 5	—	—	—	—	—	Δ	Δ	0	—
ASL (opr)	Arithmetic Shift Left			EXT IND,X IND,Y	78 88 68	hh    ff ff	6 6 7	—	—	—	—	—	Δ	Δ	Δ	Δ
ASLA	Arithmetic Shift Left A		A	INH	48	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		B	INH	58	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D			INH	05	—	3	—	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic Shift Right			EXT IND,X IND,Y	77 67 67	hh    ff ff	6 6 7	—	—	—	—	—	Δ	Δ	Δ	Δ
ASRA	Arithmetic Shift Right A		A	INH	47	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic Shift Right B		B	INH	57	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	r	—	3	—	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear Bit(s)	$M * (mm) \Rightarrow M$	DIR IND,X IND,Y	15 1D 1D	dd ff ff	mm mn mn	6 7 8	—	—	—	—	—	Δ	Δ	0	—
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	r	—	3	—	—	—	—	—	—	—	—	—

Table 3-2. Instruction Set (2 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes										
				Opcode	Operand	Cycles	B	X	H	I	N	Z	V	C			
BEO (rel)	Branch if = Zero	? Z = 1	REL	27	r	3	---	---	---	---	---	---	---	---	---	---	---
BGE (rel)	Branch if ≥ Zero	? N ⊕ V = 0	REL	2C	r	3	---	---	---	---	---	---	---	---	---	---	---
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	r	3	---	---	---	---	---	---	---	---	---	---	---
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	r	3	---	---	---	---	---	---	---	---	---	---	---
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	r	3	---	---	---	---	---	---	---	---	---	---	---
BITA (opr)	Bit(s) Test A with Memory	A + M	A IMM A DIR A EXT A IND,X A IND,Y	85 95 B5 A5 18 A5	ll dd hh # #	2 3 4 4 5	---	---	---	---	Δ	Δ	0	---	---	---	---
BITB (opr)	Bit(s) Test B with Memory	B + M	B IMM B DIR B EXT B IND,X B IND,Y	85 95 B5 E5 18 E5	ll dd hh # #	2 3 4 4 5	---	---	---	---	Δ	Δ	0	---	---	---	---
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ V) = 1	REL	2F	r	3	---	---	---	---	---	---	---	---	---	---	---
BLO (rel)	Branch if Lower	? C = 1	REL	25	r	3	---	---	---	---	---	---	---	---	---	---	---
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	r	3	---	---	---	---	---	---	---	---	---	---	---
BLT (rel)	Branch if < Zero	? N ⊕ V = 1	REL	2D	r	3	---	---	---	---	---	---	---	---	---	---	---
BMF (rel)	Branch if Minus	? N = 1	REL	2B	r	3	---	---	---	---	---	---	---	---	---	---	---
BNE (rel)	Branch if not = Zero	? Z = 0	REL	26	r	3	---	---	---	---	---	---	---	---	---	---	---
BPL (rel)	Branch if Plus	? N = 0	REL	2A	r	3	---	---	---	---	---	---	---	---	---	---	---
BRA (rel)	Branch Always	? 1 = 1	REL	20	r	3	---	---	---	---	---	---	---	---	---	---	---
BRCLR(opr) (mask) (rel)	Branch if Bit(s) Clear	? M + mm = 0	DIR IND,X IND,Y	13 1F 18 1F	dd mm rr # mm rr # mm rr	6 7 8	---	---	---	---	---	---	---	---	---	---	---
BRN (rel)	Branch Never	? 1 = 0	REL	21	r	3	---	---	---	---	---	---	---	---	---	---	---
BRSET(opr) (mask) (rel)	Branch if Bit(s) Set	? (M) + mm = 0	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr # mm rr # mm rr	6 7 8	---	---	---	---	---	---	---	---	---	---	---
BSET (opr) (mask)	Set Bit(s)	M + mm = M	DIR IND,X IND,Y	14 1C 18 1C	dd mm # mm # mm	6 7 8	---	---	---	---	Δ	Δ	0	---	---	---	---
BSR (rel)	Branch to Subroutine	See Figure 3-2	REL	8D	r	6	---	---	---	---	---	---	---	---	---	---	---
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	r	3	---	---	---	---	---	---	---	---	---	---	---
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	r	3	---	---	---	---	---	---	---	---	---	---	---
CBA	Compare A to B	A - B	INH	11	---	2	---	---	---	---	Δ	Δ	Δ	Δ	---	---	---
CLC	Clear Carry Bit	0 ⇒ C	INH	0C	---	2	---	---	---	---	---	---	---	---	---	---	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	---	2	---	---	0	---	---	---	---	---	---	---	---
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT IND,X IND,Y	7F 6F 18 6F	hh # # # # #	6 7 7	---	---	---	---	0	1	0	0	---	---	---

Table 3-2. Instruction Set (3 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes									
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C		
CLRA	Clear Accumulator A	$0 \Rightarrow A$	A INH	4F	—	2	—	—	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	$0 \Rightarrow B$	B INH	5F	—	2	—	—	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	$0 \Rightarrow V$	INH	0A	—	2	—	—	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A to Memory	$A - M$	A IMM A DIR A EXT A IND,X A IND,Y	81 91 B1 A1 A1	ii dd hh # ff # ff	2 3 4 4 5	—	—	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B to Memory	$B - M$	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 E1	ii dd hh # ff # ff	2 3 4 4 5	—	—	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	One's Complement Memory Byte	$\$FF - M \Rightarrow M$	EXT IND,X IND,Y	73 63 63	hh # ff # ff	5 6 7	—	—	—	—	—	—	Δ	Δ	0	1
COMA	One's Complement A	$\$FF - A \Rightarrow A$	A INH	43	—	2	—	—	—	—	—	—	Δ	Δ	0	1
COMB	One's Complement B	$\$FF - B \Rightarrow B$	B INH	53	—	2	—	—	—	—	—	—	Δ	Δ	0	1
CPD (opr)	Compare D to Memory 16-Bit	$D - M : M + 1$	IMM DIR EXT IND,X IND,Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd # hh # ff # ff	5 6 7 7 7	—	—	—	—	—	—	Δ	Δ	Δ	Δ
CPX (opr)	Compare X to Memory 16-Bit	$IX - M : M + 1$	IMM DIR EXT IND,X IND,Y	8C 9C BC AC AC	jj kk dd # hh # ff # ff	4 5 6 6 7	—	—	—	—	—	—	Δ	Δ	Δ	Δ
CPY (opr)	Compare Y to Memory 16-Bit	$IY - M : M + 1$	IMM DIR EXT IND,X IND,Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd # hh # ff # ff	5 6 7 7 7	—	—	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19	—	2	—	—	—	—	—	—	Δ	Δ	Δ	Δ
DEC (opr)	Decrement Memory Byte	$M - 1 \Rightarrow M$	EXT IND,X IND,Y	7A 6A 6A	hh # ff # ff	6 6 7	—	—	—	—	—	—	Δ	Δ	Δ	—
DECA	Decrement Accumulator A	$A - 1 \Rightarrow A$	A INH	4A	—	2	—	—	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement Accumulator B	$B - 1 \Rightarrow B$	B INH	5A	—	2	—	—	—	—	—	—	Δ	Δ	Δ	—
DES	Decrement Stack Pointer	$SP - 1 \Rightarrow SP$	INH	34	—	3	—	—	—	—	—	—	—	—	—	—
DEX	Decrement Index Register X	$IX - 1 \Rightarrow IX$	INH	09	—	3	—	—	—	—	—	—	—	Δ	—	—
DEY	Decrement Index Register Y	$IY - 1 \Rightarrow IY$	INH	18 09	—	4	—	—	—	—	—	—	—	Δ	—	—
EDRA (opr)	Exclusive OR A with Memory	$A \oplus M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	88 98 B8 A8 A8	ii dd hh # ff # ff	2 3 4 4 5	—	—	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with Memory	$B \oplus M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 E8	ii dd hh # ff # ff	2 3 4 4 5	—	—	—	—	—	—	Δ	Δ	0	—

Table 3-2. Instruction Set (4 of 7)

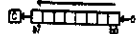


Mnemonic	Operation	Description	Addressing		Instruction			Condition Codes										
			Mode	Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C				
FDIV	Fractional Divide 16 by 16	D / IX $\Rightarrow$ IX; r $\Rightarrow$ D	INH	03	—	41	—	—	—	—	—	—	—	—	—	—	—	—
IDIV	Integer Divide 16 by 16	D / IX $\Rightarrow$ IX; r $\Rightarrow$ D	INH	02	—	41	—	—	—	—	—	—	—	—	—	—	—	—
INC (opr)	Increment Memory Byte	M + 1 $\Rightarrow$ M	EXT IND,X IND,Y	7C 6C 16 6C	hh ff ff	6 6 7	—	—	—	—	—	—	—	—	—	—	—	—
INCA	Increment Accumulator A	A + 1 $\Rightarrow$ A	A INH	4C	—	2	—	—	—	—	—	—	—	—	—	—	—	—
INCB	Increment Accumulator B	B + 1 $\Rightarrow$ B	B INH	5C	—	2	—	—	—	—	—	—	—	—	—	—	—	—
INS	Increment Stack Pointer	SP + 1 $\Rightarrow$ SP	INH	31	—	3	—	—	—	—	—	—	—	—	—	—	—	—
INX	Increment Index Register X	IX + 1 $\Rightarrow$ IX	INH	08	—	3	—	—	—	—	—	—	—	—	—	—	—	—
INY	Increment Index Register Y	IY + 1 $\Rightarrow$ IY	INH	18 08	—	4	—	—	—	—	—	—	—	—	—	—	—	—
JMP (opr)	Jump	See Figure 3-2	EXT IND,X IND,Y	7E 6E 18 6E	hh ff ff	3 4	—	—	—	—	—	—	—	—	—	—	—	—
JSR (opr)	Jump to Subroutine	See Figure 3-2	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd ff ff	5 6 7	—	—	—	—	—	—	—	—	—	—	—	—
LDAA (opr)	Load Accumulator A	M $\Rightarrow$ A	A IMM A DIR A EXT A IND,X A IND,Y	88 98 86 A6 18 A6	ll dd hh ff ff	2 3 4 5	—	—	—	—	—	—	—	—	—	—	—	—
LDAB (opr)	Load Accumulator B	M $\Rightarrow$ B	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 18 E8	ll dd hh ff ff	2 3 4 5	—	—	—	—	—	—	—	—	—	—	—	—
LDD (opr)	Load Double Accumulator D	M $\Rightarrow$ A, M + 1 $\Rightarrow$ B	IMM DIR EXT IND,X IND,Y	CC DC FC 18 EC	ll kk dd hh ff	3 4 5 6	—	—	—	—	—	—	—	—	—	—	—	—
LDS (opr)	Load Stack Pointer	M : M + 1 $\Rightarrow$ SP	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	ll kk dd hh ff	3 4 5 6	—	—	—	—	—	—	—	—	—	—	—	—
LDX (opr)	Load Index Register X	M : M + 1 $\Rightarrow$ IX	IMM DIR EXT IND,X IND,Y	CE DE FE 18 EE	ll kk dd hh ff	3 4 5 6	—	—	—	—	—	—	—	—	—	—	—	—
LDY (opr)	Load Index Register Y	M : M + 1 $\Rightarrow$ IY	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	ll kk dd hh ff	4 5 6 6 6	—	—	—	—	—	—	—	—	—	—	—	—
LSL (opr)	Logical Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ff ff	6 7	—	—	—	—	—	—	—	—	—	—	—	—
LSLA	Logical Shift Left A		A INH	48	—	2	—	—	—	—	—	—	—	—	—	—	—	—
LSLB	Logical Shift Left B		B INH	58	—	2	—	—	—	—	—	—	—	—	—	—	—	—

Table 3-2. Instruction Set (5 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes										
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C			
LSLD	Logical Shift Left Double		INH	05	—	3	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical Shift Right		EXT IND,X IND,Y	74 64 64	hh ff ff	6 6 7	—	—	—	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical Shift Right A		A INH	44	—	2	—	—	—	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		B INH	54	—	2	—	—	—	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical Shift Right Double		INH	04	—	3	—	—	—	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply B by B	$A + B \Rightarrow D$	INH	3D	—	10	—	—	—	—	—	—	—	—	—	—	Δ
NEG (opr)	Two's Complement Memory Byte	$0 - M \Rightarrow M$	EXT IND,X IND,Y	70 60 60	hh ff ff	6 6 7	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Two's Complement A	$0 - A \Rightarrow A$	A INH	40	—	2	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Two's Complement B	$0 - B \Rightarrow B$	B INH	50	—	2	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—	—	—	—
ORAA (opr)	OR Accumulator A (Inclusive)	$A + M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A 8A AA AA	ij dd hh ff ff	2 3 4 4 5	—	—	—	—	—	—	—	Δ	Δ	0	—
ORAB (opr)	OR Accumulator B (Inclusive)	$B + M \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA EA	ij dd hh ff ff	2 3 4 4 5	—	—	—	—	—	—	—	Δ	Δ	0	—
PSHA	Push A onto Stack	$A \downarrow \text{Stk}, SP = SP - 1$	A INH	36	—	3	—	—	—	—	—	—	—	—	—	—	—
PSHB	Push B onto Stack	$B \downarrow \text{Stk}, SP = SP - 1$	B INH	37	—	3	—	—	—	—	—	—	—	—	—	—	—
PSHX	Push X onto Stack (Lo First)	$IX \downarrow \text{Stk}, SP = SP - 2$	INH	3C	—	4	—	—	—	—	—	—	—	—	—	—	—
PSHY	Push Y onto Stack (Lo First)	$IY \downarrow \text{Stk}, SP = SP - 2$	INH	18 3C	—	5	—	—	—	—	—	—	—	—	—	—	—
PULA	Pull A from Stack	$SP = SP + 1, A \uparrow \text{Stk}$	A INH	32	—	4	—	—	—	—	—	—	—	—	—	—	—
PULB	Pull B from Stack	$SP = SP + 1, B \uparrow \text{Stk}$	B INH	33	—	4	—	—	—	—	—	—	—	—	—	—	—
PULX	Pull X From Stack (Hi First)	$SP = SP + 2, IX \uparrow \text{Stk}$	INH	38	—	5	—	—	—	—	—	—	—	—	—	—	—
PULY	Pull Y from Stack (Hi First)	$SP = SP + 2, IY \uparrow \text{Stk}$	INH	18 38	—	6	—	—	—	—	—	—	—	—	—	—	—
ROL (opr)	Rotate Left		EXT IND,X IND,Y	78 68 68	hh ff ff	6 6 7	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
ROLA	Rotate Left A		A INH	49	—	2	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		B INH	59	—	2	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate Right		EXT IND,X IND,Y	76 66 66	hh ff ff	6 6 7	—	—	—	—	—	—	—	Δ	Δ	Δ	Δ

Table 3-2. Instruction Set (6 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes								
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C	
RORA	Rotate Right A		A INH	48	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		B INH	56	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from Interrupt	See Figure 3-2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from Subroutine	See Figure 3-2	INH	38	—	5	—	—	—	—	—	—	—	—	—
SBA	Subtract B from A	$A - B \Rightarrow A$	INH	10	—	2	—	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with Carry from A	$A - M - C \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	82 92 B2 A2 18 A2	ii dd hh ff ff ff	2 3 4 4 5	—	—	—	—	—	Δ	Δ	Δ	Δ
SBCB (opr)	Subtract with Carry from B	$B - M - C \Rightarrow B$	B IMM B DIR B EXT B IND,X B IND,Y	C2 D2 F2 E2 18 E2	ii dd hh ff ff ff	2 3 4 4 5	—	—	—	—	—	Δ	Δ	Δ	Δ
SEC	Set Carry	$1 \Rightarrow C$	INH	0D	—	2	—	—	—	—	—	—	—	—	1
SEI	Set Interrupt Mask	$1 \Rightarrow I$	INH	0F	—	2	—	—	—	1	—	—	—	—	—
SEV	Set Overflow Flag	$1 \Rightarrow V$	INH	0B	—	2	—	—	—	—	—	—	—	1	—
STAA (opr)	Store Accumulator A	$A \Rightarrow M$	A DIR A EXT A IND,X A IND,Y	97 B7 A7 18 A7	dd hh ff ff ff	3 4 4 5	—	—	—	—	—	Δ	Δ	0	—
STAB (opr)	Store Accumulator B	$B \Rightarrow M$	B DIR B EXT B IND,X B IND,Y	D7 F7 E7 18 E7	dd hh ff ff ff	3 4 4 5	—	—	—	—	—	Δ	Δ	0	—
STD (opr)	Store Accumulator D	$A \Rightarrow M, B \Rightarrow M + 1$	DIR EXT IND,X IND,Y	D0 F0 E0 18 E0	dd hh ff ff ff	4 5 5 6	—	—	—	—	—	Δ	Δ	0	—
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—	—
STP (opr)	Store Stack Pointer	$SP \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	3F BF AF 18 AF	dd hh ff ff ff	4 5 5 6	—	—	—	—	—	Δ	Δ	0	—
STX (opr)	Store Index Register X	$IX \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	DF FF EF CD EF	dd hh ff ff ff	4 5 5 6	—	—	—	—	—	Δ	Δ	0	—
STY (opr)	Store Index Register Y	$IY \Rightarrow M : M + 1$	DIR EXT IND,X IND,Y	18 DF 18 FF 1A EF 18 EF	dd hh ff ff ff	5 6 6 6	—	—	—	—	—	Δ	Δ	0	—
SUBA (opr)	Subtract Memory from A	$A - M \Rightarrow A$	A IMM A DIR A EXT A IND,X A IND,Y	80 90 B0 A0 18 A0	ii dd hh ff ff ff	2 3 4 4 5	—	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract Memory from B	$B - M \Rightarrow B$	A IMM A DIR A EXT A IND,X A IND,Y	C0 D0 F0 E0 18 E0	ii dd hh ff ff ff	2 3 4 4 6	—	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract Memory from D	$D - M : M + 1 \Rightarrow D$	IMM DIR EXT IND,X IND,Y	83 93 B3 A3 18 A3	jj kk hh ff ff ff	4 5 6 6 7	—	—	—	—	—	Δ	Δ	Δ	Δ



**Table 3-2. Instruction Set (7 of 7)**

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	N	I	Z	V	C	
SWI	Software Interrupt	See Figures 3-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A = B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	A = CCR	INH	08	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B = A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00	—	*	—	—	—	—	—	—	—	—
TPA	Transfer CC Register to A	CCR = A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opf)	Test for Zero or Minus	M = 0	EXT IND,X IND,Y	7D 6D 6D	hh ff ff ff	8 8 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for Zero or Minus	A = 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	B = 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer Stack Pointer to X	SP + 1 = X	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer Stack Pointer to Y	SP + 1 = Y	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to Stack Pointer	X - 1 = SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to Stack Pointer	Y - 1 = SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E	—	**	—	—	—	—	—	—	—	—
XGDY	Exchange D with X	X = D	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	Y = D	INH	18 8F	—	4	—	—	—	—	—	—	—	—

**Cycle**

- \* Infinity or until reset occurs
- \*\* 12 Cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-Clock cycles (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (14 + n total)

**Operands**

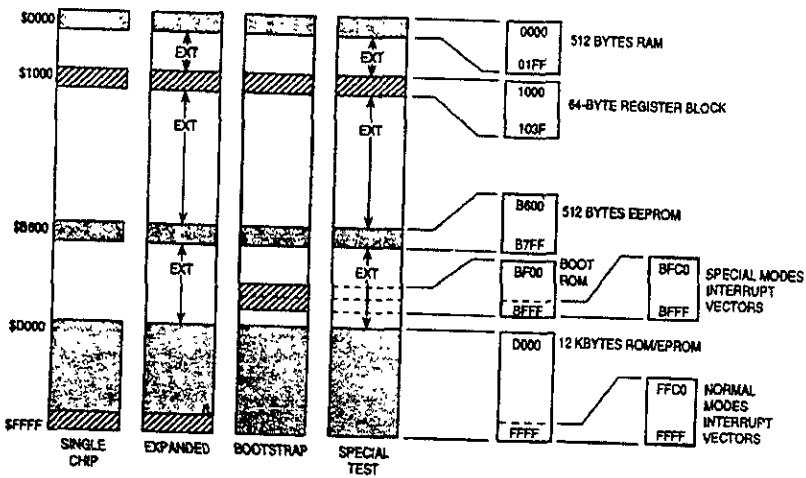
- dd = 8-Bit Direct Address (\$0000-\$00FF) (High Byte Assumed to be 00)
- ff = 8-Bit Positive Offset \$00 (0) to \$FF (255) (Is Added to Index)
- hh = High-Order Byte of 16-Bit Extended Address
- ll = Low-Order Byte of 16-Bit Extended Address
- ll = One Byte of Immediate Data
- ll = High-Order Byte of 16-Bit Immediate Data
- kk = Low-Order Byte of 16-Bit Immediate Data
- ll = Low-Order Byte of 16-Bit Extended Address
- mm = 8-Bit Mask (Set Bits to be Affected)
- r = Signed Relative Offset \$80 (-128) to \$7F (+127) (Offset Relative to Address Following Machine Code Offset Bytes)

**Operators**

- () Contents of register shown inside parentheses
- = is transferred to
- ≡ is exchanged with
- ↑ is pulled from stack
- ↓ is pushed onto stack
- Boolean AND
- + Arithmetic Addition Symbol except where used as Inclusive-OR symbol in Boolean Formula
- ⊕ Exclusive-OR
- Multiply
- Concatenation
- Arithmetic subtraction symbol or Negation symbol (Two's Complement)

**Condition Codes**

- Bit not changed
- 0 Bit always cleared
- 1 Bit always set
- Δ BR cleared or set, depending on operation
- ↓ Bit can be cleared, cannot become set



2 PAGES MEM MAP P1

Figure 4-2. Memory Map for MC68HC11E0, MC68HC11E1, and MC68HC(7)11E9

**Table 4-1. Register and Control Bit Assignments (1 of 2)**

The register block can be remapped to any 4K boundary

	Bit 7	6	5	4	3	2	1	Bit 0	
\$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
\$1001									Reserved
\$1002	STAF	STAI	OWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
\$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
\$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
\$1005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	PORTCL
\$1006									Reserved
\$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
\$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
\$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
\$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
\$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
\$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
\$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
\$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
\$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)
\$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
\$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
\$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
\$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
\$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)
\$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)
\$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (High)
\$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
\$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
\$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
\$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
\$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
\$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
\$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)
\$101E	Bit 15	14	13	12	11	10	9	Bit 8	T14/O5 (High)
\$101F	Bit 7	6	5	4	3	2	1	Bit 0	T14/O5 (Low)
\$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
\$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
\$1022	OC11	OC21	OC31	OC41	I4/O51	IC11	IC21	IC31	TMSK1

**Table 4-1. Register and Control Bit Assignments (2 of 2)**

\$1023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
\$1024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
\$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
\$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL
\$1027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT
\$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
\$1029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
\$102A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
\$102B	TCLR	SCP2 <sup>1</sup>	SCP1	SCP0	rCKB	SCR2	SCR1	SCR0	BAUD
\$102C	RB	TB	0	M	WAKE	0	0	0	SCCR1
\$102D	TIE	TCIE	RIE	ILIE	TE	RE	RMU	SBK	SCCR2
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
\$102F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDR
\$1030	COF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
\$1031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
\$1032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
\$1033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
\$1034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4
\$1035	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
\$1036	MBE	0	ELAT	EXCOL	EXROW	T1	T0	PGM	EPROG <sup>2</sup>
\$1037									Reserved
\$1038									Reserved
\$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
\$103A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
\$103B	ODD	EVEN	ELAT <sup>3</sup>	BYTE	ROW	ERASE	EELAT	EPGM	PPROG
\$103C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
\$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
\$103E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
\$103F	EE3 <sup>4</sup>	EE2 <sup>4</sup>	EE1 <sup>4</sup>	EE0 <sup>4</sup>	NOSEC	NOCOP	ROMON	EEON	CONFIG

**NOTES:**

1. SCP2 adds +39 to SCI prescaler and is present only in MC68HC(7)11E20.
2. EPROG is present only in MC68HC711E20.
3. ELAT in PPROG is present only on MC68HC711E9 and MC68S711E9.
4. EEPROM mapping control bits EE[3:0] apply only to MC68HC811E2.

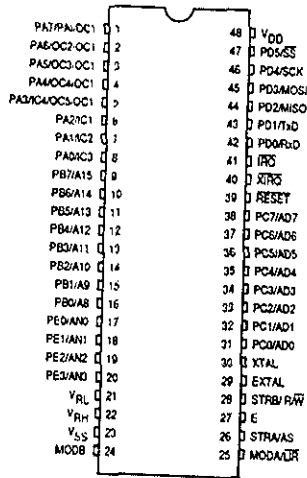


Figure 2-4. MC68HC811E2 Pin Assignments (48-Pin DIP)

# Anexo B

## Registros de la interface de comunicación serie (SCI)

All six bits of port D are affected by the port D wired-OR mode control bit (DWOM in the SPCR). Whenever DWOM is one, the high-side driver (P-channel device) for all port D pins is disabled. This disabling makes port D pins behave somewhat like open-collector outputs; thus, an external pullup resistor is needed for any port D pin being used as an output (general-purpose or peripheral subsystem outputs). The DWOM bit does not affect the use of port D pins as inputs.

### 9.2.2 Baud-Rate Control Register (BAUD)

The following register and paragraphs describe the BAUD control register, which is used to set the bit rate for the SCI system. Normally, this register is written once during initialization to set the baud rate for SCI communications. Both the receiver and the transmitter use the same baud rate, which is derived from the MCU bus rate clock. A two-stage divider is used to develop customary baud rates from normal MCU crystal frequencies; therefore, it is not necessary to use special baud-rate crystal frequencies. Tables 9-1 and 9-2 should be adequate for most users, but a more comprehensive tabulation of baud rates is provided in Table 9-3 to help users with unusual requirements.

7	6	5	4	3	2	1	0	
TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD \$1028
RESET.	0	0	0	0	U	U	U	

#### SCP1–SCP0 — SCI Baud-Rate Prescale Selects

These two bits select a prescale factor for the SCI baud-rate generator. The output frequency of this prescaler determines the highest available baud rate in the system. The actual  $16 \times$  baud rate will be a binary submultiple ( $-1, -2, -4, \dots -128$ ) of this prescaler output as selected by the SCR2–SCR0 bits. Table 9-1 shows the highest baud rates that result for various combinations of crystal frequency and prescaler select control bit values. The actual frequency coming out of this prescaler stage is 16 times the baud rates listed. Since not all of these entries are of practical use, the ones of interest have been highlighted in bold. This prescaler is set to its fastest rate by default out of reset and may be changed at any time. Of course, it makes no sense to change the baud rate while any SCI transfer is in progress.

**Table 9-1. Baud-Rate Prescale Selects**

SCP1	SCP0	Division Factor	Crystal Frequency				
			2 <sup>23</sup> Hz	8 MHz	4.9152 MHz	4 MHz	3.6864 MHz
			Highest Baud Rate				
0	0	1	<b>131.072K Baud</b>	125.000K Baud	<b>76.80K Baud</b>	62.50K Baud	57.60K Baud
0	1	3	<b>43.691K Baud</b>	41.667K Baud	<b>25.60K Baud</b>	20.833K Baud	<b>19.20K Baud</b>
1	0	4	<b>32.768K Baud</b>	31.250K Baud	<b>19.20K Baud</b>	15.625K Baud	14.40K Baud
1	1	13	<b>10.082K Baud</b>	<b>9600 Baud</b>	<b>5.908K Baud</b>	<b>4800 Baud</b>	<b>4431K Baud</b>
			2.1 MHz	2 MHz	1.2288 MHz	1 MHz	921.6 kHz
Bus Frequency (E clock)							

**SCR2-SCR0 — SCI Baud-Rate Selects**

These three bits are used in conjunction with the SCI prescaler bits (see Table 9-1) to specify the SCI baud rate. The prescale bits, SCP1-SCP0, determine the highest baud rate; whereas, the SCR2-SCR0 bits select an additional binary submultiple ( $\div 1, -2, -4, \dots -128$ ) of this highest baud rate. The result of these two dividers working in series is the  $16 \times$  receiver baud-rate clock. Table 9-2 shows the SCI baud rates that result for various settings of SCR2-SCR0 and the highest baud rates from Table 9-1. Since not all of these entries are of practical use, the ones of interest have been highlighted in bold. The SCR2-SCR0 bits are not affected by reset and may be changed at any time. Of course, it makes no sense to change the baud rate while any SCI transfer is in progress.

**Table 9-2. Baud-Rate Selects**

SCR2	SCR1	SCR0	Division Factor	Highest Baud Rate (from Table 9-1)				
				131.072K Baud	32.768K Baud	76.80K Baud	19.20K Baud	9600 Baud
				SCI Baud Rate				
0	0	0	1	<b>131.072K Baud</b>	<b>32.768K Baud</b>	76.80K Baud	19.20K Baud	<b>9600 Baud</b>
0	0	1	2	65.536K Baud	16.384K Baud	38.40K Baud	<b>9600 Baud</b>	<b>4800 Baud</b>
0	1	0	4	<b>32.768K Baud</b>	<b>8192 Baud</b>	19.20K Baud	<b>4800 Baud</b>	<b>2400 Baud</b>
0	1	1	8	16.384K Baud	4096 Baud	<b>9600 Baud</b>	<b>2400 Baud</b>	<b>1200 Baud</b>
1	0	0	16	<b>8192 Baud</b>	2048 Baud	<b>4800 Baud</b>	<b>1200 Baud</b>	<b>600 Baud</b>
1	0	1	32	4096 Baud	1024 Baud	<b>2400 Baud</b>	<b>600 Baud</b>	<b>300 Baud</b>
1	1	0	64	2048 Baud	512 Baud	<b>1200 Baud</b>	<b>300 Baud</b>	<b>150 Baud</b>
1	1	1	128	1024 Baud	256 Baud	600 Baud	150 Baud	75 Baud

**TCLR — Clear Baud-Rate Timing Chain (Test Modes Only)**

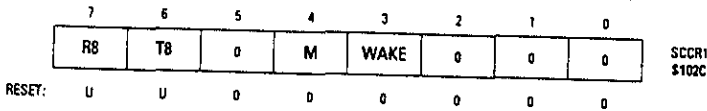
This bit is disabled and remains low in any mode other than test or bootstrap modes. Reset clears this bit. While in test or bootstrap modes, writing a one to this bit causes the baud-rate counter chains to be reset. Because the one state of this bit is transitory, reads always return a logic zero. This control bit is intended only for factory testing of the MCU.

**RCKB — SCI Baud-Rate Clock Test (Test Modes Only)**

This bit is disabled and remains low in any mode other than test or bootstrap modes. Reset clears this bit. While in test or bootstrap modes, this bit may be written but not read (reads always return a logic zero). Writing a one to this bit enables a baud-rate counter test mode where the exclusive-OR of the receiver clock ( $16 \times$  the baud rate) and the transmit clock ( $1 \times$  the baud rate) is driven out the PD1/TxD pin. This control bit is intended only for factory testing of the MCU.

### 9.2.3 SCI Control Register 1 (SCCR1)

The SCCR1 contains control bits related to the 9-bit data character format and the receiver wake-up feature. Four of the bits in this register are not used and always read as zeros.



#### R8 — Receive Data Bit 8

When the SCI system is configured for 9-bit data characters, this bit acts as an extra (ninth) bit of the RDR. The MSB of received characters is transferred into this bit at the same time the remaining eight bits are transferred from the serial receive shifter to the SCDR.

#### T8 — Transmit Data Bit 8

When the SCI system is configured for 9-bit data characters, this bit acts as the extra (ninth) bit of the TDR. When the low-order eight bits of a transmit character are transferred from the SCDR to the serial transmit shift register, this bit is transferred to the ninth bit position of the shifter. In cases where the sense of this bit is the same as it was for the previous character, it is not necessary to write to this bit before transmission of the new character. For example, if the 9-bit format is used to get an extra stop bit (logic one), the T8 bit is written to one before transmitting the first character, and no other writes are needed.

#### M — SCI Character Length

0 = One start bit, eight data bits, one stop bit

1 = One start bit, nine data bits, one stop bit

The M bit controls the character length for both the transmitter and receiver at the same time. The 9-bit data format is most commonly used for an extra stop bit or in conjunction with the address-mark wake-up method, but it can also be used for parity. Mark and space parity are trivial, but odd and even parity require software calculations.

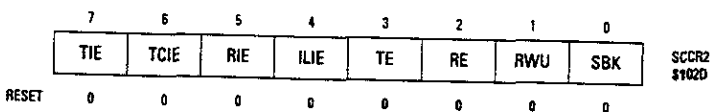
#### WAKE — Wake-Up Method Select

0 = Idle line; detection of at least a full character time of idle line causes the receiver to wake up.

1 = Address mark; a logic one in the MSB position (eighth or ninth data bit depending on character size selected by M bit) causes the receiver to wake up.

### 9.2.4 SCI Control Register 2 (SCCR2)

The SCCR2 is the main control register for the SCI subsystem.





Although it is possible for software to write the RWU bit to zero, is very unusual to do so. The normal sequence is for software to set the RWU bit after deciding that a particular SCI message is of no interest. Setting the RWU bit causes the receiver to go to sleep (ignore further receiver interrupt sources) until the start of the next message. Receiver wake-up logic recognizes when the unimportant message is over and automatically clears the RWU bit to wake up the sleeping receiver.

**SBK — Send Break**

0 = Normal transmitter operation.

1 = Enable transmitter to send synchronous break characters.

Whenever the SBK bit is written to one, at least one character time of break will be queued and sent. In the context of the M68HC11, a break character causes the TxD line to go to logic zero for 10 (11 if M = 1) bit times.

In old teletype systems, a break was caused by simply disconnecting the serial line, which caused the line to go to logic zero for some asynchronous length of time (usually as long as the break key was pressed). A receiver seeing a break character produced by an M68HC11 would receive an all-zero character with a framing error (FE) because the line would be low where the receiver expected to see a logic-high stop bit.

The break characters in the M68HC11 are synchronous because no partial character times of the break condition are ever produced. When SBK is set to one, a break character is queued pending completion of any character currently shifting out of the transmit shift register. When the transmit shift register becomes available, the queued break character is jammed into the shift register to be serially sent, and, if the SBK bit is still one, another break is queued. It will always be at least one character time from when the SBK bit is written back to zero before the transmitter can resume sending normal characters.

**9.2.5 SCI Status Register (SCSR)**

The seven status bits associated with the SCI system are located in the SCSR, which is depicted in the following register. Some of these bits optionally generate hardware interrupt requests; whereas, others simply indicate errors in the reception of a character. These status bits are automatically set by the corresponding conditions having been met in the SCI logic. Once set, these bits remain set until software completes a clearing sequence. The clearing sequences are somewhat automatic in that they are satisfied by performing functions normally done anyway. For example, to clear the TDRE flag, software must read the SCSR while TDRE is set, and then write to the TDR. Since these are exactly the normal steps in response to the TDRE, no instructions are needed to clear the flag.

	7	6	5	4	3	2	1	0	
	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR2 \$102E
RESET:	1	1	0	0	0	0	0	0	

**TDRE — Transmit Data Register Empty**

0 = Not empty; a character was previously written to the SCDR and has not yet transferred to the transmit shift register to be serially sent.

1 = Indicates a new character may now be written to the SCDR.

In normal transmit operations, this bit is checked before each new character is sent to see if the SCDR can accept the new data. The SCI transmitter is double buffered so the TDR holds the second character in line while the transmit serial shift register holds the character in the process of being transmitted serially.

The TDRE flag is cleared by reading SCSR, followed by a write to the SCDR. TDRE must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

The TDRE bit is set to one during reset to indicate that there is no meaningful data in the SCDR.

#### TC — Transmit Complete

0 = The transmitter is busy sending a character, preamble, or break character.

1 = The transmitter has completed sending and has reached an idle state.

This bit is useful in systems where the SCI is driving a modem. When TC is set at the end of a transmission, the modem can be disabled. In older ACIA and SCI systems, the TDRE status bit was the only indication that a transmission was near completion. Since TDRE only indicated that the last character had transferred to the transmit shift register, software had to delay an amount of time greater than or equal to the time it took for this last character to finish transmitting serially. Since the delay time depended on the baud rate, it was relatively difficult to know when it was safe to disable the modem. The TC bit on the M68HC11 offers a much more convenient way to tell when the transmitter has completed sending.

The TC flag is cleared by reading SCSR, followed by a write to the SCDR. TC must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

The TC bit is set to one during reset to indicate that the transmitter is not busy transmitting anything.

#### RDRF — Receive Data Register Full

0 = Not full; nothing has been received since the last character was read out of the SCDR.

1 = A character has been received and has transferred from the receive shift register to the parallel SCDR where software can read it.

This is the normal indication that a character has been received by the SCI. The NF and FE status bits provide additional information about this normally received character in the SCDR. If set, the OR flag would indicate that another character was serially received and was ready to be transferred to the SCDR, but the previously received character was not yet read. Software should check RDRF, OR, NF, and FE to tell if there was any error in the reception of a character (RDRF should be set and the other three flags should be clear).

The RDRF flag is cleared by reading SCSR, followed by a read of the SCDR. RDRF must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied. Since the NF and FE flags are set at the same time as RDRF, the clearing sequence for RDRF will also clear NF and FE if they were set for this received character.

#### IDLE — Idle-Line Detect

0 = The RxD line is either active now or has never been active since IDLE was last cleared.

1 = The RxD line has become idle.

The idle condition is defined as at least a full character time of logic one on the RxD line. A character time is 10 bit times if  $M=0$  or 11 bit times if  $M=1$ .

The IDLE flag is cleared by reading SCSR, followed by a read of the SCDR. IDLE must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

Once IDLE has been cleared, it cannot be set again until the RxD line has been active and then becomes idle again. The idle-line function is inhibited while the receiver wake-up function is enabled ( $RWU=1$ ).

#### OR — Overrun Error

0 = No overrun error.

1 = Indicates that another character was serially received and was ready to be transferred to the SCDR, but the previously received character was not yet read.

Since the SCI receiver is double buffered, there is a full character time between reception of a character and when it must be read from the SCDR to avoid an OR caused by a subsequent character. In an OR condition, the character that caused the OR is lost, but the previously received character in the SCDR is not disturbed.

The NF and FE status bits are associated with the normally received character in the SCDR (never the character that caused an OR).

The OR flag is cleared by reading SCSR, followed by a read of the SCDR. OR must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

#### NF — Noise Flag

0 = No noise detected during reception of the character in the SCDR.

1 = Data recovery logic detected noise during reception of the character in the SCDR.

NF does not generate interrupt requests because it is always associated with the setting of RDRF, which does cause interrupts. Even if the noise is detected early in the reception of the character, NF is not set until RDRF is set at the end of reception of a character. Perceived noise in any of the data bit times and the stop bit or stop bit times will cause NF to be set. During data bit times and the stop bit time, noise is indicated if the three samples taken near the middle of the bit time do not unanimously agree. During reception of the start bit, four additional samples are taken during the first half of the bit time to detect the leading edge of the bit time and to verify that it is a start bit. If all seven samples taken during the start bit time are not zero, noise is indicated and NF is set. Many systems ignore the NF because the data recovery logic has already made a good first-order attempt to correct the problem. In critical applications, the NF could be used to generate a request for retransmission of the questionable data.

The NF flag is cleared by reading SCSR followed by a read of the SCDR. NF must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

**FE — Framing Error**

0 = No framing error detected.

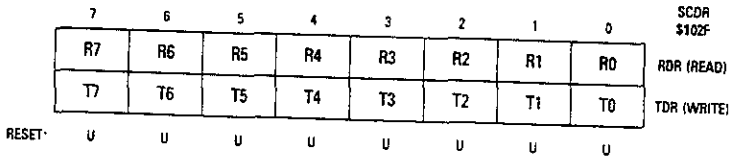
1 = A framing error was detected for the character in the SCDR.

Asynchronous serial data reception requires the receiver to properly align the character reception frame with the incoming serial data. This alignment is achieved by asynchronously searching for the falling edge of the start bit; alignment is verified by looking for the expected logic high during the last bit time (stop bit) of the character. If a logic zero is detected where the stop bit was expected, the FE flag is set. The FE indicator is not a foolproof indication of improper framing. It is possible for the receiver to be misframed without there being any FE indication because the RxD line could by chance be high when the receiver expected to see the stop bit.

The FE flag is cleared by reading SCSR followed by a read of the SCDR. FE must be read as one during the read of SCSR, or the first step of the clearing sequence is not satisfied.

**9.2.6 SCI Data Register (SCDR)**

The SCDR shown in the following register is actually two separate registers. When SCDR is read, the read-only RDR is accessed; when SCDR is written, the write-only TDR is accessed. In discussions of the SCI system, any of the mnemonics SCDR, TDR, or RDR might be used to refer to this register location.



**Table 9-3. Baud Rates by Crystal Frequency, SCP1-SCP0 and SCR2-SCR0**

SCP1	SCP0	SCR2	SCR1	SCR0	Crystal Frequency				
					2 <sup>23</sup> Hz	8 MHz	4.9152 MHz	4 MHz	3.6864 MHz
					Baud Rates				
0	0	0	0	0	131.072K Baud	126.00K Baud	76.80K Baud	62.50K Baud	57.60K Baud
0	0	0	0	1	65.536K Baud	62.60K Baud	38.40K Baud	31.25K Baud	28.80K Baud
0	0	0	1	0	32.768K Baud	31.25K Baud	19.20K Baud	15.625K Baud	14.40K Baud
0	0	0	1	1	16.384K Baud	15.625K Baud	9600 Baud	7812.5 Baud	7200 Baud
0	0	1	0	0	8192 Baud	7812.5 Baud	4800 Baud	3906 Baud	3600 Baud
0	0	1	0	1	4096 Baud	3906 Baud	2400 Baud	1953 Baud	1800 Baud
0	0	1	1	0	2048 Baud	1953 Baud	1200 Baud	977 Baud	900 Baud
0	0	1	1	1	1024 Baud	977 Baud	600 Baud	488 Baud	450 Baud
0	1	0	0	0	43.691K Baud	41.666K Baud	25.60K Baud	20.833K Baud	19.20K Baud
0	1	0	0	1	21.845K Baud	20.833K Baud	12.80K Baud	10.417K Baud	9600 Baud
0	1	0	1	0	10.923K Baud	10.417K Baud	6400 Baud	5208 Baud	4800 Baud
0	1	0	1	1	5461 Baud	5208 Baud	3200 Baud	2604 Baud	2400 Baud
0	1	1	0	0	2731 Baud	2604 Baud	1600 Baud	1302 Baud	1200 Baud
0	1	1	0	1	1365 Baud	1302 Baud	800 Baud	651 Baud	600 Baud
0	1	1	1	0	683 Baud	651 Baud	400 Baud	326 Baud	300 Baud
0	1	1	1	1	341 Baud	326 Baud	200 Baud	163 Baud	150 Baud
1	0	0	0	0	32.768K Baud	31.250K Baud	19.20K Baud	15.625K Baud	14.40K Baud
1	0	0	0	1	16.384K Baud	15.625K Baud	9600 Baud	7812.5 Baud	7200 Baud
1	0	0	1	0	8192 Baud	7812.5 Baud	4800 Baud	3906 Baud	3600 Baud
1	0	0	1	1	4096 Baud	3906 Baud	2400 Baud	1953 Baud	1800 Baud
1	0	1	0	0	2048 Baud	1953 Baud	1200 Baud	977 Baud	900 Baud
1	0	1	0	1	1024 Baud	977 Baud	600 Baud	488 Baud	450 Baud
1	0	1	1	0	512 Baud	488 Baud	300 Baud	244 Baud	225 Baud
1	0	1	1	1	256 Baud	244 Baud	150 Baud	122 Baud	112.5 Baud
1	1	0	0	0	10.082K Baud	9600 (+0.16%)	5908 Baud	4800 (+0.16%)	4431 Baud
1	1	0	0	1	5041 Baud	4800 Baud	2954 Baud	2400 Baud	2215 Baud
1	1	0	1	0	2521 Baud	2400 Baud	1477 Baud	1200 Baud	1108 Baud
1	1	0	1	1	1260 Baud	1200 Baud	738 Baud	600 Baud	554 Baud
1	1	1	0	0	630 Baud	600 Baud	369 Baud	300 Baud	277 Baud
1	1	1	0	1	315 Baud	300 Baud	185 Baud	150 Baud	138 Baud
1	1	1	1	0	158 Baud	150 Baud	92 Baud	75 Baud	69 Baud
1	1	1	1	1	79 Baud	75 Baud	46 Baud	38 Baud	35 Baud
					2.1 MHz	2 MHz	1.2288 MHz	1 MHz	921.6 kHz
Bus Frequency (E clock)									

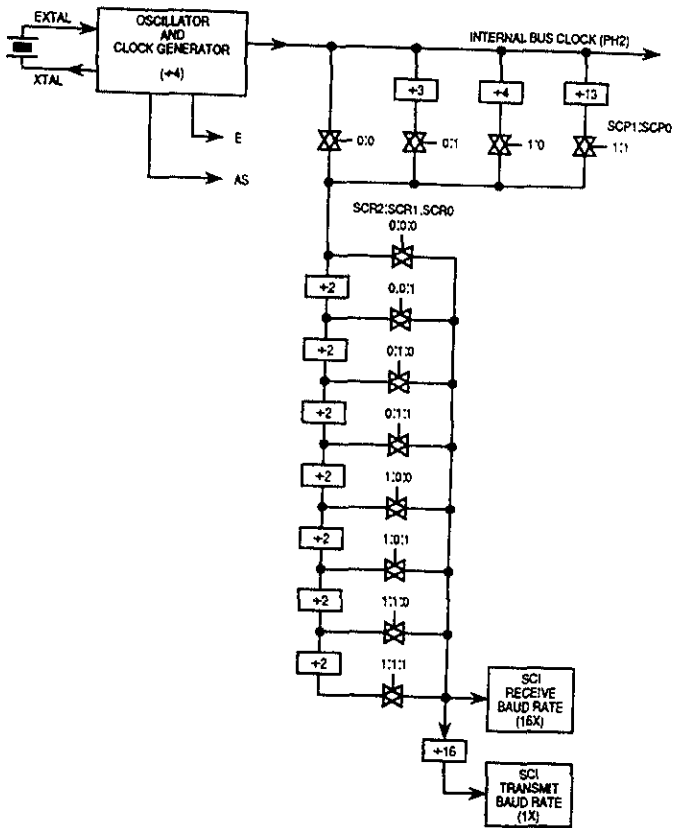


Figure 9-12. Baud-Rate Generator Block Diagram

# Anexo C

## SCORBOT-ER III

Con los desarrollos logrados en robótica y el aumento de las instalaciones de robots en la industria, hay una creciente demanda de conocimientos sobre el funcionamiento y mantenimiento de los robots. Es por esta razón que fue creado el robot Scorbot, el cual permite mediante su programa exclusivo, manejar sistemas de robots, simular aplicaciones industriales, aprender a mantenerlos, y además, construir células de trabajo robóticas completas.

El robot Scorbot consta de dos componentes principales, el brazo mecánico y el controlador electrónico.

El brazo mecánico está construido como un brazo articulado con 5 grados de libertad y una pinza. Ver la siguiente figura.

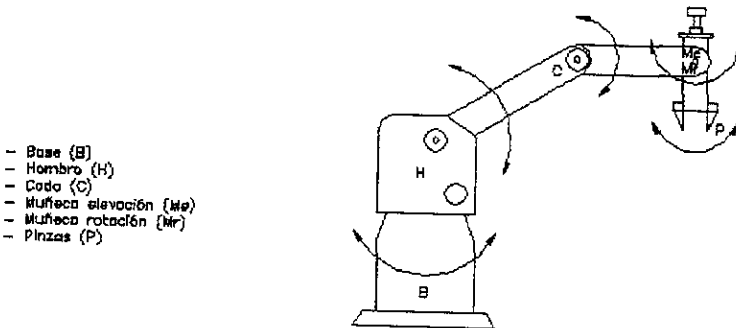


Fig. C. Esquema principal del robot SCORBOT-ER III.

Este tipo de construcción se denomina vertical articulada. El brazo se asemeja al brazo humano en que los grados de libertad son también rotativos. Los 5 grados de libertad y la pinza se activan mediante servo-motores de corriente continua (C.C.) en lazo cerrado por codificadores acoplados al eje de los ejes motrices. Todos los servo-motores de CC llevan una caja de engranajes de reducción, de forma que el eje de salida de la caja de engranajes gira a menor velocidad que el eje motor. Ver la figura C1.

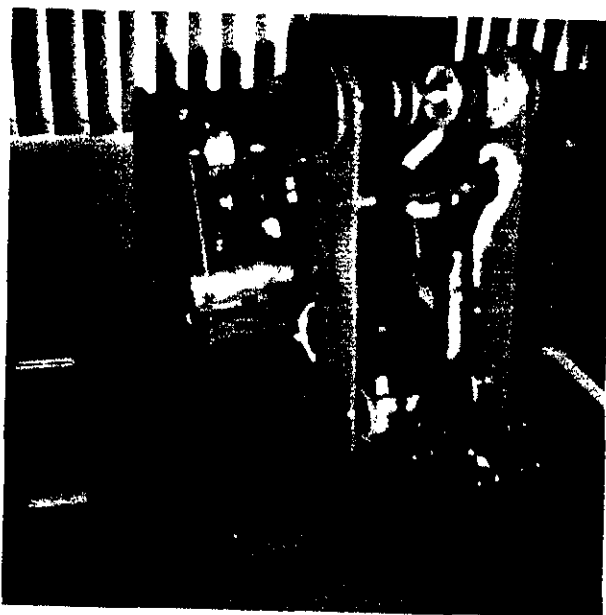


Fig. C1. Mecanismo interno del robot Scorbot.

El brazo mecánico lleva instalados 5 microinterruptores. Cuando todos ellos están cerrados, el robot está en una posición única y predeterminada. Esta posición se denomina Hard Home y es el único punto desde el que se hace referencia para el comienzo de cualquier trabajo con el robot. Debe señalarse que cada vez que se conecta el ordenador, el robot debe reponerse a la posición Hard Home.

## TRANSMISIÓN

Para accionar los eslabones del brazo mecánico se emplean varios tipos de transmisión.

- El movimiento de la base y el hombro del robot se crea mediante engranajes dentados.
- El movimiento del codo del robot se crea mediante engranajes dentados y correas de regulación.
- El movimiento de la muñeca se crea mediante correas de regulación y una unidad diferencial de engranajes dentados en el extremo del brazo.
- El cierre y la apertura de la pinza se crea mediante un tornillo de avance acoplado directamente a un servo-motor de CC.



## **CABLE DEL ROBOT**

El cable de red del robot, que va fijado a la base, consta de 42 conductores.

Los conductores se componen de 6 grupos de cables o alambres (un grupo para cada motor utilizado). Cada grupo tiene 7 conductores cuya función es la siguiente:

- 2 conductores para suministrar tensión al motor.
- 2 conductores para recibir impulsos del encoder óptico.
- 1 conductor al microinterruptor que determina la posición "Hard Home".
- 1 conductor para suministrar tensión al encoder.
- 1 conductor que proporciona conexión para el interruptor y el encoder.

Este cable es la única conexión entre el brazo del robot y el controlador. Todos los comandos al robot, tanto operativos como de control, pasan por este único cable.

## **EL CONTROLADOR**

Está formado por 6 componentes principales:

- La carcasa del controlador.
- La fuente de alimentación.
- El circuito principal del controlador.
- El circuito de visualización.
- El circuito de entradas/salidas.
- Los dos circuitos motrices (NPN y PNP).

## **CONEXIÓN CONTROLADOR-ORDENADOR**

El controlador electrónico del robot contiene un microprocesador que funciona bajo el control de un ordenador externo. Para conectar las dos unidades se ha elegido un sistema estándar de comunicación asíncrona en serie (norma RS-232C).

Del gran número de funciones de la norma RS-232C, solo empleamos las siguientes:

- Clavija 2 ➡ Transmisión del controlador
- Clavija 3 ➡ Recepción del controlador
- Clavija 7 ➡ Tierra

## PARÁMETROS DE COMUNICACIONES

Estos parámetros son los siguientes:

- Régimen de baudios: Este régimen es de 9,600 baudios, el cual lo determina el programa interno de la EPROM. Si el robot se conecta a un ordenador COMMODORE 64, el régimen de baudios necesario sería de 2,400 baudios, y por consiguiente se necesitaría una EPROM con el programa cargado a esta velocidad de transmisión.
- 1 bit de comienzo
- 8 bits de datos
- 1 ó 2 bits de parada
- Sin paridad

## SISTEMA OPERATIVO

El sistema operativo incorporado en el controlador electrónico del robot permite la programación del robot.

## COMANDOS PARA EL MOVIMIENTO DEL ROBOT

Los códigos ASCII se transmiten al controlador en el siguiente formato;

a M +/- b c d e CR, donde:

- a.- Número del 1 al 8 que identifica el motor que se desea mover.
- M.- Designación del tipo de comando (movimiento del motor).
- +/-.- "+" es un movimiento en dirección positiva y "-" es un movimiento en dirección opuesta.
- b c d e.- Número que no debe exceder de 8000. Da instrucciones al motor sobre la cantidad de movimiento en impulsos del encoder, cada uno de los caracteres se transmite independientemente.
- CR.- Retorno del carro.

Como vemos, el comando de movimiento máximo consta de 8 caracteres, por ejemplo, 3M-5432 CR. El código ASCII de estos 8 caracteres se envía uno detrás del otro. El controlador almacena el número recibido en el registro del motor respectivo y acciona el motor. En el momento del accionamiento, el controlador lee la cantidad de movimiento mediante un encoder y actualiza continuamente el número en el registro del motor en funcionamiento, hasta llegar a cero.

Esta instrucción no recibe respuesta del controlador.

#### Comando para comprobar el resto del movimiento del motor

Los códigos ASCII se envían al controlador en el siguiente formato;

aQ, donde:

- a.- Es un número entre 1 y 8 que identifica el motor que se desea comprobar.
- Q.- Define el tipo de comando (para comprobar el resto de los impulsos necesarios para completar el movimiento).

En respuesta a este comando el controlador envía un resultado binario sobre el número de impulsos que el motor ha de realizar. Como esta respuesta máxima es aproximadamente +/- 8000, la respuesta es en 14 bits enviados en dos bytes.

El controlador divide el número de 14 bits en 2, suma a cada parte el bit "1" como bit más significativo (MSB), lo que no tiene consecuencia alguna, y devuelve al ordenador externo los dos bytes que comienzan por el bit más significativo y después por el bit menos significativo. No hay ninguna señal adicional entre la transmisión de los 2 bytes.

#### Comando para activar las salidas

Los códigos ASCII se envían al controlador en el siguiente formato;

aS, donde:

- a.- Es un número entre el 1 y el 8 que identifica la salida.
- S.- Define el tipo de comando (SET OUTPUT).

Este comando no recibe respuesta del controlador.

### Comando para reponer las salidas

Los códigos ASCII se envían al controlador en el siguiente formato;  
aR, donde:

- a.- Es un número entre el 1 y el 8 que identifica la salida.
- R.- Define el tipo de comando (RESET OUTPUT).

Este comando no recibe respuesta del controlador.

### Comando para comprobar el estado de las entradas

Los códigos ASCII se envían al controlador en el siguiente formato;

aI, donde:

- a.- Es un número entre el 1 y el 8 que identifica la entrada.
- I.- Define el tipo de comando (INPUT).

El controlador responde con los siguientes códigos ASCII;  
"0" si la entrada esta en posición OFF  
"1" si la entrada esta en posición ON

Puede ser necesario enviar aIaI y hacer referencia a la segunda respuesta solamente.

### Comando para comprobar el final de carrera

Los códigos ASCII se envían al controlador en el siguiente formato;

aL, donde:

- a.- Es un número entre el 1 y el 8 que identifica el final de carrera.
- L.- Define el tipo de comando (LIMIT SWITCH).

El controlador responde con los siguientes códigos ASCII;  
"0" si el final de carrera esta en posición OFF.  
"1" si el final de carrera esta en posición ON.

Puede ser necesario enviar aLaL y hacer referencia a la segunda respuesta solamente.

### Comando para comprobar la terminación de una operación

El código ASCII se envía al controlador en el siguiente formato;

T, donde:

T.- Define el termino de la operación.

El controlador responde con los siguientes códigos ASCII:

"0" cuando el movimiento de los motores todavía no está completo.

"1" cuando todos los motores han terminado sus movimientos.

Puede ser necesario enviar TT y hacer referencia a la segunda respuesta solamente.