

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS BASADO EN UN MICROCONTROLADOR Y MANEJADO POR UN ENLACE DE LAN INALÁMBRICA**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES**

**ANDRÉS PATRICIO CARRIÓN BRAVO  
DIEGO FERNANDO MANCERO ZAMBONINO**

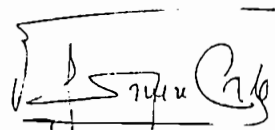
**DIRECTOR: ING. FERNANDO FLORES**

**Quito, Febrero 2006**

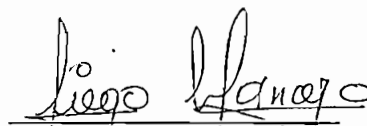
## DECLARACIÓN

Nosotros, Andrés Patricio Carrión Bravo y Diego Fernando Mancero Zambonino, declaramos que el presente trabajo es de nuestra autoría; que no ha sido previamente presentado por ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

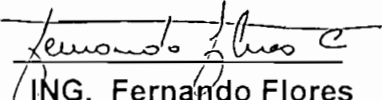


Andrés P. Carrión B.

  
Diego F. Mancero Z.

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Andrés Patricio Carrión Bravo y Diego Fernando Mancero Zambonino, bajo mi supervisión.

  
ING. Fernando Flores  
DIRECTOR DEL PROYECTO

# *Dedicatoria*

*A Mauro Salazar, compañero y amigo, cuya existencia se vio truncada prematura y trágicamente por una de las pestes modernas: los accidentes de tránsito.*

*A Dios, que me acompaña y protege siempre y a quien le debo todo.*

*A mi Madre, por su abnegación, entrega, sacrificio, y por sobre todas las cosas, por la fortaleza y valentía para afrontar la vida, cualidades demostradas en la superación de la dura prueba a la que el destino la sometió el año anterior.*

*A mi Padre, por su integridad, sabiduría, capacidad, generosidad y entereza, siendo siempre mi modelo a seguir.*

*A mis hermanos, con quienes he compartido penas y alegrías a lo largo de todos estos años.*

*A todos mis familiares que aunque ya no estén presentes físicamente, su recuerdo se mantiene latente.*

*Diego*

*Dedico este trabajo a las personas más importantes de mi vida:*

*A mi hijo Gabriel, quien vino a iluminar mi existencia dándome lecciones de vida y brindándome la felicidad más grande que el ser humano puede tener, el ser padre.*

*A mi Madre porque siempre me dieron ánimo ya sea con una palabra un gesto, una sonrisa o una caricia.*

*A Marieta, una persona muy especial que a pesar de las circunstancias estuvo siempre a mi lado.*

*A mi Tía y Hermana que siempre estuvieron apoyándome en todo momento.*

*A mi sobrino Matheo que lo quiero mucho, y a toda mi familia por ser como es.*

*Andrés*

# *Agradecimiento*

*A mis padres, que me brindaron su apoyo y estímulo incondicional, congratulándose con mis logros y confortándome en los momentos difíciles; son el pilar fundamental sobre el que baso toda mi existencia, siendo lo más importante el ejemplo de vida que me lo imparten día tras día.*

*A mis hermanos, que me acompañaron y tendieron una mano siempre que lo necesite.*

*A la Escuela Politécnica Nacional, Institución que me abrió sus puertas, ofreciéndome la oportunidad de desarrollarme humana e intelectualmente.*

*A mi prima Ruth, cuya generosidad y bondad se hizo manifiesta al acogerme en su hogar durante mis cinco primeros años de estudios universitarios, brindándome hasta hoy su apoyo sincero.*

*A la Matemática Ruth Cueva, quien además de impartir valiosos conocimientos en el aula de clase, el destino hizo que se convirtiera en una gran amiga, guiándome con sus consejos y apoyándome en todas las formas posibles.*

*Al Ing. Fernando Flores, que a pesar de sus múltiples ocupaciones, su esfuerzo y buena voluntad hizo posible la culminación del presente proyecto.*

*A todos mis familiares, amigos, maestros y compañeros de aula, quienes de una u otra manera siempre me impulsaron a seguir adelante; sin todos y cada uno de ellos hubiese sido imposible el alcanzar este logro en mi carrera profesional.*

**Diego**



*Agradezco especialmente a DIOS el cual por su gran misericordia nos da salud y vida y me dio la posibilidad de estudiar.*

*A la Escuela Politécnica Nacional por brindarme la oportunidad de cursar una Ingeniería.*

*A mi Madre y Tía, que me apoyaron incondicionalmente en los todos los momentos de mi vida; ofreciéndome sabios consejos y nutriéndome con su ejemplo, fortaleciéndome ante las adversidades.*

*A mi profesor guía, Ing. Fernando Flores por la paciencia y el tiempo dedicado y a la Mat. Ruth Cueva por brindarnos su amistad y apoyo incondicional.*

*A mis grandes amigos Marieta, Tino, Fabián, Luís, Sebas, Hugo, Daniel, Willy, y Diego, los cuales a lo largo de la carrera han sabido colaborar con un consejo o una palabra de apoyo, y a todos los que me apoyaron y ayudaron a realizar este trabajo.*

*Andrés*

## CONTENIDO

CONTENIDO.....	i
INDICE DE FIGURAS.....	v
INDICE DE TABLAS.....	vii
PRESENTACIÓN.....	viii

### CAPÍTULO 1

INTRODUCCIÓN.....	1
1.1 GENERALIDADES SOBRE COMUNICACIONES PARA LAS REDES DE ÁREA LOCAL INALÁMBRICAS.....	1
1.1.1 RESEÑA HISTÓRICA.....	1
1.1.2 CONCEPTO DE RED DE ÁREA LOCAL INALÁMBRICA.....	3
1.1.3 TECNOLOGÍAS Y CONFIGURACIONES.....	4
1.1.3.1 Banda angosta.....	4
1.1.3.2 Banda ancha.....	4
1.1.3.3 Redes Peer to peer o ad-hoc.....	5
1.1.3.4 Modo Infraestructura.....	5
1.1.3.5 Otras configuraciones.....	7
1.1.4 TECNOLOGÍAS INALÁMBRICAS.....	8
1.1.4.1 Bluetooth.....	8
1.1.4.2 Estándar HomeRF.....	9
1.1.4.3 Estándar HiperLAN2.....	10
1.1.4.4 Estándar 802.11.....	10
1.1.5 MODELO DE CAPAS.....	12
1.1.5.1 Capa Física.....	13
1.1.5.1.1 Radio Frecuencia.....	13
1.1.5.1.2 Infrarrojos.....	16
1.1.5.2 Capa de Enlace.....	18
1.1.5.2.1 Capa LLC.....	18
1.1.5.2.2 Capa MAC.....	19
1.1.6 PROTOCOLO IP MÓVIL.....	19
1.1.6.1 Funcionamiento.....	20
1.1.7 SEGURIDAD EN LAS COMUNICACIONES LAN INALÁMBRICAS..	22
1.1.7.1 Riesgos de una Infraestructura Lan Inalámbrica.....	22
1.1.7.2 Mecanismos de Seguridad.....	24
1.2 REDES DE ACCESO.....	27
1.2.1 CLASIFICACIÓN DE LAS REDES DE ACCESO.....	27
1.2.1.1 Redes Cableadas.....	27
1.2.1.1.1 Las redes de acceso vía cobre.....	27

1.2.1.1.2	Las redes de acceso vía fibra óptica .....	27
1.2.1.1.2.1	Redes Híbridas Fibra-Coaxial (HFC) .....	28
1.2.1.1.2.2	Redes Ópticas Pasivas (PON) .....	28
1.2.1.2	Las redes de acceso Inalámbrico .....	28
1.2.2	REDES DE ACCESO REMOTO.....	29
1.2.2.1	Acceso remoto Telefónico .....	29
1.2.2.2	Acceso remoto Inalámbrico .....	30
1.2.2.3	Protocolos de Acceso remoto.....	31
1.2.2.3.1	Protocolo Internet de línea serie (SLIP).....	31
1.2.2.3.2	Protocolo punto a punto (PPP).....	32
1.2.2.3.3	Protocolo de enlace múltiple PPP (MP).....	32
1.3	COMUNICACIÓN SERIAL .....	33
1.3.1	NORMAS DE COMUNICACIÓN SERIAL.....	33
1.3.1.1	Norma RS-232.....	33
1.3.1.2	Norma RS-422A y RS-423 .....	37
1.3.1.3	Norma RS-485.....	38
1.3.2	COMPARACIÓN ENTRE NORMAS DE COMUNICACIÓN SERIAL.....	39
1.4	HERRAMIENTAS UTILIZADAS.....	40
1.4.1	PROTOCOLOS TCP/IP .....	40
1.4.1.1	Arquitectura .....	40
1.4.1.1.1	Capa Acceso a la red .....	40
1.4.1.1.2	Capa Internet.....	41
1.4.1.1.3	Capa Transporte .....	42
1.4.1.1.4	Capa Aplicación .....	43
1.4.2	LENGUAJE ASSEMBLER.....	45
1.4.2.1	Introducción .....	45
1.4.2.2	Etapas para la realización de un programa .....	46
1.4.2.3	Código de Máquina.....	48
1.4.2.4	Ejecución .....	49
1.4.2.5	Depuración .....	49
1.5	ETAPAS DEL PROYECTO .....	50
1.5.1	SISTEMA DE ADQUISICIÓN DE DATOS .....	50
1.5.1.1	EI MAX 180.....	50
1.5.2	INTERFAZ RS 232/WLAN EZL-300W.....	55
1.5.2.1	Introducción .....	55
1.5.2.2	Funcionamiento .....	56
1.5.2.3	Aplicaciones .....	56
1.5.3	PROCESAMIENTO Y ADQUISICIÓN DE LOS DATOS.....	57
1.5.3.1	Microcontroladores de la familia 8051 .....	57
1.5.3.2	Configuración de la Red Inalámbrica con el EZL-300W .....	60

## CAPÍTULO 2

	DESCRIPCIÓN Y ANÁLISIS DE LOS INTERFACES PCI Y PCMCIA .....	62
2.1	LOCAL PCI (Peripheral Component Interconnect).....	63
2.1.1	INTRODUCCIÓN.....	63
2.1.2	DESCRIPCIÓN GENERAL.....	64
2.1.3	CARACTERÍSTICAS FÍSICAS .....	65
2.1.4	CAPACIDADES DEL BUS LOCAL PCI .....	67

2.2	PCMCIA (Personal Computer Memory Card International Association)	69
2.2.1	INTRODUCCIÓN	69
2.2.2	CARACTERÍSTICAS FÍSICAS	71
2.2.2.1	Zócalos y Conectores	72
2.2.2.2	Características Ambientales	73
2.2.3	CARACTERÍSTICAS ELÉCTRICAS	74
2.2.3.1	El interfaz de Memoria	74
2.2.3.2	Interoperabilidad	75
2.2.3.3	Configurando la Tarjeta	75
2.2.3.4	El Metaformato	77
2.2.4	ADAPTADOR DE BUS DEL HOST PC CARD	78
2.2.5	ESTRUCTURA DE LA INFORMACIÓN DE LA TARJETA (CIS)	79
2.2.5.1	Tuples o Bloques de Datos	80
2.2.5.2	Configuración de la Tabla	81
2.2.5.3	Dispositivos Comúnmente usados por el CIS	82
2.2.5.4	Los Registros de Configuración	82

### CAPÍTULO 3.

	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE ADQUISICIÓN DE DATOS	83
3.1	ASPECTOS PRELIMINARES DEL SISTEMA DE ADQUISICIÓN DE DATOS	83
3.2	CONFIGURACIÓN DEL HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS	84
3.2.1	MICROCONTROLADOR	84
3.2.1.1	CPU	85
3.2.1.2	Memoria de Programa	85
3.2.1.3	Memoria de Datos	85
3.2.1.4	Circuitaria de Temporización	86
3.2.1.5	Entradas/Salidas	86
3.2.1.6	Puertos	86
3.2.1.7	Características	87
3.2.2	MAX 180	88
3.2.2.1	Descripción de los componentes del MAX 180	88
3.2.2.2	Descripción de los Pines	91
3.2.3	MAX 233	91
3.2.4	DIVISOR DE FRECUENCIA	93
3.2.5	TRANSCEIVER	94
3.2.6	Alimentación de los Circuitos	95
3.3	APLICACIONES	96
3.3.1	MEDICIÓN DE TEMPERATURA	97
3.3.1.1	Termistores	98
3.3.1.2	Termocuplas	98
3.3.1.3	Resistencias Detectoras de Temperatura (RTD)	99
3.3.1.4	Sensores de temperatura embebidos en un Circuito Integrado	100
3.3.2	Diseño del Sensor de Temperatura	101
3.4	CONFIGURACIÓN DEL SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS	103

3.4.1	LECTURA DEL MAX 180 .....	104
3.4.2	SOFTWARE IMPLEMENTADO EN EL SERVIDOR.....	109
3.4.3	SOFTWARE IMPLEMENTADO EN EL CLIENTE .....	109

## CAPÍTULO 4

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA, PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS.....		110
4.1	DESCRIPCIÓN DE LAS TARJETAS Y DISTRIBUCIÓN DE LOS ELEMENTOS QUE CONFORMAN EL PROTOTIPO.....	111
4.1.1	TARJETA DE VARIACIÓN DEL VALOR LAS SEÑALES DE ENTRADA.....	111
4.1.2	SENSOR DE TEMPERATURA.....	112
4.1.3	SISTEMA DE ADQUISICIÓN DE DATOS .....	112
4.1.4	TARJETA DE ALIMENTACIÓN .....	114
4.2	INSERCIÓN DEL PROTOTIPO EN LA RED INALÁMBRICA.....	115
4.2.1	Configuración del Sistema.....	115
4.2.1.1	Servidor .....	117
4.2.1.2	Clientes Potenciales .....	118
4.2.2	COMPROBACIÓN DEL ENLACE INALÁMBRICO.....	119
4.3	PRUEBAS DEL PROTOTIPO CON PC.....	120
4.3.1	PRUEBAS DE CONEXIÓN .....	120
4.3.2	ACCESO REMOTO.....	121
4.3.3	PRUEBAS DE OPERACIÓN.....	122
4.3.4	ANÁLISIS DE RESULTADOS .....	124
4.4	PRUEBAS DEL PROTOTIPO CON EZL W300.....	126
4.4.1	CONFIGURACIÓN .....	126
4.4.2	PRUEBAS DE CONEXIÓN .....	126
4.5	COSTOS DE LA IMPLEMENTACIÓN .....	127

## CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES .....		128
REFERENCIA BIBLIOGRÁFICAS.....		136
<b>ANEXOS</b> .....		141
ANEXO 1: CÓDIGO FUENTE DE LOS PROGRAMAS DESARROLLADOS ..		142
ANEXO 2: EL EZL-300W .....		168
ANEXO 3: ESTÁNDAR PCMCIA / JEIDA .....		189
ANEXO 4: EL MAX 180.....		233
ANEXO 5: EL MICROCONTROLADOR AT89C52.....		251
ANEXO 6: GLOSARIO .....		263

## INDICE DE FIGURAS

Figura 1.1. Modo Ad – Hoc.....	5
Figura 1.2. Modo Infraestructura .....	6
Figura 1.3. Varios Puntos de acceso con capacidad de Roaming .....	7
Figura 1.4. Interconexión de LAN mediante antenas direccionales.....	7
Figura 1.5. Ejemplo de tecnología HomeRF.....	9
Figura 1.6. Estándar 802.11 dentro del modelo de capas OSI.....	13
Figura 1.7. Banda Esparcida (Spread Spectrum).....	13
Figura 1.8. Secuencia Binaria Seudoaleatoria .....	14
Figura 1.9. Codificación de Barker .....	15
Figura 1.10. Banda Esparcida por Salto de Frecuencia .....	16
Figura 1.11. Transmisión por infrarrojo.....	18
Figura 1.12. Arquitectura Móvil Ipv4.....	20
Figura 1.13. Muestra las diferentes formas de Acceso remoto Inalámbrico o por cable.....	29
Figura 1.14. Conexión entre un DTE y un DCE.....	34
Figura 1.15. Conector DB25 (derecha) y DB9 (izquierda) .....	34
Figura 1.16. Niveles de voltaje utilizados en la norma RS-232 .....	34
Figura 1.17. Ejemplo de un carácter de transmisión RS232 (La señal de reloj en una transmisión asincrónica es interno). .....	35
Figura 1.18. Conexión entre dos equipos RS-232 DTE sin señales de acoplamiento .....	36
Figura 1.19. Conexión entre dos equipos RS-422 sin señales de acoplamiento..	37
Figura 1.20. Conexión entre equipos 4D-RS-485.....	38
Figura 1.21. Capa Acceso a la red .....	41
Figura 1.22. Capa Internet.....	42
Figura 1.23. Capa Transporte .....	43
Figura 1.24. Capa Aplicación .....	45
Figura 1.25. Diagrama de Pasos para la realización de un programa.....	47
Figura 1.26. Procedimiento de Ejecución .....	49
Figura 1.27. Circuito de Operación Típico del MAX 180.....	52
Figura 1.28. Circuito equivalente del comparador analógico del conversor AD (T/H) .....	52
Figura 1.29. Modo Puerto de Entrada/Salida .....	53
Figura 1.30. Diagrama de tiempo de la lectura paralela (Modo=1, HBEN=0).....	53
Figura 1.31. Diagrama de tiempo de la lectura de 2 bytes en el Modo 1 .....	54
Figura 1.32. Conexión del EZL-300W con una red inalámbrica de infraestructura .....	56
Figura 1.32. Distribución de pines del microcontrolador 8051.....	58
Figura 1.33. Distribución de las localidades de memoria en el micri 8051 .....	59

Figura 2.1. Diagrama de un bus PCI típico con sus conexiones. ....	65
Figura 2.2. Diagramas de un conector PCI de 32 bits y de un conector PCI de 64 bits. ....	66
Figura 2.3. Diagrama Funcional de las conexiones de pines PCI .....	68
Figura 2.4. Tarjeta PC Card Tipo I .....	71
Figura 2.5. Tarjeta PC Card Tipo II .....	71
Figura 2.6. Tarjeta PC Card Tipo III .....	72
Figura 2.7. Zócalo y conector PC Card. ....	72
Figura 2.8. Zócalo de interfaz para memoria PCMCIA. ....	75
Figura 2.9. Ambiente PCMCIA .....	78
Figura 2.10 Esquema del CIS que consiste en una lista encadenada de cuatro tuples. ....	80
Figura 3.1. Diagrama funcional del Diseño.....	84
Figura 3.2. Diagrama de Bloques del MAX 180. ....	88
Figura 3.3. Distribución de pines y configuración interna del MAX 233.....	93
Figura 3.4. Diagrama y Símbolo Lógico del 74LS76 .....	93
Figura 3.5. Configuración para obtener una frecuencia de 1.382 MHz .....	94
Figura 3.6. Distribución de pines del 74LS245.....	95
Figura 3.7. Aspecto Físico de una Termocupla. ....	98
Figura 3.8. Configuración física y distribución de pines del LM335.....	101
Figura 3.9. Configuraciones Básicas del Sensor LM335.....	102
Figura 3.10. Sensor de temperatura alimentado con una fuente de corriente....	102
Figura 3.11. Diagrama de Flujo del Software implementado.....	105
Figura 3.12. Diagrama de flujo de la lectura de un dato del MAX 180 .....	106
Figura 3.13. Diagrama de flujo del Software Servidor .....	1067
Figura 3.14. Diagrama de flujo del Software Cliente .....	108
Figura 4.1. Tarjeta de canales variables.....	111
Figura 4.2. Tarjeta del Sensor de Temperatura.....	112
Figura 4.3. Tarjeta del Sistema de Adquisición de Datos.....	114
Figura 4.4. Sensor de Temperatura.....	115
Figura 4.5. Configuraciones de la conexión del prototipo con la red inalámbrica. ....	116
Figura 4.6. Configuración de la tarjeta de red en el Servidor. ....	117
Figura 4.7. Configuración del software en el Servidor. ....	117
Figura 4.8. Redes en la zona de recepción de la tarjeta inalámbrica .....	118
Figura 4.9. Ejecución del comando PING en el cliente.....	119
Figura 4.10. Ejecución del comando PING en el cliente.....	120
Figura 4.11. Proceso de conexión desde el servidor.....	121
Figura 4.12. Proceso de conexión desde el cliente. ....	121
Figura 4.13. Visualización de los datos adquiridos y recibidos.....	122
Figura 4.14. Visualización de los datos en formato de valores numéricos .....	1223
Figura 4.15. Visualización de los datos en forma de barras.....	124
Figura 4.16. Comparación del valor adquirido y la medición de voltaje con un multímetro digital. ....	125
Figura 4.17. Ejecución del comando PING en el cliente.....	126

## INDICE DE TABLAS

Tabla 1.1. Clases de dispositivos Bluetooth .....	8
Tabla 1.2. Resumen de Tecnologías Wireless .....	11
Tabla 1.3. Terminales del conector DB9. ....	35
Tabla 1.4. Comparación entre los estándares RS-232C, RS-422A y RS-423A. ..	39
Tabla 2.1. Formato Básico de un Tuple.....	81
Tabla 3.1. Direccionamiento y Canal Seleccionado del MAX 180.....	90
Tabla 3.2. Pines del MAX 180 y su función .....	91
Tabla 3.3. Características Eléctricas del MAX 233.....	92
Tabla 3.4. Habilitaciones del 74LS245 .....	95
Tabla 3.5. Voltaje de alimentación y Potencia consumida por cada dispositivo. .	96
Tabla 4.1. Valores resultantes de las pruebas de conexión cliente – servidor. .	125
Tabla 4.2. Costos de la Implementación .....	127



## PRESENTACIÓN

El propósito principal hacia el que está encaminado este proyecto es la construcción de un prototipo que permita la captación y medición de señales analógicas, que serán transformadas a digitales para ser transmitidas hacia una red inalámbrica, para ser utilizadas, procesadas y visualizadas en un terminal remoto, inicialmente dentro de la misma red inalámbrica.

Para este efecto, primero es necesario recabar todos los conceptos referentes a los elementos que integran un sistema de comunicación electrónico inalámbrico; sus interfaces, estándares, protocolos y los dispositivos que son comúnmente utilizados dentro de éste, para así alcanzar la compatibilidad entre los elementos del sistema a diseñar y la red de la que se requiere llegar a formar parte.

En este sentido, los capítulos I y II enmarcan todos los tópicos que se han considerado indispensables para con la información proporcionada poseer los argumentos necesarios para emprender la elección de elementos a utilizar en el dispositivo, y diseñarlo posteriormente.

Se pone especial énfasis en la descripción de los interfaces típicamente utilizados para transferir información de una red inalámbrica a un computador: PCI y PCMCIA.

Para facilitar el diseño se lo divide en subsistemas o bloques: adquisición de datos analógicos, transferencia de estos datos hacia un elemento externo y finalmente envío de esta información hacia una red inalámbrica; el capítulo III

describe cada etapa y los motivos de la selección de los elementos que conforman cada una de ellas.

Además, como complemento a la comprobación de la utilidad de este dispositivo, se diseña un sensor de temperatura, para mostrar una variable física real como una aplicación de las más simples de este dispositivo.

El paso siguiente al diseño es la implementación, interconexión y finalmente las pruebas del funcionamiento del dispositivo proyectado; analizar los resultados obtenidos, y de ser necesario realizar variaciones en el hardware o software diseñado si el comportamiento resultante no es el esperado o existe alguna dificultad no prevista.

El capítulo IV proporciona los suficientes elementos de juicio para mostrar si el dispositivo está correctamente diseñado e implementado, tanto a nivel de adquisición de datos, como de transmisión de éstos a una red inalámbrica.

Finalmente se presentan las conclusiones y recomendaciones alcanzadas a lo largo de todo el proyecto, como consecuencia tanto de la búsqueda de información, como del diseño, implementación y pruebas del prototipo construido.

# CAPÍTULO 1

## INTRODUCCIÓN

En este capítulo se describe las distintas características que existen en un sistema de comunicación inalámbrica entre dispositivos electrónicos como por ejemplo: radio frecuencia, Infrarrojo, etc., con la finalidad de obtener una mayor experiencia con los diferentes dispositivos que interactuarán en el sistema a ser desarrollado, el cual se refiere a un Sistema de Adquisición de Datos manejado por un microcontrolador mediante un enlace de radio frecuencia. El objetivo de este capítulo es analizar la forma en que una señal viaja a través del aire, los distintos tipos de protocolos y las herramientas utilizadas, abordar diferentes conceptos que existen en una comunicación inalámbrica. Además de conocer las diferentes etapas que constituyen el proyecto, sus características, componentes y limitaciones.

### **1.1 GENERALIDADES SOBRE COMUNICACIONES PARA LAS REDES DE ÁREA LOCAL INALÁMBRICAS**

#### **1.1.1 RESEÑA HISTÓRICA**

Los pioneros en el uso de redes inalámbricas fueron los radioaficionados mediante sus emisoras, transmitieron datos a una velocidad de 9600 bps. Este tipo de redes se implementaron por primera vez en 1979. La casa IBM Suiza, utilizó enlaces infrarrojos creando una red de área local en una fábrica, y

posteriormente se utilizaron implementaciones basadas en tecnologías de enlaces por microondas según los esquemas de transmisión de espectro ensanchado.

En marzo de 1985 la Comisión Federal de Comunicaciones, FCC, organismo encargado de la regulación de las telecomunicaciones en Estados Unidos, asignó a los sistemas WLAN las bandas de frecuencia 902-928 MHz, 2.400-2.4835 GHz y 5.725-5.850 GHz también conocidas como ISM (Industrial, Científica y Médica). La asignación de una localización frecuencial fija permitió un mayor desarrollo a nivel industrial. A partir de este momento, las redes de área local inalámbrica dejaron de ser experimentales para comenzar a introducirse en el mercado.

Entre los años 1985 y 1990 se trabajó en el desarrollo de productos WLAN y finalmente, en mayo de 1991, se publicaron algunos trabajos que tratan sobre redes inalámbricas que superaban una velocidad de transferencia de 1 Mbps, velocidad mínima a partir de la cual el comité de la IEEE considera que una red es de área local.

Hasta ese momento las WLAN habían tenido una aceptación marginal en el mercado por dos razones fundamentales: falta de un estándar y precios elevados de las soluciones inalámbricas.

Luego, en el año 1997 el organismo regulador IEEE publicaba el estándar 802.11 (802 hace referencia al grupo de documentos que describen las características de las LAN alámbricas o Ethernet) dedicado a redes LAN inalámbricas. Dentro de este mismo campo y anteriormente, en el año 1995, tenemos la aparición del Bluetooth, una tecnología de la empresa Ericsson, utilizada para conectar mediante ondas de radio los teléfonos móviles con diversos accesorios. En poco tiempo se formó un grupo de estudio constituido por fabricantes que estaban interesados en esta tecnología para aplicarla a otros dispositivos, como PDAs, terminales móviles e incluso electrodomésticos.

Pero el verdadero desarrollo de este tipo de redes surgió a partir de que la FCC aprobó el uso civil de la tecnología de transmisiones de espectro disperso

(SS spread spectrum, o espectro ensanchado), pese a que en un principio lo prohibió por la amplia ocupación del espectro que presentaba.

### 1.1.2 CONCEPTO DE RED DE ÁREA LOCAL INALÁMBRICA

Una red de área local inalámbrica (WLAN) se define como una red con un alcance geográfico limitado (1Km como máximo), una velocidad de transferencia de datos relativamente alta (superior a 1 Mbps), con baja tasa de errores; y que utiliza ondas electromagnéticas como medio de transmisión de la información que viaja a través del aire, enlazando los diferentes equipos o terminales móviles asociados a la red básicamente implementados a través de enlaces con tecnologías de microondas o de infrarrojos, administrados de forma privada.

Las redes locales inalámbricas más que una sustitución de las LAN alámbricas convencionales son un complemento de las mismas, ya que permite el intercambio de información entre los distintos medios en una forma transparente al usuario.

En este sentido el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red total donde converjan los dos sistemas.

Este hecho proporciona al usuario una gran movilidad sin perder conectividad. El atractivo fundamental de este tipo de redes es la facilidad de instalación y el ahorro que supone la supresión del medio de transmisión cableado. Aún así sus prestaciones son menores en lo referente a la velocidad de transmisión que se sitúa entre los 2 y los 11 Mbps frente a los 10 y los 100 Mbps ofrecidos por una modesta red convencional actual.

Las redes WLAN se componen fundamentalmente de dos tipos de elementos, los puntos de acceso y los dispositivos de cliente. Los puntos de acceso actúan como un concentrador o hub que reciben y envían información vía

radio a los dispositivos de clientes, que pueden ser de cualquier tipo, habitualmente, un PC o PDA con una tarjeta de red inalámbrica.

El uso más popular de las WLAN implica la utilización de tarjetas de red inalámbricas, cuya función es permitir al usuario conectarse a la LAN empresarial sin la necesidad de una interfaz física.

### **1.1.3 TECNOLOGÍAS Y CONFIGURACIONES**

Según el diseño requerido se tienen distintas tecnologías aplicables:

#### **1.1.3.1 Banda angosta**

Se transmite y recibe en una específica banda de frecuencia lo más angosta posible para el paso de información. Los usuarios tienen distintas frecuencias de operación de modo que se evitan las interferencias. Así mismo un filtro en el receptor de radio se encarga de dejar pasar únicamente la señal esperada en la frecuencia asignada. Una desventaja del uso de esta tecnología es que la utilización de una determinada banda de frecuencias requiere la autorización del organismo regulador local.

#### **1.1.3.2 Banda ancha**

La técnica de espectro ensanchado es actualmente la más utilizada en las LANs inalámbricas. Inicialmente, las técnicas de espectro disperso fueron desarrolladas con el propósito de combatir las interferencias en las comunicaciones militares, lo cual se logra esparciendo el espectro de la señal transmitida sobre determinadas bandas de frecuencias.

Existen dos técnicas de dispersión del espectro que se han desarrollado: el salto de frecuencia y secuencia directa, más adelante se explican estas técnicas.

La versatilidad y flexibilidad de las redes inalámbricas es la principal razón para que la complejidad de una WLAN sea muy variable y variada, esta gran

variedad de configuraciones conlleva a que estas redes se adapten a casi cualquier necesidad y entorno. Estas configuraciones se pueden dividir en dos grandes grupos:

- Redes Peer to peer o modo ad-hoc.
- Modo Infraestructura.

### 1.1.3.3 Redes Peer to peer o ad-hoc

El modo ad-hoc (también conocido como Peer to peer) es la configuración más básica de una red inalámbrica, permitiendo un máximo de 256 clientes ubicados en estaciones inalámbricas que se comunican directamente entre sí, sin utilizar puntos de acceso para conectarse a la LAN cableada. Para que la comunicación entre las estaciones sea posible es necesario que cada una de ellas esté en el rango de cobertura radioeléctrica de las demás. Las redes de tipo ad-hoc son muy sencillas de implementar y no requieren ningún tipo de gestión administrativa o preconfiguración.

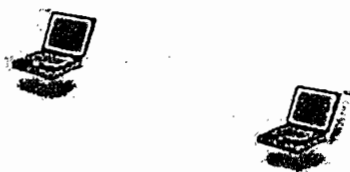


Figura 1.1. Modo Ad –Hoc

Esta modalidad resulta útil cuando se requiere una red donde previamente no existe o no se requiere infraestructura cableada permanente, como en hoteles, aeropuertos, o para eventos ocasionales como convenciones, conferencias, exposiciones, etc.

### 1.1.3.4 Modo Infraestructura

Estas configuraciones utilizan el concepto de celda, considerada como el área en la que una señal radioeléctrica es efectiva. A pesar de que en el caso de las redes inalámbricas esta celda suele tener un tamaño reducido, mediante el

uso de varias fuentes de emisión, es posible combinar adecuadamente las celdas de estas señales para cubrir un área más extensa. Para poder aumentar el número de celdas, y por lo tanto el área de cobertura de la red, se utilizan los llamados Puntos de Acceso, que tienen un alcance finito, del orden de 150 m en lugares cerrados y 300 m en zonas abiertas.

Este tipo de redes inalámbricas consta de al menos un punto de acceso conectado a la red cableada y un conjunto de estaciones finales inalámbricas. La configuración que posee un solo punto de acceso se denomina Basic Service Set (BSS).

Un Extended Service Set (ESS) es un conjunto de dos o más BSS que forman una sola subred.

Teniendo en cuenta que la mayoría de las WLAN tendrán la necesidad de conectarse a las LAN cableadas corporativas, éste será el modo de operación generalmente adoptado.

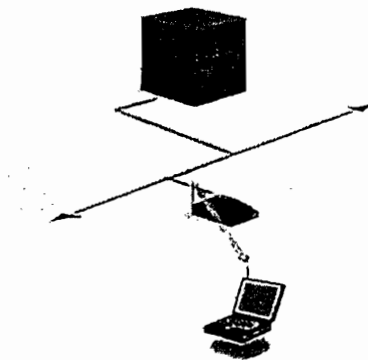


Figura 1.2. Modo Infraestructura

La topología de modo infraestructura es capaz de dotar a una red inalámbrica de varias potencialidades. Además del evidente aumento del alcance de la red, cuando se utilizan varios puntos de acceso, y por lo tanto el empleo de varias celdas que cubran el totalmente el lugar de interés, solapando sus áreas de modo que los clientes puedan movilizarse por toda la red sin cortes entre un grupo de puntos de acceso. Este proceso se denomina "roaming".



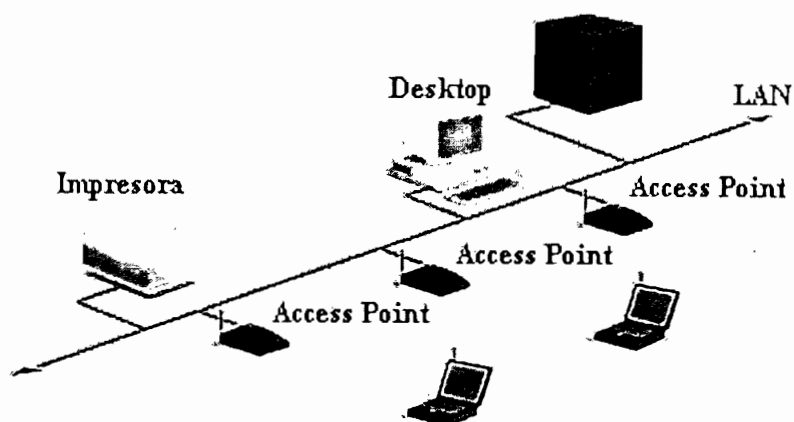


Figura 1.3. Varios Puntos de acceso con capacidad de Roaming

#### 1.1.3.5 Otras configuraciones

Las potencialidades de las redes inalámbricas pueden verse ampliadas gracias a la interconexión con otras redes, sobre todo con redes cableadas, de esta forma los recursos disponibles en ambas redes se comparten.

Mediante el uso de antenas (direccionales u omnidireccionales) es posible conectar dos redes separadas por varios cientos de metros, como por ejemplo dos redes locales situadas en dos edificios distintos.

De esta forma, una LAN cableada se beneficia de la tecnología inalámbrica para realizar interconexiones con otras redes, que de otra forma serían más costosas, o simplemente imposibles.

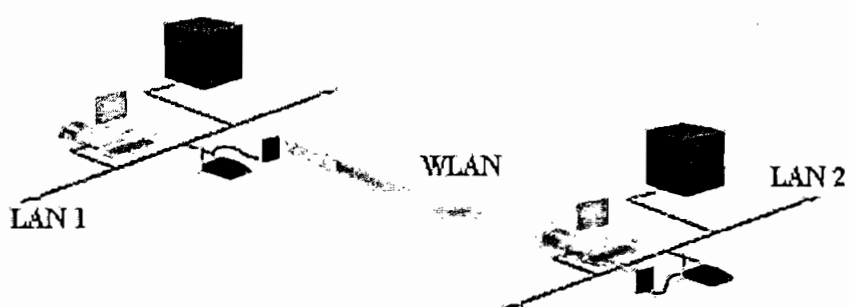


Figura 1.4. Interconexión de LAN mediante antenas direccionales

### 1.1.4 TECNOLOGÍAS INALÁMBRICAS

En la actualidad existen varias tecnologías inalámbricas estandarizadas para el diseño de redes de datos: Bluetooth, HomeRF, HiperLAN2 y 802.11 a continuación se detallará las principales características de cada una ellas:

#### 1.1.4.1 Bluetooth

En el diseño de Bluetooth primó la importancia de obtener dispositivos de pequeño tamaño, bajo consumo de potencia y bajo costo. Esta tecnología está orientada a interconectar cualquier dispositivo electrónico: ordenadores, PDA, teléfonos, electrodomésticos, periféricos, etc. en pequeños radios de cobertura (10m) conformando redes PAN (Personal Area Networks), capaces de transmitir voz y datos. La velocidad de transmisión es de 720 kbps por canal. Si se emplean puntos de acceso, el radio de cobertura puede llegar hasta 100m.

Los dispositivos bluetooth operan en la banda ISM de los 2.4 GHz, para la cual no es necesario licencia. La banda de frecuencias empleada se extiende desde los 2.400 a los 2.4835 GHz, conformando un total de 79 canales de RF de la forma:  $f=2402+k$  MHz,  $k=0,\dots,78$ .

Emplean FHSS (Frequency Hopping Spread Spectrum) utilizando señales full-duplex a 1600 saltos o "hops" por segundo. Dependiendo de la potencia del dispositivo existen 3 clases:

	Potencia de Salida	
	Máxima	Mínima
<b>Clase 1</b>	100 mW	1 mW
<b>Clase 2</b>	2,5 mW	0,25 mW
<b>Clase 3</b>	1 mW	-

Tabla 1.1. Clases de dispositivos Bluetooth

### 1.1.4.2 Estándar HomeRF

Las redes de este tipo están diseñadas principalmente para su uso en ambientes domésticos. La base de estas redes es el protocolo de acceso compartido SWAP-CA (Shared Wireless Access Protocol – Cordless Access), que une las cualidades de CSMA/CA para la transmisión de datos y las de TDMA para la transmisión de voz, siendo a su vez capaz de integrarse tanto con las redes de telefonía como con Internet.

Esta tecnología la administra el grupo HRFWG (*HomeRF Working Group*) que engloba diferentes fabricantes y regula las características de la norma para evitar incompatibilidades en los elementos que se fabriquen y asegurar así la interoperatividad entre ellos. Este tipo de redes operará en la frecuencia de 2,4 GHz con transmisión por FHSS. Entre sus características se destacan:

- Soporta tres canales de voz, con lo que se puede utilizar el teléfono a la vez que se transmiten datos.
- Soporta hasta 128 dispositivos en red.
- Emplea encriptación Blowfish (codificador simétrico de bloques, con claves de 32 y 448 bits)<sup>1</sup> y encriptación con claves de 56 bits.

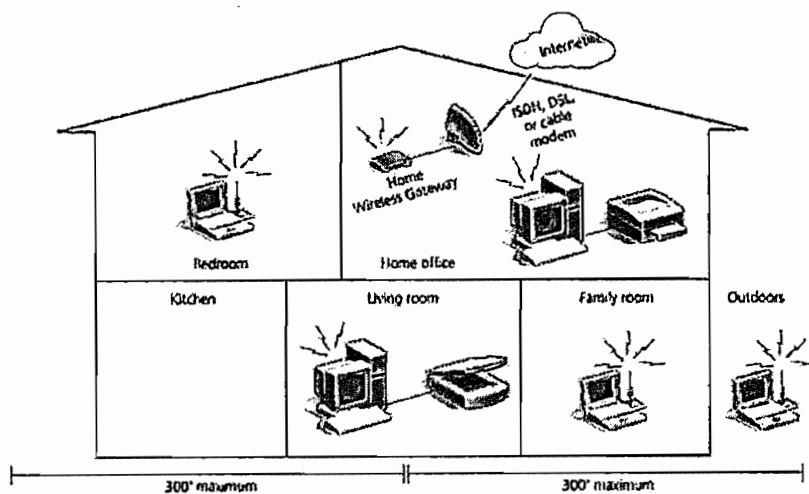


Figura1.5. Ejemplo de tecnología HomeRF

<sup>1</sup> [http://www.batihuevo.tripod.com/III\\_ex\\_parcial.doc](http://www.batihuevo.tripod.com/III_ex_parcial.doc)

#### 1.1.4.3 Estándar HiperLAN2

Desarrollado por el ETSI (European Telecommunications Standard Institute) para redes LAN. Ofrece una velocidad de transmisión de 54 mbits/seg, utiliza la banda de los 5 GHz. Para conseguir estas velocidades, la tecnología HiperLAN2 hace uso de OFDM (Orthogonal Frequency Digital Multiplexing) para transmitir las señales analógicas. Las conexiones emplean multiplexación por división de tiempo y pueden ser punto a punto o punto a multipunto, las primeras son bidireccionales, mientras que las segundas son unidireccionales.

Los puntos de acceso (APs) seleccionan automáticamente el canal de radio adecuado para las transmisiones, soportan autenticación y encriptación, los puntos de acceso y los clientes inalámbricos pueden autenticarse unos a otros para asegurar un acceso autorizado y válido a la red. Todos los datos del usuario viajan encriptados para garantizar la confidencialidad.<sup>2</sup>

El estándar al ser orientado a conexión permite proporcionar calidad de servicio, pudiendo establecer a cada conexión variables como: ancho de banda, el retraso y establece prioridades distintas a cada conexión.

#### 1.1.4.4 Estándar 802.11

802.11 es un estándar de redes inalámbricas (WLAN), desarrollado por el Instituto de Ingenieros Electrónicos y Eléctricos (IEEE), que apareció en el año 1997. En la primera versión del estándar, 802.11, proporcionaba unas velocidades de transmisión de 1 o 2 Mbps y una serie de métodos de señalización y otros servicios. El problema básico de este estándar, fue su baja tasa de transferencia de datos, incapaz de soportar los requerimientos de las redes actuales.

---

<sup>2</sup> [http://wupf.upf.edu/documents/visualizar/Estudio\\_tecnol\\_inal.htm](http://wupf.upf.edu/documents/visualizar/Estudio_tecnol_inal.htm).

En consecuencia se trabajó en un nuevo estándar, el 802.11b (también conocido como 802.11 High Rate), que apareció en 1999 y proporcionaba unas tasas de transferencia de hasta 11 Mbps. Gracias a las prestaciones ofrecidas por 802.11b, similares a las de las redes cableadas 10BaseT, ha tenido una buena acogida en el mundo empresarial, siendo una de las tecnología más expandidas en el mercado.

Varias empresas dedicadas al desarrollo de equipamiento informático se han unido en una alianza denominada WECA (Wireless Ethernet Compatibility Alliance), cuyo objetivo es el de asegurar la interoperabilidad entre productos 802.11b de distintos fabricantes y promocionarlos en el ámbito empresarial y del hogar. Cuando se comprueba que un producto interactúa correctamente con otros dispositivos 802.11b, recibe el certificado **Wi-Fi** (Wireless Fidelity) como garantía de interoperabilidad y buen funcionamiento.

	<b>802.11b</b>	<b>HomeRF</b>	<b>HiperLAN2</b>	<b>Bluetooth</b>
<b>Velocidad</b>	11 Mbps	10Mbps	54Mbps	1Mbps
<b>Distancia</b>	100 – 300m	50m	150m	10-100m
<b>Método de Acceso</b>	Punto – Multipunto	Punto – Multipunto	Punto – Punto Punto - Multipunto	Punto – Multipunto
<b>Aplicación</b>	Todo tipo de red de datos Ethernet	Ámbito domestico	WAN/LAN, voz encapsulada, vídeo, datos	Kits de manos libres para teléfonos, PDA, conexión de periféricos cámaras de fotos, etc

**Tabla 1.2. Resumen de Tecnologías Wireless**

Estándares 802.11 existentes de la familia 802.11 son los siguientes:

- **802.11.** Es el estándar original de WLANs que soporta 1 o 2 Mbps en la banda de 2.4 GHz usando Frequency Hopping Spread Spectrum (FHSS) o Direct Sequence Spread Spectrum (DSSS).

- **802.11a.** Es un estándar de alta velocidad que soporta hasta 54 Mbps en la banda de 5 GHz, usando un esquema de codificación Orthogonal Frequency Division Multiplexing más que FHSS o DSSS.
- **802.11b.** Es el estándar dominante de las WLANs (conocido también como Wi-Fi) que soporta velocidades de hasta 11 Mbps en la banda de 2.4 GHz. Usa solo DSSS. 802.11b fue una ratificación de 1999 al estándar original de 802.11, permitiendo comparar a la funcionalidad inalámbrica con Ethernet.
- **802.11g.** Es otro estándar de las WLANs que soporta velocidades de hasta 20 Mbps en la banda de 2.4 GHz. Las redes inalámbricas tendrán pronto una nueva adición a la familia conforme crecen los planteamientos sobre el futuro de 802.11n.
- **Nuevo estándar 802.11n.** Las principales razones para el desarrollo del nuevo estándar respecto a los anteriores es:
  1. Por la sobrecarga que se produce principalmente por los preámbulos necesarios para cada paquete. Los problemas de sobrecarga se hicieron más agudos con el incremento de la velocidad de transmisión.
  2. La posibilidad de crear paridad en cuanto a capacidad de transmisión de datos entre los sistemas inalámbricos y los no inalámbricos para extender el uso de redes inalámbricas a aquellas áreas en las cuales no se puede utilizar debido a que la tasa de transmisión de datos de los productos inalámbricos existentes es insuficiente.
  3. En los estándares anteriores la porción de datos acarreados dentro de los paquetes se encogió, mientras que la sobrecarga permaneció fija.

### 1.1.5 MODELO DE CAPAS

El estándar 802.11b comprende las capas física y de enlace del modelo OSI, mientras las otras capas no se ven alteradas.

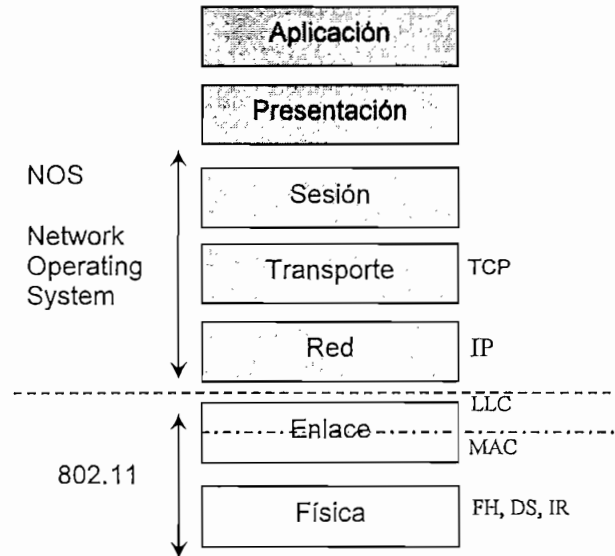


Figura 1.6. Estándar 802.11 dentro del modelo de capas OSI

### 1.1.5.1 Capa Física

La Capa Física de una red define la modulación y señalización de la transmisión de datos. IEEE 802.11 define dos opciones para la capa física: Radiofrecuencia y Luz Infrarroja.

#### 1.1.5.1.1 Radio Frecuencia

La tecnología más utilizada en radiofrecuencias es la de banda ancha o espectro ensanchado.

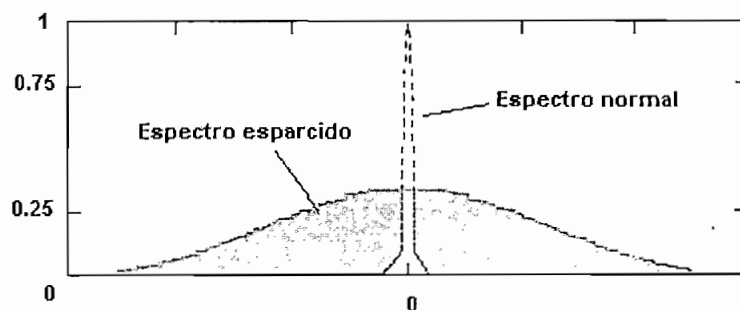


Figura 1.7. Banda Esparcida (Spread Spectrum)

La técnica de espectro ensanchado, utiliza todo el ancho de banda disponible, en lugar de utilizar una portadora para concentrar la energía a su alrededor.

Está diseñada para intercambiar eficiencia en ancho de banda por confiabilidad, integridad y seguridad.

Existen dos tipos de tecnología de espectro ensanchado: Espectro expandido por secuencia directa o DSSS y Espectro expandido por salto de frecuencias o FHSS ambas en la banda de frecuencia 2.4 GHz ISM.

**Espectro expandido por Secuencia Directa:** En este tipo de tecnología los datos fuente a transmitir se someten a una operación OR exclusiva con una secuencia binaria pseudoaleatoria originando una secuencia de salida con una tasa mucho mayor que la tasa de datos fuente. La señal resultante es modulada y transmitida y ocupa una banda de frecuencia mucho mayor (esparcida o dispersa) que la señal original de los datos fuente.

Para otros usuarios de la misma banda, esta señal aparece como (pseudo) ruido. La secuencia de bits utilizada para modular los bits se conoce como secuencia de Barker (también llamado código de dispersión o PseudoNoise (PN)). Cada período de la secuencia PN está formado por N dígitos o "chips" de duración  $T_s$ . Las secuencias PN no se generan libremente, sino que son generadas por métodos artificiales lo que permite la reproducción de secuencias PN idénticas que son imposible de lograr con secuencias aleatorias de cualquier otro tipo.

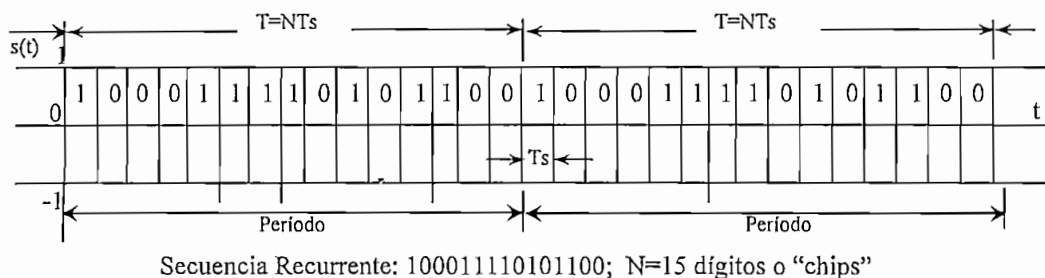


Figura 1.8. Secuencia Binaria Pseudoaleatoria



A continuación podemos observar como se utiliza la secuencia de Barker para codificar la señal original a transmitir:

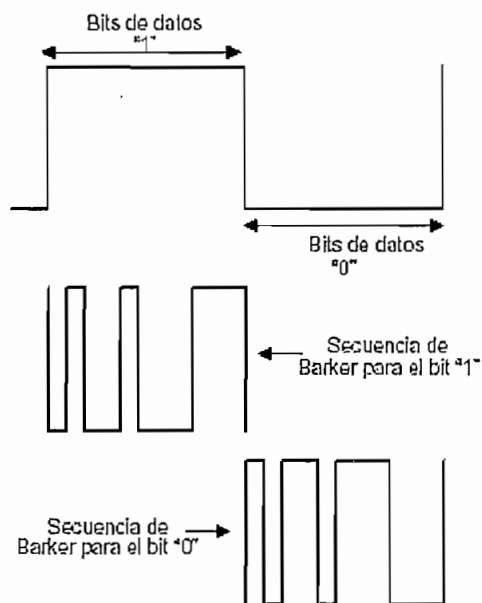


Figura 1.9. Codificación de Barker

Una vez aplicada la secuencia PN, el estándar IEEE 802.11 ha definido dos tipos de modulación para la técnica de espectro ensanchado por secuencia directa (DSSS), la modulación **DBPSK** (Differential Binary Phase Shift Keying) y la modulación **DQPSK** (Differential Quadrature Phase Shift Keying), que proporcionan una velocidad de transferencia de 1 y 2 Mbps respectivamente.

**Espectro expandido por Salto de Frecuencias.** La tecnología de espectro ensanchado por salto de frecuencia (FHSS) consiste en transmitir una parte de la información en una determinada frecuencia durante un intervalo de tiempo denominado dwell time inferior a 400 ms. Cumplido este tiempo se cambia la frecuencia de emisión y se sigue transmitiendo a otra frecuencia. De esta manera cada tramo de información se va transmitiendo en una frecuencia distinta durante un intervalo muy corto de tiempo.

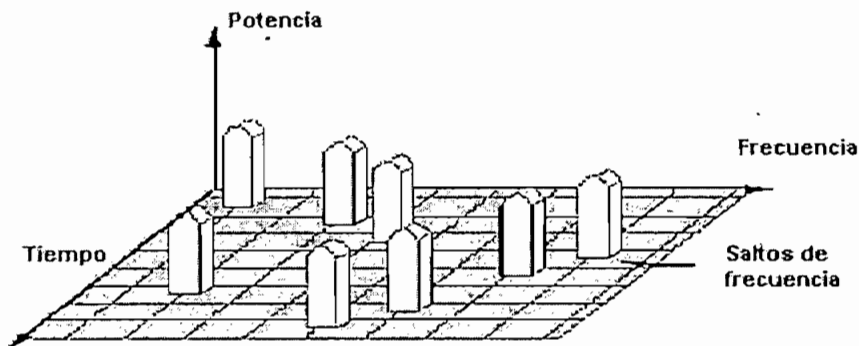


Figura 1.10. Banda Esparcida por Salto de Frecuencia

El orden en los saltos en frecuencia se determina según una secuencia pseudoaleatoria almacenada en unas tablas, y que el emisor y el receptor deben conocer.

Esta técnica también utiliza la banda de los 2.4GHz, la cual se organiza en 79 canales con un ancho de banda de 1MHz cada uno.

El estándar IEEE 802.11 define la modulación aplicable, en este caso se utiliza la modulación en frecuencia FSK (Frequency Shift Keying), con una velocidad de 1Mbps ampliable a 2Mbps.

Los sistemas que usan la técnica de salto de frecuencia consumen menos potencia que los que emplean secuencia directa y generalmente son más económicos. Es importante resaltar que con cualquiera de los dos métodos el resultado es un sistema que es extremadamente difícil de violar, que no interfiere con otros sistemas y que transporta grandes cantidades de información.

#### 1.1.5.1.2 Infrarrojos

Las WLAN por infrarrojos son aquellas que usan el rango infrarrojo del espectro electromagnético para transmitir información mediante ondas por el espacio libre. Los sistemas de infrarrojos se sitúan en altas frecuencias, justo por debajo del rango de frecuencias de la luz visible. Las propiedades de los infrarrojos son, por tanto, básicamente las mismas que tiene la luz visible.

Los sistemas infrarrojos utilizan las siguientes velocidades de transmisión:

- 1 y 2 Mbps, Infrarrojos de modulación directa.
- 4 Mbps, mediante Infrarrojos portadora modulada.
- 10 Mbps, Infrarrojos con modulación de múltiples portadoras.

La tecnología infrarrojo cuenta con muchas características sumamente atractivas:

- Tiene una longitud de onda cercana a la de la luz y comportamiento similar (no puede atravesar objetos sólidos como paredes, por lo que es inherentemente seguro contra receptores no deseados).
- Debido a su alta frecuencia de operación, presenta una fuerte resistencia a las interferencias electromagnéticas artificiales radiadas por dispositivos hechos por el hombre (motores, luces ambientales, etc.).
- La transmisión infrarroja con láser o con diodos no requiere autorización especial en ningún país (excepto por los organismos de salud que limitan la potencia de la señal transmitida).
- Utiliza un protocolo simple y componentes sumamente económicos y de bajo consumo de potencia.

Entre las limitaciones principales de esta tecnología se pueden señalar las siguientes:

- Es sumamente sensible a objetos móviles que interfieren y perturban la comunicación entre emisor y receptor.
- Las restricciones en la potencia de transmisión limitan la cobertura de estas redes a unas cuantas decenas de metros.
- La luz solar directa, las lámparas incandescentes y otras fuentes de luz brillante pueden interferir seriamente la señal.
- Las velocidades de transmisión de datos no son suficientemente elevadas, por ello, lejos de poder competir globalmente con las LAN de radio frecuencia, su

uso está indicado más bien como apoyo y complemento a las LAN ya instaladas (cableadas o por radio).

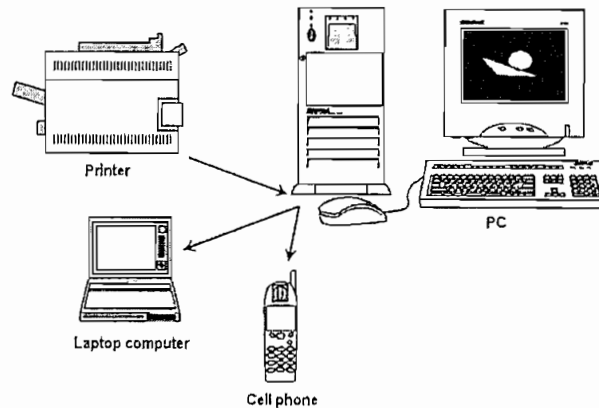


Figura 1.11. Transmisión por infrarrojo

### 1.1.5.2 Capa de Enlace

La capa de enlace de 802.11 está conformada por dos subcapas: LLC (Logical Link Control) y MAC (Media Access Control).

#### 1.1.5.2.1 Capa LLC

La subcapa LLC, se encuentra en la parte superior de la capa de Enlace de Datos, y se encarga de las siguientes funciones:

- Administrar la comunicación de enlace de datos.
- Direccionar los enlaces.
- Definir los Puntos de Acceso al Servicio (SAP's).
- Secuenciación.

La subcapa LLC provee a las capas superiores la forma de conciliar cualquier tipo de protocolo utilizado por la capa MAC, como pueden ser, Ethernet – IEEE 802.3 CSMA/CD ó Token Ring, IEEE 802.5 Token Passing. La subcapa LLC de la capa de Enlace de Datos, administra la comunicación entre los dispositivos a través de un solo enlace a una red. LLC se define en la especificación IEEE 802.2 y soporta tanto servicios orientados a conexión, como

servicios no orientados a conexión, utilizados por los protocolos de las capas superiores<sup>3</sup>.

#### 1.1.5.2.2 Capa MAC

Diseñar un protocolo de acceso al medio para las redes inalámbricas es mucho más complejo que hacerlo para redes cableadas; ya que además de tener en cuenta las dos topologías de una red inalámbrica: (ad-hoc y basadas en infraestructura), se debe también considerar:

- Perturbaciones ambientales (**interferencias**).
- Variaciones en la **potencia** de la señal.
- **Conexiones y desconexiones** repentinas en la red.
- **Roaming**. Nodos móviles que van pasando de celda en celda.

A pesar de todo ello la norma IEEE 802.11 define una única capa MAC (dividida en dos subcapas) para todas las redes físicas.

En las LAN inalámbricas la capa MAC, además de efectuar la función de controlar el acceso al medio, desempeña otras funciones como: *Fragmentación, Control de flujo, Gestión de Potencia y Roaming*

#### 1.1.6 PROTOCOLO IP MÓVIL

Es una tecnología a nivel del protocolo de Red que permite la movilidad de equipos entre diferentes redes sin que estos pierdan su dirección original y que por lo tanto puedan seguir ofreciendo los servicios y recursos prestados. IP Móvil, en su versión actual (versión 4) representa la mejor opción a la movilidad, ya que, ofrece una solución a nivel de Red a diferencia de otros protocolos que lo hacen a nivel Físico y de Enlace; por ejemplo el IEEE 802.11a (Ethernet Inalámbrico "wireless"). La arquitectura del protocolo propuesta por la IETF (Internet

<sup>3</sup> [http://www.htmlweb.net/redes/tcp\\_ip/capa\\_2/acceso\\_red\\_1.html](http://www.htmlweb.net/redes/tcp_ip/capa_2/acceso_red_1.html).

Engineering Task Force), reconoce tres actores: Agente Local o Home Agent, Agente Externo o Foreign Agente y Nodo Móvil.

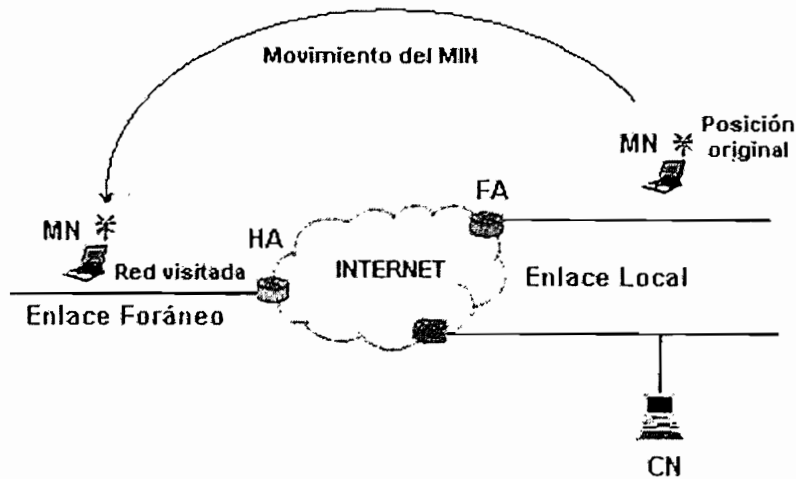


Figura 1.12. Arquitectura Móvil Ipv4

Cuando el nodo móvil está fuera de su red, se identifican cuatro posibles participantes en una comunicación: el nodo móvil (MN); el nodo correspondiente (CN); el agente local (HA); y el agente externo (FA). En este caso, el nodo móvil debe adquirir una care of address "dirección provisional" (posiblemente con el agente externo) y registrarlo con el agente local. El HA retransmite a través de un túnel los mensajes enviados por el nodo correspondiente con destino al nodo móvil, mientras este se encuentra fuera de su red. La figura 1.12 muestra la arquitectura Mobile IP<sup>4</sup>.

#### 1.1.6.1 Funcionamiento

La operación del protocolo Mobile IPv4 puede ser brevemente descrita por los siguientes pasos:

- 1 Los agentes de movilidad (agente local y agente externo) anuncian su presencia mediante los mensajes llamados "**Agent Advertisement**".
- 2 Esos mensajes pueden, opcionalmente, ser solicitados a los agentes de movilidad, a través de mensajes llamados "**Agent Solicitation**".
- 3 Un nodo móvil recibe esos anuncios enviados por los agentes de movilidad y determina si se encuentran en su red local o en una red externa.

<sup>4</sup> <http://www.cujae.edu.cu/revistas/telematica/Articulos/106.htm>.

- 4 Si el nodo móvil detecta que está en su red original, él opera sin el servicio de movilidad. Si acaba de volver a su red, él retira el registro hecho anteriormente con su HA, a través de un intercambio de mensajes "**Registration Request**" y "**Registration Reply**" con el agente.
- 5 Cuando un nodo móvil detecta que se movió para una red externa, él obtiene una dirección provisional (care-of address) de esa red. La dirección provisional puede ser asignada por el agente externo o por otro mecanismo, como DHCP.
- 6 Cuando el nodo móvil está operando fuera de su red, él necesita registrar su dirección provisional con su agente local. Esto se hace a través del intercambio de mensajes **Registration Request** y **Registration Reply**.
- 7 Los datagramas enviados hacia la dirección de origen del nodo móvil (home address), por un nodo correspondiente, son interceptados por el HA, y a través de un túnel son enviados a la dirección provisional de éste, recibidos en la salida del túnel y, finalmente, son entregados al nodo móvil.
- 8 Los datagramas enviados por el nodo móvil son, generalmente, entregados al destino usando los mecanismos de enrutamiento estándar, no necesariamente pasando por el HA.

El funcionamiento del MIP genera un "enrutamiento triangular", o sea, un nodo correspondiente, conociendo apenas el home address del MN, enviará los paquetes hacia la red original. Sin embargo, como el MN se movió, el agente local intercepta los paquetes y a través de un "túnel" los envía al MN en su dirección provisional, o sea, envía el paquete hacia la red donde estuvo momentáneamente. Este hecho es uno de los problemas del MIP, ya que, todos los paquetes serán enviados a través de un túnel hacia el MN en otra red, lo que genera sobrecarga de procesamiento en el HA, además de este ser un punto de falla único en la red.

El MIPv6 (próxima versión) soluciona el problema anterior a través de la optimización de ruta.

### 1.1.7 SEGURIDAD EN LAS COMUNICACIONES LAN INALÁMBRICAS

La seguridad se ha convertido en uno de los principales desafíos para cualquier tecnología. En el caso de las redes locales inalámbricas, garantizar el control de acceso y la privacidad se han constituidos en sus dos grandes retos. Para ello, fabricantes y organismos de estandarización han optado por recurrir a una amplia gama de mecanismos (identificadores SSID, sistemas de encriptación, etc).

La diferencia principal de los entornos inalámbricos con los entornos de cable tradicionales, radica únicamente en el medio en el que se transmiten los datos. Esto hace necesaria la redefinición del concepto de perímetro, ya que en las redes 802.11b, el mismo no está establecido de forma fija, sino que depende del alcance de una señal de radio, algo más complejo de medir.

Esto obliga a replantear varios conceptos tradicionalmente asociados a la seguridad, para ello es necesario conocer a qué riesgos se enfrenta una organización que despliega este tipo de infraestructuras.

#### 1.1.7.1 Riesgos de una Infraestructura Lan Inalámbrica

Son varios los puntos críticos en lo que respecta a la seguridad de una red inalámbrica; a continuación se mostrarán los más esenciales:

**Alcance:** Cuando uno lee las especificaciones de la caja, ve que el alcance de los AP es de 100mts como máximo (especificaciones del estándar 802.11b o Wi-Fi). Gran parte de los usuarios piensan que este limite les brinda una seguridad natural contra “intrusos”, sin embargo esto es totalmente falso.

El alcance de 100 metros es la distancia máxima en que se garantiza que operará el AP a la máxima velocidad, sin embargo en mediciones de campo se puede comprobar que los AP pueden llegar hasta los 600mts, con señales débiles



por supuesto, por lo que un receptor que utilice una antena potente, podría entrar a la red desde esa distancia.

**Exposición de los Puntos de Acceso:** Los AP normalmente operan con un SSID (service set identifier), este código identifica a un grupo de access points y estaciones. Este dato es normalmente transmitido continuamente al aire y enviado en texto puro. Así cualquier laptop con una tarjeta wi-fi, podría leer este dato, copiarlo y hacerse pasar por un integrante de la red.

La mayoría de las veces los AP operan como DHCP, esto es, entregan una dirección IP válida en la red a la estación que la pida, simplemente con un SSID válido. Esto implica que esas estaciones, que podrían estar en el edificio de al frente o una casa a 200 metros del AP, pueden hacer uso de los recursos de la red, ver los discos de los usuarios, el contenido de servidores y navegar por Internet a través del enlace de la empresa.

**Red Abierta:** El estándar de redes inalámbricas incluye un protocolo de seguridad WEP (wireless equivalent protocol), el que permite encriptar las comunicaciones entre los AP y las estaciones. Este protocolo tiene dos opciones en cuanto al largo de la llave a usar en el algoritmo de encriptación: una de 40 bits y otra de 104 (a esto se debe agregar 24 bits del vector de inicialización IV). La de 40 bits es fácilmente "crackeable" en pocos minutos con software de dominio público. La segunda a pesar que es la más recomendada no se salva que sea crackeada.

Otro detalle peor es que la mayoría de los sistemas WEP usa llaves estáticas, esto es, que no cambian con el tiempo lo que ayuda a quienes tratan de detectar la llave WEP.

**Autenticación y Certificación:** Siempre será recomendable proveer un medio de garantizar que el equipo sea quien dice ser, esto es que efectivamente pertenezca a la organización. También hay que considerar un sistema que permita autenticar a los usuarios, mediante una cuenta y clave probar que tienen derecho a los recursos.

**Defensas de Red Inadecuadas:** Es importante tener conciencia que la red inalámbrica tiene los mismos problemas básicos de cualquier red cableada, se pueden tomar todos los resguardos en el lado inalámbrico, pero no sirve de nada si la red no tiene reglas de seguridad mínimas, tales como acceso controlado a los recursos, políticas de autenticación y seguridad de la información sensible, etc.

La carencia de políticas de seguridad es el principal riesgo de pérdida de información. En el caso de las redes inalámbricas la inexistencia de políticas o las malas prácticas hacen que sea tan fácil obtener un SSID o acceso de administrador a un AP sólo leyendo el manual del AP, ya que las cuentas de administrador y el SSID tienen su valor default (por defecto).

#### 1.1.7.2 Mecanismos de Seguridad

Los sistemas wireless utilizan el protocolo **WEP** (Wired Equivalent Privacy) como base de la mayoría de sus mecanismos de seguridad. WEP simula la seguridad que existe en los entornos con cable fijo, que carecen de cifrado en las dos primeras capas OSI, y no supone en ningún momento una solución de seguridad de extremo a extremo. WEP es, por tanto, el encargado de autenticar las estaciones y de cifrar las comunicaciones.

La seguridad de una LAN inalámbrica por consiguiente se sustenta en tres mecanismos: autenticación, filtrados MAC y sistema de encriptación WEP.

**Autenticación** : El primer paso para poder autenticar un cliente en una red wireless es el conocimiento del SSID (Service Set Identifier), el cual proporciona mecanismos que permiten segmentar la red inalámbrica en múltiples redes mantenidas por uno o más puntos de acceso, de forma que cada uno de ellos está programado con su propio identificador.

El SSID realiza una función muy similar a las contraseñas en otras tecnologías debido a que el dispositivo móvil del usuario, a la hora de conectarse a la red, debe contar con el identificador correcto si quiere acceder al punto de acceso. Sin embargo, plantea dudas en cuanto a que puede ser fácilmente conocido y compartido por los usuarios, que en la mayoría de los casos son quienes normalmente configuran sus propios sistemas. Además, también la seguridad puede verse comprometida si el punto de acceso se configura para que emita de forma abierta su propio identificador.

El estándar 802.11b plantea dos posibles formas de autenticación:

- **Open System:** Es el mecanismo de autenticación por defecto, y permite que cualquier estación se una al sistema tras la negociación de los parámetros de red necesarios, es decir, se utiliza autenticación NULA, en la que cualquier dispositivo puede obtener acceso a la red.
- **Shared Key:** Se lleva a cabo mediante un mecanismo de desafío/respuesta cifrado, siendo necesario durante el proceso que ambas estaciones posean una clave común (autenticación simétrica).

**Filtrado de direcciones MAC :** El filtrado de direcciones MAC permite la identificación de los usuarios, de forma que cuando solicitan la conexión a la red son identificados gracias a una dirección MAC recogida en su tarjeta de red inalámbrica. Para su implantación, cada punto de acceso debe ser manualmente programado con la lista de direcciones MAC (que pueden ser variadas cuando se crea conveniente) asociadas a los dispositivos móviles de los usuarios.

**Sistema de Encriptación:** WEP se define como un mecanismo de encriptación y autenticación especificado en el estándar 802.11b que garantiza la seguridad de las comunicaciones entre los usuarios y los puntos de acceso. La norma 802.11b establece una clave de encriptación de 40 bits en la modalidad estándar, también se puede optar por otra de 128 bits. Estas claves son

asignadas tanto de forma estática como manual para todos los clientes y puntos de acceso.

Sin embargo, WEP plantea algunas dudas que pueden poner en riesgo la seguridad de la red. Uno de sus inconvenientes se deriva en que todos los usuarios de la red comparten la misma clave, a lo que hay que sumar que la asignación de las claves de forma estática hace que, si se produce la pérdida o robo de un dispositivo móvil de un usuario, el administrador se ve obligado a cambiar de forma manual la clave en todos y cada uno de los equipos, otros inconvenientes de WEP residen en la dificultad de escalabilidad debido a que sólo soporta claves manuales y a que únicamente es interoperable con otros productos certificados Wi-Fi con encriptaciones de 40 bits.

**Seguridad 802.1x:** A pesar de las deficiencias que presenta el sistema de encriptación y autenticación WEP del estándar 802.11b, WECA recomienda seguir utilizando este sistema pero reforzando la seguridad de estas redes inalámbricas con la amplia gama de herramientas hoy disponibles, como VPN (redes privadas virtuales), IPSec (IP Security), cortafuegos y servidores de autenticación RADIUS.

Sin embargo para grandes corporaciones que poseen mayores demandas de protección, es necesario proporcionar un mejor mecanismo para el control de acceso y la seguridad de la red inalámbrica, como lo es el estándar 802.1x.

Precisamente, el protocolo IEEE 802.1x se ha convertido en el nuevo referente de la seguridad inalámbrica al proporcionar autenticación en el acceso a la red.

La norma 802.1x, cuya mayor escalabilidad le hace idóneo para su uso por las grandes corporaciones, utiliza el protocolo EAP (Extensible Authentication Protocol) como método de encapsulación que permite seleccionar y utilizar conjuntamente diferentes protocolos de autenticación.

## 1.2 REDES DE ACCESO

Una red de acceso engloba todos los elementos encargados de llevar los contenidos multimedia hasta el usuario y atender las peticiones de éste por el canal de retorno.

El primer nivel de la red de acceso es su red troncal de transporte, que se encarga de hacer posible que la red alcance cualquier extensión geográfica. Posterior a ésta se tiene la red de distribución, a través de la que deben llevarse a cabo las tareas de transmisión de datos y conmutación, teniendo como misión principal multiplexar la información proveniente de diferentes proveedores de servicios o distintos usuarios y adaptar el sistema de transporte a las características específicas del bucle de abonado.

### 1.2.1 CLASIFICACIÓN DE LAS REDES DE ACCESO

Se clasifican en dos grandes grupos cableados e inalámbricos:

#### 1.2.1.1 Redes Cableadas

##### *1.2.1.1.1 Las redes de acceso vía cobre*

Este tipo de red está formado por un conjunto de tecnologías que proveen un gran ancho de banda sobre circuitos locales de cable de cobre, sin amplificadores ni repetidores de señal a lo largo de la ruta del cableado, entre la conexión del cliente y el primer nodo de la red. Esta tecnología de acceso punto a punto a través de la red pública, que permiten un flujo de información tanto simétrico como asimétrico y de alta velocidad sobre el bucle de abonado.

##### *1.2.1.1.2 Las redes de acceso vía fibra óptica*

La introducción de la fibra óptica en el nodo de acceso va a permitir disponer de un medio de transmisión de gran capacidad para el soporte de servicios de banda ancha, tanto actual como futuro.

En función de la aplicación particular y de los servicios que serán entregados, podemos encontrar diversas soluciones técnicas. A continuación enumeramos algunas de ellas:

#### *1.2.1.1.2.1 Redes Híbridas Fibra-Coaxial (HFC)*

Una red de acceso HFC está constituida, genéricamente, por tres partes principales: elementos de red, infraestructura HFC, terminal de usuario. El ancho de banda de la red HFC es la clave en la que se fundamentan las ventajas de este tipo de redes, entre las que se incluyen: Posibilidad de ofrecer una amplia gama de servicios tanto analógicos como digitales, Soporte de servicios conmutados y de difusión.

Capacidad de adaptación dinámica a los cambios de la demanda y del mercado, debida, en gran parte, a la gran flexibilidad y modularidad de que están dotadas este tipo de redes.

#### *1.2.1.1.2.2 Redes Ópticas Pasivas (PON)*

En este caso la técnica de transmisión más utilizada es la multiplexación por división en longitud de onda **WDM** (Wavelength División Multiplexing) y la configuración punto a punto.

Los usuarios de negocios o comunidades científicas o educativas se suelen conectar a un anillo de distribución SDH que permite velocidades de varios cientos de Mbit/s. Al ser toda la infraestructura de fibra óptica, se proporciona una transmisión muy segura y libre de errores, con una alta capacidad de transferencia si se emplea, por ejemplo, ATM.

#### **1.2.1.2 Las redes de acceso Inalámbrico**

Los sistemas vía radio presentan una alternativa clara a las redes de cable. La ventaja más clara de este tipo de sistemas es la reducción de los costos de infraestructura, además del pequeño margen de tiempo necesario para su

funcionamiento, puesto que en el momento en que se dispone de la antena, se llega inmediatamente a miles de usuarios.

Los sistemas que se presentan y desarrollan en la actualidad para el acceso a los servicios de banda ancha son, fundamentalmente el WLL(Wireless Local Loop), MMDS (Multichannel Multipoint Distribution System) y el LMDS (Local Multipoint Distribution System).

### 1.2.2 REDES DE ACCESO REMOTO

El Acceso Remoto es una de las formas para lograr la comunicación entre computadoras situadas en locaciones distintas. Esto se obtiene mediante el uso de un medio de comunicaciones cableado como la línea telefónica o a su vez por medios Inalámbricos.

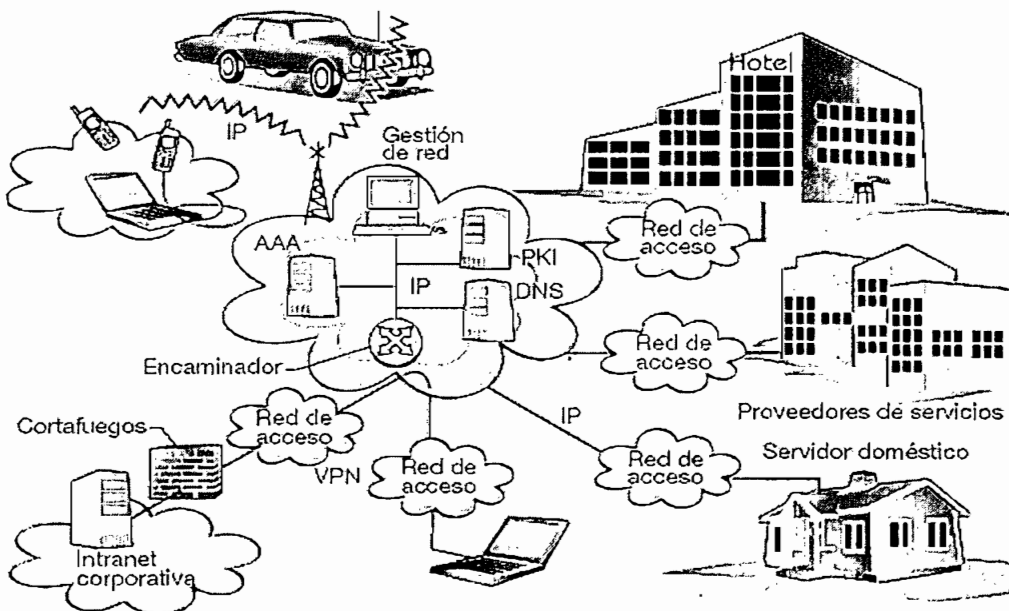


Figura 1.13. Muestra las diferentes formas de Acceso remoto Inalámbrico o por cable.

#### 1.2.2.1 Acceso remoto Telefónico

Este acceso permite al cliente acceder de forma remota y segura a la LAN de la oficina principal, utilizando para ello una línea telefónica.

El cliente debe disponer de un nombre de usuario y un password que le permiten ser autenticado de una forma adecuada. Permite el acceso vía línea telefónica analógica, RDSI y vía móvil.

Sabemos que mediante las líneas telefónicas es posible interconectar dos puntos cualquiera del planeta sin importar que tan distante estén uno del otro, lo cual las hace un buen medio para las comunicaciones; sin embargo, por estar diseñado exclusivamente para la transmisión de voz presenta problemas al momento de ser utilizado para la transmisión de señales digitales.

De hecho, si se trata de transmitir una señal digital ésta llegaría tan distorsionada al otro extremo, la cual sería imposible de reconocer. Entonces, para aprovechar ésta capacidad instalada para la transmisión de señales digitales, la solución fue la implementación de un dispositivo que convirtiera esas señales digitales generadas por el computador en señales analógicas que pudieran ser enviadas por la línea telefónica y luego en el otro extremo, convertir esas señales e introducirlas al computador receptor.

El dispositivo capaz de efectuar esta conversión es el Módem (modulador y demodulador). Estos dispositivos no sólo facilitan el proceso de transmisión sino que además proporcionar una serie de características adicionales que ayudan en la comunicación. Entre estas características está el re-discado cuando el número está ocupado, contestar llamadas, corrección y detección de errores, selección automática de la velocidad de conexión, etc.

#### **1.2.2.2 Acceso remoto Inalámbrico.**

Este servicio permitirá la conexión a red de equipos portátiles con tarjetas inalámbricas a los usuarios.

Gracias a las comunicaciones inalámbricas, los usuarios disfrutaran de servicio siempre conectado (siempre conectado, siempre en línea). De esta forma los usuarios experimentarán servicios de IP constantes cuando se muevan del acceso inalámbrico al fijo. Los servicios mismos se extenderán por diferentes



redes, asignado a lo largo de sus caminos diversos recursos, tales como proxies, agentes de movilidad, y agentes intermediarios.

Para conectar a la red deberá seguir los siguientes pasos:

- Asegurarse de que tiene los drivers de su tarjeta de red y el protocolo TCP/IP instalado en su equipo y con la configuración IP automática.
- Conecte su equipo a la red inalámbrica. En caso de encontrar problemas en la conexión, utilice como SSID.
- En ese momento su ordenador se encuentra en una red con direccionamiento privado y sin acceso ni a los servidores ni a Internet.

Par conseguir dicho acceso, debe realizar una conexión de red privada virtual (VPN). Este tipo de conexión garantiza que su comunicación sea encriptada, de forma que no pueda ser capturada por el resto de usuarios.

### **1.2.2.3 Protocolos de Acceso remoto**

Se pueden establecer conexiones RAS mediante el protocolo Internet de línea serie (SLIP) o el protocolo punto a punto (PPP).

#### ***1.2.2.3.1 Protocolo Internet de línea serie (SLIP)***

El protocolo Internet de línea serie (SLIP) es un estándar para conexiones TCP/IP efectuadas a través de líneas seriales. SLIP es compatible con el Acceso telefónico a redes de Windows NT y ofrece a los clientes de Windows NT un acceso más sencillo a los servicios de Internet. SLIP tiene varias limitaciones:

Requiere una dirección IP estática, por lo que los servidores SLIP no pueden utilizar DHCP o el Servicio de nombres de Internet para Windows (WINS, Windows Internet Name Service).

Normalmente, está basado en sesiones en modo texto y por eso requiere un sistema de archivos de comandos para automatizar el proceso de inicio de sesión.

#### *1.2.2.3.2 Protocolo punto a punto (PPP)*

El protocolo punto a punto (PPP) se diseñó como una mejora a la especificación original, SLIP. PPP es un conjunto de protocolos de empaquetado y autenticación estándar de la industria que permiten que los clientes y servidores RAS operen en redes heterogéneas. Proporciona un método estándar para el envío de datos de red a través de un vínculo punto a punto.

PPP es compatible con varios protocolos, entre los que se incluyen Macintosh AppleTalk, DEC DECnet, Interconexión de sistemas abiertos OSI (Open Systems Interconnection), TCP/IP, IPX y NetBEUI. Windows NT es compatible con NetBEUI, TCP/IP e IPX.

#### *1.2.2.3.3 Protocolo de enlace múltiple PPP (MP)*

El protocolo de enlace múltiple PPP proporciona un medio para aumentar la velocidad de transmisión al combinar varios enlaces físicos en un único enlace virtual que aumenta el ancho de banda. RAS y el protocolo de enlace múltiple PPP pueden utilizarse para combinar rutas de módems analógicos, rutas ISDN e incluso enlaces de comunicaciones mixtos, analógicos y digitales, en los clientes y en los servidores.

Por ejemplo, un cliente con dos módems de 28800 bps y dos líneas PSTN, puede utilizar el protocolo de enlace múltiple PPP para establecer una conexión a 57600 bps con un servidor que ejecute el protocolo de enlace múltiple PPP. Esto aumentará la velocidad de acceso a Internet o a la Intranet, y reducirá el tiempo de establecimiento de la conexión remota, con lo que disminuye el costo del acceso remoto, para utilizar este protocolo, tanto el cliente de Acceso telefónico a redes como el servidor de RAS deben tener activado el protocolo de enlace múltiple PPP.

## 1.3 COMUNICACIÓN SERIAL

Existen dos formas de intercambiar información binaria: paralela y serial. La comunicación paralela transmite todos los bits de un dato de manera simultánea y tiene la ventaja que la transferencia es rápida, pero la desventaja de necesitar una gran cantidad de hilos o líneas, situación que encarece los costos, además surgen problemas de comunicación cuando la distancia es grande, debido a que la capacitancia entre los conductores limita el correcto intercambio de datos a unos pocos metros. La comunicación serial por su parte transmite un bit a la vez, por lo tanto es más lenta, pero posee la ventaja de necesitar un menor número de líneas para la transferencia de la información y las distancias a las cuales se puede realizar el intercambio de datos es mayor.

### 1.3.1 NORMAS DE COMUNICACIÓN SERIAL

Para la comunicación de datos en serie la Unión Internacional para la Reglamentación de las Telecomunicaciones (UIT) dispone del Comité Consultivo Internacional de Telegrafía y Telefonía (CCITT), el cual establece una serie de normas para diferentes campos de las comunicaciones, dentro de estas descripciones de normatividad y control de las telecomunicaciones está el libro naranja para la telegrafía y la telefonía que contiene a la serie "X" para las redes y circuitos especializados en la transmisión de datos, estas normas se designan como RS-232, RS-422, RS-423 y RS-485.

#### 1.3.1.1 Norma RS-232

En 1969 la Asociación de Industrias Electrónicas (EIA), conjuntamente con los Laboratorios Bell y los fabricantes de equipos de comunicaciones, formularon el EIA RS-232- C<sup>5</sup>. El propósito inicial fue la conexión entre un DTE (Equipo terminal de datos) y un DCE (Equipo terminal del circuito de datos), empleando un intercambio de datos binarios en serie.

---

<sup>5</sup> <http://eq3.uab.es/personal/baeza/comunicaciones/comunica.htm>

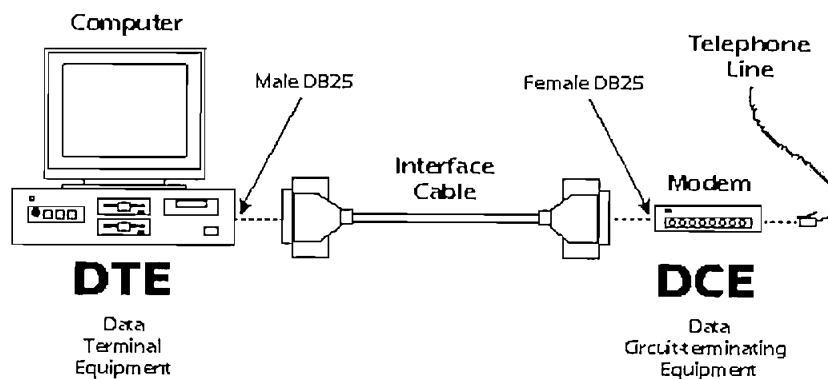


Figura 1.14. Conexión entre un DTE y un DCE

Con respecto a las características mecánicas del estándar RS-232, existen dos tipos de conectores, DB25 (con 25 terminales) y DB9 (con 9 terminales). Sin embargo actualmente la mayoría de los equipos utilizan éste último. En la figura 1.15 se muestran gráficamente los conectores.

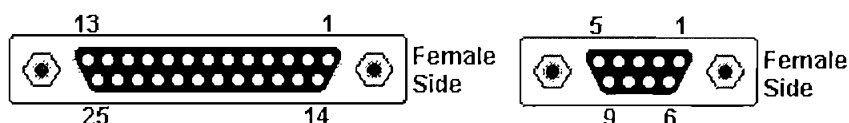


Figura 1.15. Conector DB25 (derecha) y DB9 (izquierda)

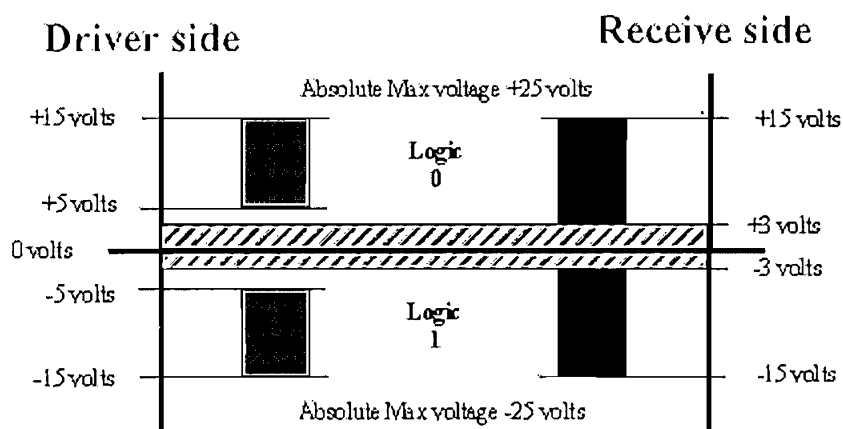


Figura 1.16. Niveles de voltaje utilizados en la norma RS-232

En la figura 1.16 se detallan los niveles de voltaje utilizados por este estándar, observándose claramente que se tiene una lógica negativa de asignación de niveles.

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor es pasiva (idle) y permanece en un estado alto. Para empezar a transmitir datos, el transmisor coloca esta línea en bajo durante un tiempo de un bit, lo cual se conoce como bit de inicio (start bit) y a continuación empieza a transmitir con el mismo intervalo de tiempo los bits correspondientes al dato (que pueden ser 7 u 8 bits), empezando por el menos significativo (LSB) y terminando con el más significativo (MSB). Al finalizar se agrega el bit de paridad (parity) si es que está activada esta opción, y los bits de parada (stop) que pueden ser 1 o 2, en los cuales la línea regresa a un estado alto. Cuando esta operación concluye, el transmisor está preparado nuevamente para transmitir el siguiente dato. Entonces el receptor no está sincronizado con el transmisor y desconoce cuando va a recibir datos.

La transición de alto a bajo de la línea del transmisor activa al receptor y este genera un conteo de tiempo de tal manera que realiza una lectura de la línea medio bit después del evento; si la lectura realizada es un estado alto asume que la transición ocurrida fue ocasionada por ruido en la línea; si por el contrario la lectura es un estado bajo, considera válida la transición y empieza a realizar lecturas secuenciales a intervalos regulares de tiempo hasta conformar el dato recibido. El receptor puede tomar el bit de paridad al igual que los bits de parada para determinar la existencia o no de errores y realizar las acciones correspondientes, físicamente la transmisión de datos se realiza cambiando la diferencia de potencial entre la línea transmisora y la línea de tierra. En la recepción de datos se realiza el proceso inverso, las diferencias de potencial de la línea transmisora se detectan entre la línea receptora y la línea de tierra. Para conectar dos dispositivos DTE sin señales de acoplamiento, deben efectuarse las conexiones descritas en la figura 1.18.

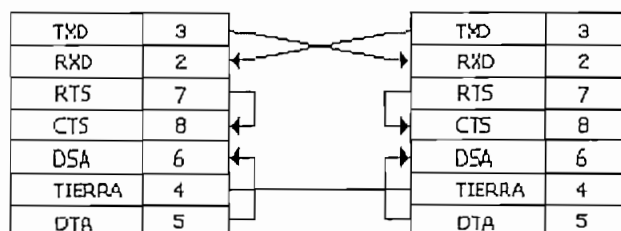


Figura 1.18 Conexión entre dos equipos RS-232 DTE sin señales de acoplamiento



### 1.3.1.3 Norma RS-485

El modelo EIA-RS-485 es similar al RS-422 con respecto a la longitud y velocidad de transmisión, pero permite el alargamiento de la red en otros 1200 metros insertando un repetidor RS-485 en la línea. Dentro del estándar RS-485 existen diferentes variantes, una de ellas es conocida como 4D-RS-485, donde los pares de cables para transmisión y recepción se mantienen por separado, además soporta hasta 30 equipos emisores/receptores conectados en cada segmento de la red, como se muestra en la figura 1.20.

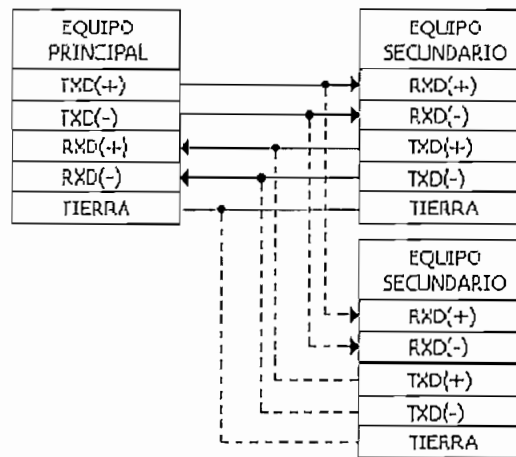


Figura 1.20. Conexión entre equipos 4D-RS-485.

El estándar 2D-RS-485 comúnmente conocido como estándar RS-485, utiliza una sola línea de transmisión balanceada bidireccional y la diferencia con el anterior es que los dispositivos deben conmutar entre modo receptor y modo transmisor.

En este modelo sólo uno de los puntos puede iniciar una comunicación, debido a que pueden conectarse hasta 30 equipos diferentes, por lo que no pueden ocupar la línea de envío simultáneamente. En este caso, el equipo principal (maestro), puede mandar datos en cualquier momento, mientras que los equipos secundarios (esclavos), únicamente deben responder cuando el mensaje va dirigido a ellos.

Por otra parte, los equipos deben estar siempre en modo receptor, excepto en el momento que transmitirán, en el cual pasan a modo de envío. El equipo principal debe cambiar a modo de envío cuando mande datos a un equipo secundario. Una vez finalizado el envío de datos, debe pasar a modo de recepción y esperar a que el equipo secundario le devuelva una respuesta a su mensaje, en ambos casos la línea de tierra común es opcional, pero debe utilizarse en conexiones donde puedan existir interferencias.

### 1.3.2 COMPARACIÓN ENTRE NORMAS DE COMUNICACIÓN SERIAL.

En la tabla 1.4. se muestra una comparación entre los estándares más usuales de comunicación serial. No se incluye el estándar RS-485 debido a que sus características sobrepasan las requeridas en este sistema de comunicación.

Con base en sus características y en la tabla 1.4, la norma RS-422 resulta ser la más eficiente para la transmisión serial entre el subsistema de control y la central, ya que si el microcontrolador se configura para transmitir a una velocidad de 9600 baudios, es posible establecer una transmisión hasta una distancia de 12 mil metros, incrementando la posibilidad de ampliar las aplicaciones del sistema.

PARÁMETROS	RS232C	RS422A	RS423A
Longitud máxima de la línea	15 m	1220 m	1220 m
Velocidad de transmisión	9600 baudios a 15m	10 Mbaudios/12 m 1 Mbaudios/122 m 100 Mbaudios/122 m	100 Kbaudios/10 m 10 Kbaudios/90 m 1Kbaudios/1220 m
Modo de transmisión	Simple	Diferencial	Diferencial
Nivel Lógico "0"	+3 V ≤ "0" ≤ +12 V	A < B	+4 V ≤ "0" ≤ +6 V
Nivel Lógico "1"	-3 V ≤ "1" ≤ -12 V	A > B	-4V ≤ "1" ≤ -6 V
Número de receptores permitidos en línea	1	10	10
Impedancia de entrada	De 3 – 7 K Ω en 2500 pF	>4 K Ω	>4 K Ω
Impedancia de salida		<100 Ω	<50 Ω

Tabla 1.4. Comparación entre los estándares RS-232C, RS-422A y RS-423A.



## **1.4 HERRAMIENTAS UTILIZADAS**

### **1.4.1 PROTOCOLOS TCP/IP**

El conjunto de protocolos TCP/IP se desarrolló como parte de la investigación realizada por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA), originalmente, se desarrolló para proveer comunicaciones a través de DARPA. Posteriormente, TCP/IP se incluyó en el set de distribución del Software Berkeley de UNIX.

TCP/IP permite la comunicación entre cualquier conjunto de redes interconectadas y sirve tanto para las comunicaciones LAN como para las de WAN. TCP/IP incluye no sólo las especificaciones de las Capas 3 y 4 (como, por ejemplo, IP y TCP) sino también especificaciones para aplicaciones tan comunes como el correo electrónico, la conexión remota, la emulación de terminales y la transferencia de archivos.

#### **1.4.1.1 Arquitectura**

La función de la pila, o conjunto, de protocolo TCP/IP es la transferencia de información desde un dispositivo de red a otro. TCP/IP está conformado por cuatro capas: Acceso a la red, Internet, Transporte y Aplicación.

Dentro de las cuatro capas se incluyen varios tipos de protocolos que tienen diferentes funciones, todas relacionadas con la transferencia de información, algunos de ellos se mencionan a continuación:

##### ***1.4.1.1.1 Capa Acceso a la red.***

La capa de acceso a la red determina la manera en que las estaciones envían y reciben la información a través del soporte físico proporcionado por la capa inferior.

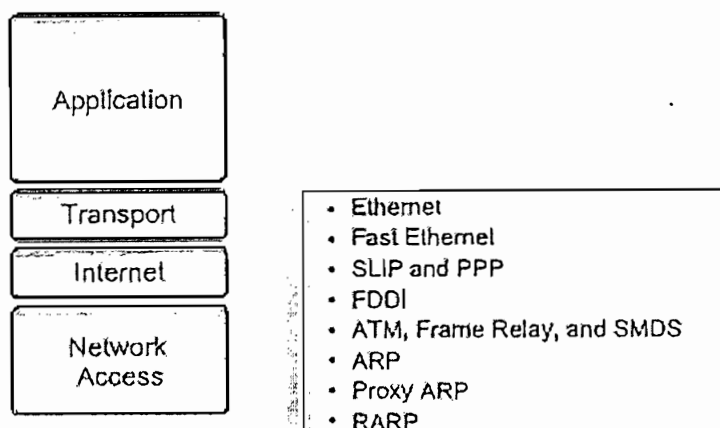


Figura 1.21. Capa Acceso a la red

#### 1.4.1.1.2 Capa Internet.

Esta capa tiene la responsabilidad de transportar paquetes a través de una red utilizando el direccionamiento por software.

Como se muestra en la figura 1.22, varios protocolos operan en la capa Internet de TCP/IP, que corresponde a la capa de red del modelo OSI:

**IP:** suministra enrutamiento de datagramas no orientado a conexión, de máximo esfuerzo de entrega; no se ocupa del contenido de los datagramas; busca la forma de desplazar los datagramas al destino.

**ICMP:** aporta capacidad de control y mensajería.

**ARP:** determina direcciones a nivel de capa de enlace de datos para las direcciones IP conocidas.

**RARP:** determina las direcciones de red cuando se conocen las direcciones a nivel de la capa de enlace de datos.

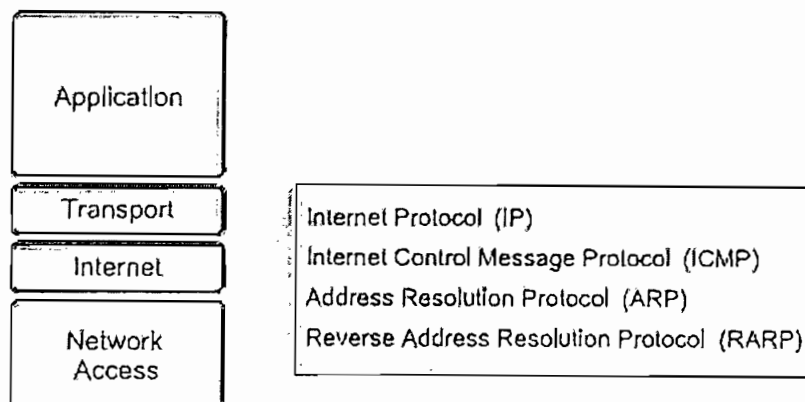


Figura 1.22. Capa Internet

#### 1.4.1.1.3 Capa Transporte

Permite que un dispositivo de usuario divida en segmentos varias aplicaciones de capas superiores para colocarlas en el mismo flujo de datos de la Capa 4, y permite que un dispositivo receptor pueda reensamblar los segmentos de las aplicaciones de las capas superiores. Los datos de Capa 4 es una conexión lógica entre los extremos de una red, y brinda servicios de transporte desde un host hasta un destino. Este servicio a veces se denomina servicio de extremo a extremo.

La capa de transporte también proporciona dos protocolos:

**TCP:** un protocolo confiable, orientado a conexión; suministra control de flujo a través de ventanas deslizantes, y confiabilidad a través de los números de secuencia y acuses de recibo. TCP vuelve a enviar cualquier mensaje que no se reciba y suministra un circuito virtual entre las aplicaciones del usuario final. La ventaja de TCP es que proporciona una entrega garantizada de los segmentos.

**UDP:** protocolo no orientado a conexión y no confiable; aunque tiene la responsabilidad de transmitir mensajes, en esta capa no se suministra ninguna verificación de software para la entrega de segmentos. La ventaja de UDP es la velocidad. Como UDP no suministra acuses de recibo, se envía menos cantidad de tráfico a través de la red, lo que agiliza la transferencia.

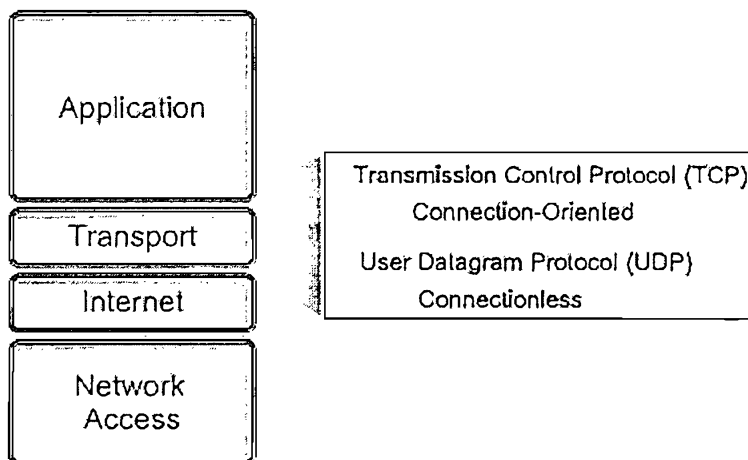


Figura 1.23. Capa Transporte

#### 1.4.1.1.4 Capa Aplicación

Soporta los protocolos de direccionamiento y la administración de red. Además posee protocolos para transferencia de archivos, correo electrónico y conexión remota:

**DNS:** (Sistema de nombre de dominio) es un sistema utilizado en Internet para convertir los nombres de los dominios y de sus nodos de red publicados abiertamente en direcciones IP.

**WINS** (Servicio de nombre para Internet de Windows) es un estándar desarrollado para Windows NT de Microsoft que asocia las estaciones de trabajo NT con los nombres de dominio de Internet de forma automática.

**HOSTS:** es un archivo creado por los administradores de red que se mantiene en los servidores. Se utiliza para suministrar asignación estática entre direcciones IP y nombres de computadores.

**POP3:** (Protocolo de la oficina de correos) es un estándar de Internet para almacenar correo electrónico en un servidor de correo hasta que se pueda acceder a él y descargarlo al computador. Permite que los usuarios reciban correo desde sus buzones de entrada utilizando varios niveles de seguridad.

**SMTP:** (Protocolo simple de transferencia de correo) maneja la transmisión de correo electrónico a través de las redes informáticas. El único soporte para la transmisión de datos que suministra es texto simple.

**SNMP:** (Protocolo simple de administración de red) es un protocolo que suministra un medio para monitorear y controlar dispositivos de red, y para administrar configuraciones, recolección de estadísticas, desempeño y seguridad.

**FTP:** (Protocolo de transferencia de archivos) es un servicio confiable orientado a conexión que utiliza TCP para transferir archivos entre sistemas que soportan FTP. Soporta transferencias bidireccionales de archivos binarios y archivos ASCII.

**TFTP:** (Protocolo trivial de transferencia de archivos) es un servicio no confiable no orientado a conexión que utiliza UDP para transferir archivos entre sistemas que soportan el Protocolo TFTP. Es útil en algunas LAN porque opera más rápidamente que FTP en un entorno estable.

**HTTP:** (Protocolo de transferencia de hipertexto) es el estándar Internet que soporta el intercambio de información en la World Wide Web, así como también en redes internas. Soporta muchos tipos de archivos distintos, incluyendo texto, gráfico, sonido y vídeo. Define el proceso a través del cual los navegadores de la Web originan solicitudes de información para enviar a los servidores de Web.

**TELNET:** es un protocolo estándar de emulación de terminal utilizado por los clientes con el propósito de realizar conexiones de terminal remota con los servicios del servidor Telnet; permite que los usuarios se conecten de forma remota con los routers para introducir comandos de configuración.

**PING:** (Packet Internet Groper) es una utilidad de diagnóstico que se utiliza para determinar si un computador está conectado correctamente a los dispositivos o a Internet.

**TRACEROUTE:** es un programa que está disponible en varios sistemas y es similar a PING, excepto que traceroute suministra más información que PING. Traceroute rastrea la ruta que toma un paquete hacia el destino y se utiliza para depurar problemas de enrutamiento.

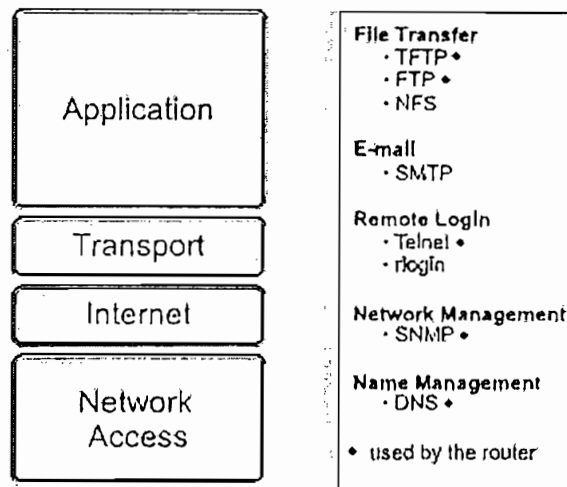


Figura 1.24. Capa Aplicación

## 1.4.2 LENGUAJE ASSEMBLER

### 1.4.2.1 Introducción

Programar en Assembler no es más que escribir una serie de códigos entendibles por el usuario que posteriormente serán convertidos en código de máquina que lo ejecutará el microprocesador.

La programación en Assembler requiere ante dos instrucciones que obtengan el mismo resultado se debe elegir aquella que tenga menos ciclos de máquina o de reloj, o aquella que ocupe menos localidades de memoria; incluso en algunos casos habrá que elegir entre ocupar menos posiciones o ser más rápidos, en función de las necesidades que se tengan.

Esto no quiere decir que sea necesario conocer de memoria los ciclos de cada instrucción; un manual de Assembler debe contener toda la información necesaria, con un método de acceso fácil, a pesar de que en algún caso resulte redundante.

Se recomienda cuando se programa en Assembler poner comentarios; Los comentarios sólo ocupan lugar en el código simbólico o programa fuente; cualquier ensamblador los ignora cuando convierte el programa en código de máquina.

Las Subrutinas son importantes porque se trata de un problema de estructura del programa. Las ventajas son múltiples; una estructura de subrutinas es más fácil de entender, por lo tanto de modificar.

Se da con frecuencia el caso de necesitar en un programa operaciones iguales o semejantes a las de otro, por lo tanto, con limitarse a copiar totalmente estas partes o como mucho, adaptarlas algo a las características del nuevo programa.

#### 1.4.2.2 Etapas para la realización de un programa

Existen una serie de pasos que se deben seguir para dar por definitiva la realización de un programa, los pasos a seguir son:

**Planteamiento del problema:** Conocido como análisis funcional.

Es la definición de lo que se quiere realizar, la información de que se dispone, la información que se quiere obtener, la información que se quiere volver a utilizar, los formatos de pantalla deseados; todas estas definiciones se deben hacer con el máximo detalle para poder explicar lo que se desee hacer.

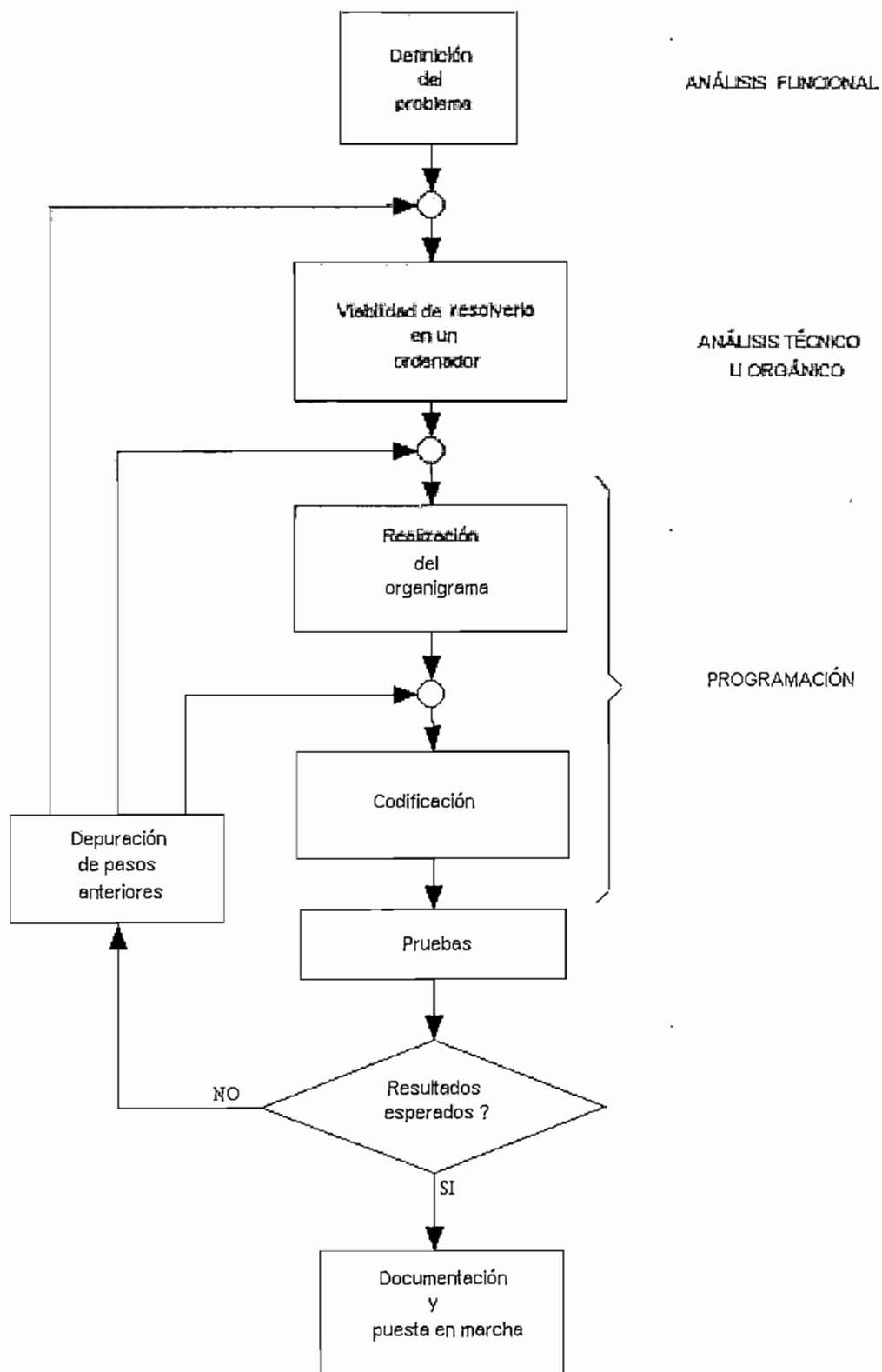


Figura 1.25. Diagrama de Pasos para la realización de un programa



**Viabilidad de resolver el problema con un ordenador.** Se lo conoce como análisis técnico u orgánico. En este paso se valorará la posibilidad de resolver el problema con el ordenador de que se dispone, así como de la periferia a nuestro alcance (impresora, microdrive, etc.). Se definirá toda la información del paso anterior como datos entendibles para el ordenador, indicando en qué código irán, cuánta memoria ocuparán, en qué lugar han de almacenarse y cómo se procesarán; lo mismo para los diseños de pantalla. En este paso es donde se debe decidir qué procesos se hacen por medio de subrutinas, y como curiosidad, si el programa se hace en BASIC o en ASSEMBLER.

**Realización del programa:** Es decir, la programación propiamente dicha. Este paso se divide en dos partes: la construcción del organigrama y la codificación. Estos dos pasos son complementarios. El organigrama es una construcción gráfica del flujo o caminos posibles que tiene el programa. La codificación es la escritura del lenguaje simbólico contemplando todas las alternativas posibles definidas en el organigrama.

**Prueba del programa:** En este paso se preparan unos datos de entrada al programa de los cuales ya se conocen los resultados que obtienen, contemplando, si es posible, todos los casos.

**Documentación:** Una vez concluidos los pasos anteriores, se reúne todo el material e incluso se comentan los problemas o dificultades encontrados y cómo se han solucionado. Ver Figura 1.25.

#### 1.4.2.3 Código de Máquina

Lo que se conoce como lenguaje o código de máquina es el conjunto de códigos numéricos (comúnmente expresados en hexadecimal) para cada una de las instrucciones que el microprocesador es capaz de reconocer y ejecutar.

El lenguaje Assembler fue desarrollado para liberar al programador de la difícil tarea de tener que recordar todos los códigos y de las tediosas labores de cálculo de localidades de memoria. Los códigos nemónicos del ensamblador son

una sustitución de los códigos numéricos mejor que en una relación 1:1. Por ejemplo, resulta que mientras tenemos de manera general un nemónico para cargar un registro con un valor (MOV) internamente, para cada una de las modalidades de direccionamiento o registro afectado existe un código numérico.

La programación del computador en lenguaje de máquina, tanto la introducción de los programas, su ejecución y su depuración es posible realizarla usando alguna utilidad provista por el sistema operativo.

#### 1.4.2.4 Ejecución

Para la ejecución del programa simplemente basta teclear su nombre en el prompt de comandos y teclear ENTER. Con esto el programa será cargado en memoria y el sistema procederá a ejecutarlo. El proceso completo para poder crear un programa ejecutable con el Microsoft Macro Assembler se muestra en el siguiente cuadro:

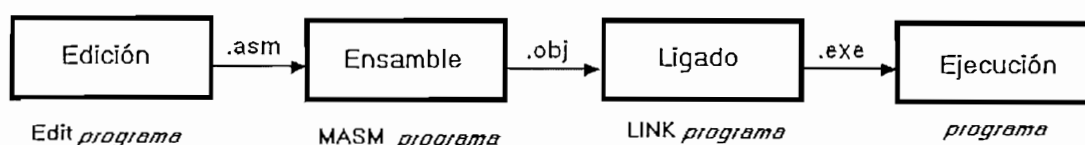


Figura 1.26. Procedimiento de Ejecución

#### 1.4.2.5 Depuración

Para la depuración de un programa en ensamblador tenemos disponibles dos herramientas. Por un lado tenemos el debugger (DEBUG.EXE) y por otro lado tenemos el que nos proporciona Microsoft (SYMDEB.EXE). Una de ellas es la facilidad de desplegar el código fuente correspondiente a la instrucción que se está ejecutando (si el programa ejecutable fue ensamblado o compilado con un ensamblador o compilador compatible), nos permite ejecutar comandos del Sistema Operativo y nos permite obtener información de las interrupciones de manera simbólica.

## 1.5 ETAPAS DEL PROYECTO

### 1.5.1 SISTEMA DE ADQUISICIÓN DE DATOS

Esta etapa del proyecto se la realizará primordialmente en función del circuito integrado de la familia MAXIM, el MAX-180.

#### 1.5.1.1 El MAX 180

El MAX180 es un sistema de adquisición de datos completo, de 12 bits que combina 8 multiplexores para los canales de entrada, gran ancho de banda de Track-and-Hold (T/H) e interfaz flexible para microprocesador con alta velocidad de conversión y bajo consumo de energía.

El MAX180 se puede configurar con un microprocesador para entradas simples o diferenciales. Este dispositivo muestrea y convierte a digital una señal con una tasa de 100kHz y ofrece un interfaz de alta velocidad para microprocesadores de 8 o de 16 bits.

El MAX180 tiene 8 canales de entrada analógicos. La salida del multiplexor del MAX180 se alimenta directamente en la entrada del convertidor analógico-digital. Permite además que un amplificador programable sea insertado entre el MUX y el ADC.

Este sistema permite que el usuario elija entre una referencia interna o externa. Además, el valor de referencia interno y externo se puede ajustar, permitiendo la compensación del sistema con lo que los errores se anularán. El multiplexor tiene una alta impedancia de entrada, simplificando los requerimientos de manejo analógico.

Por todas sus características, el MAX 180 es utilizado para aplicaciones como: equipos automatizados para pruebas, procesamiento de señales digitales, procesos de control que requieren gran exactitud, etc.

El MAX 180 utiliza internamente el método de aproximaciones sucesivas y un circuito de entrada de tipo track-and-hold (T/H) para convertir una señal analógica en una serie de códigos digitales de salida de 12 bits. El control lógico permite una fácil comunicación con el microcontrolador, requiriendo solo unos pocos elementos pasivos para la mayoría de aplicaciones. El proceso T/H no requiere de un capacitor externo. En la figura 1.27 se muestra un circuito típico de operación del MAX-180.

En general, independientemente del modo o interfaz elegido, la siguiente secuencia se presenta internamente cuando una conversión empieza:

1. Los datos ingresan y configuran el latch del sistema de adquisición de datos (DAS), y el interfaz indica al microprocesador que la conversión ha empezado.
2. El mux direcciona la señal de entrada seleccionada a la entrada del T/H.
3. Un retardo completo permite al T/H tomar la señal. En general este tiempo de retardo es tres ciclos de reloj.
4. El T/H conmuta al modo de retención. La salida del T/H entrega a las entradas del conversor analógico digital una muestra simple y estable de la señal de entrada. En la figura 1.28 se muestra el circuito equivalente del T/H.
5. Comienza el ciclo de aproximaciones sucesivas. El conversor ANALÓGICO DIGITAL prueba y ubica 12 bits por vez, desde el bit más significativo hasta el menos significativo. Esto se cumple en 12 ciclos de máquina.
6. El dato de salida es almacenado en los latches correspondientes y el interfaz indica al microprocesador que la conversión está completa y los datos son válidos.

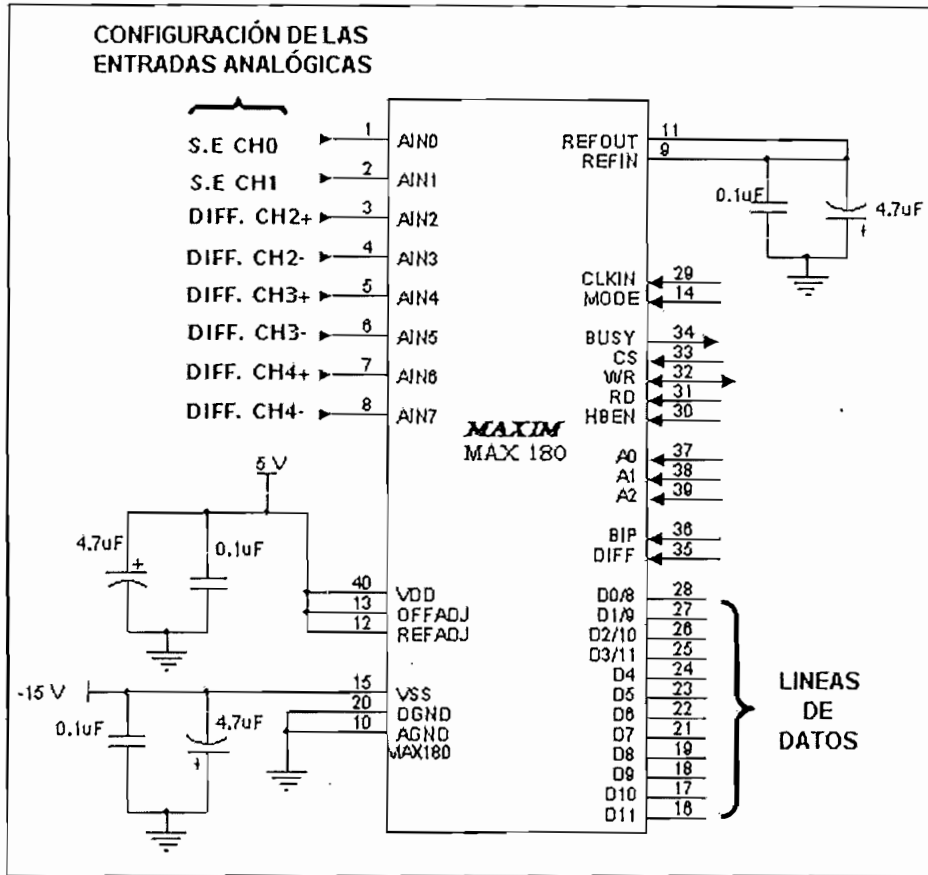


Figura 1.27. Circuito de Operación Típico del MAX 180

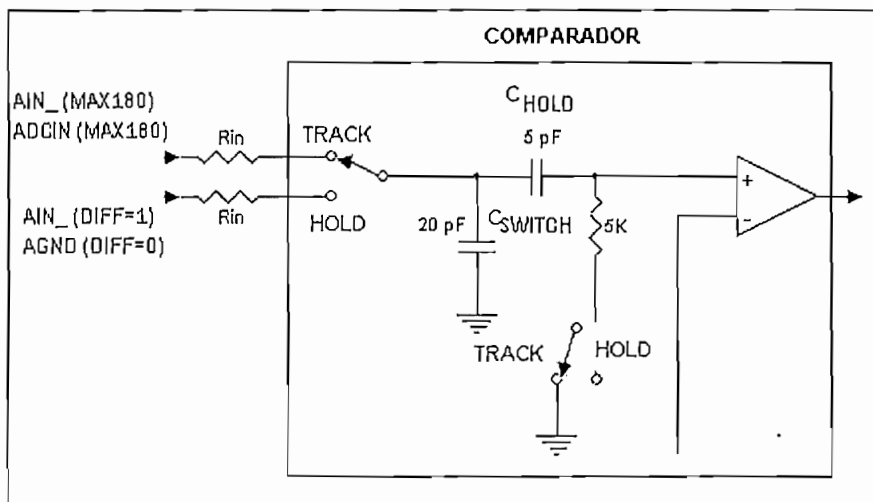


Figura 1.28. Circuito equivalente del comparador analógico del conversor AD (T/H)

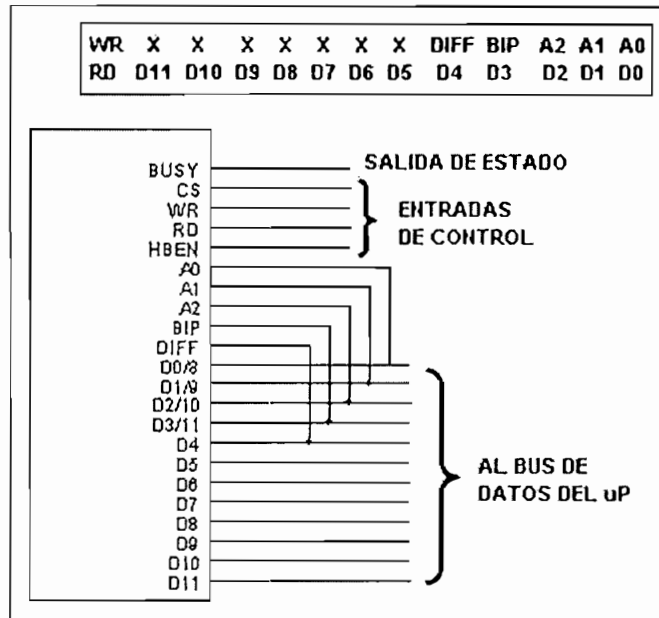


Figura 1.29. Modo Puerto de Entrada/Salida

El MAX 180 posee varios modos de establecer el interfaz externo, dependiendo del dispositivo con el que va comunicarse. El Modo 1 o Modo Puerto de Entrada/Salida es el procedimiento de interés en para proyecto, ya que en este modo las entradas y salidas de datos son usualmente conectadas hacia un microprocesador, y este escribe los datos de configuración en el registro interno del MAX 180, como se muestran en los diagramas de tiempo de las figuras 1.30 y 1.31.

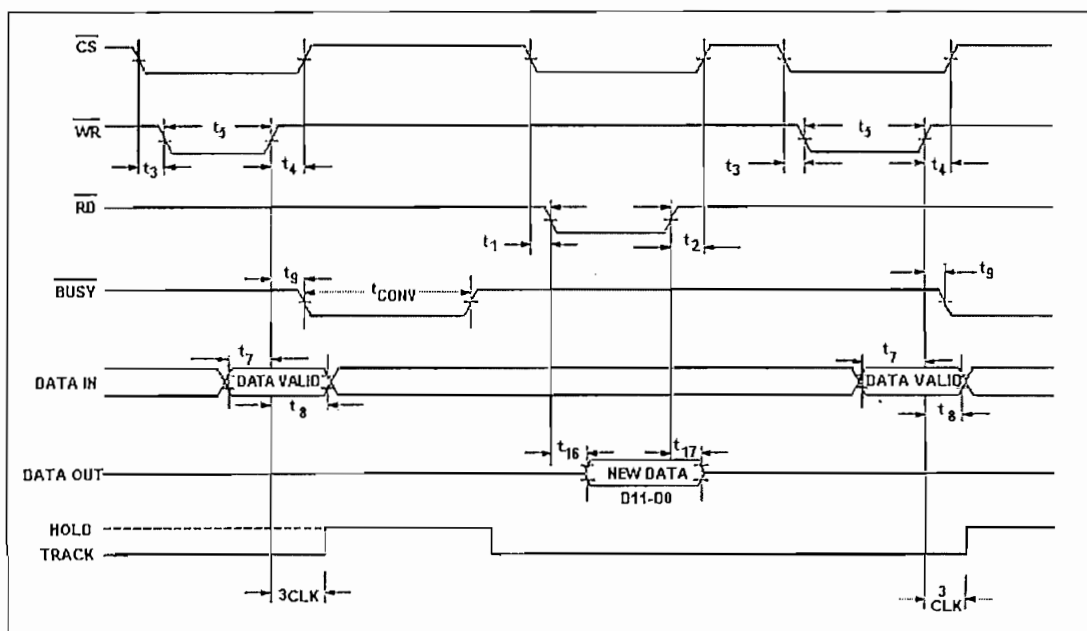


Figura 1.30. Diagrama de tiempo de la lectura paralela (Modo=1, HBEN=0)

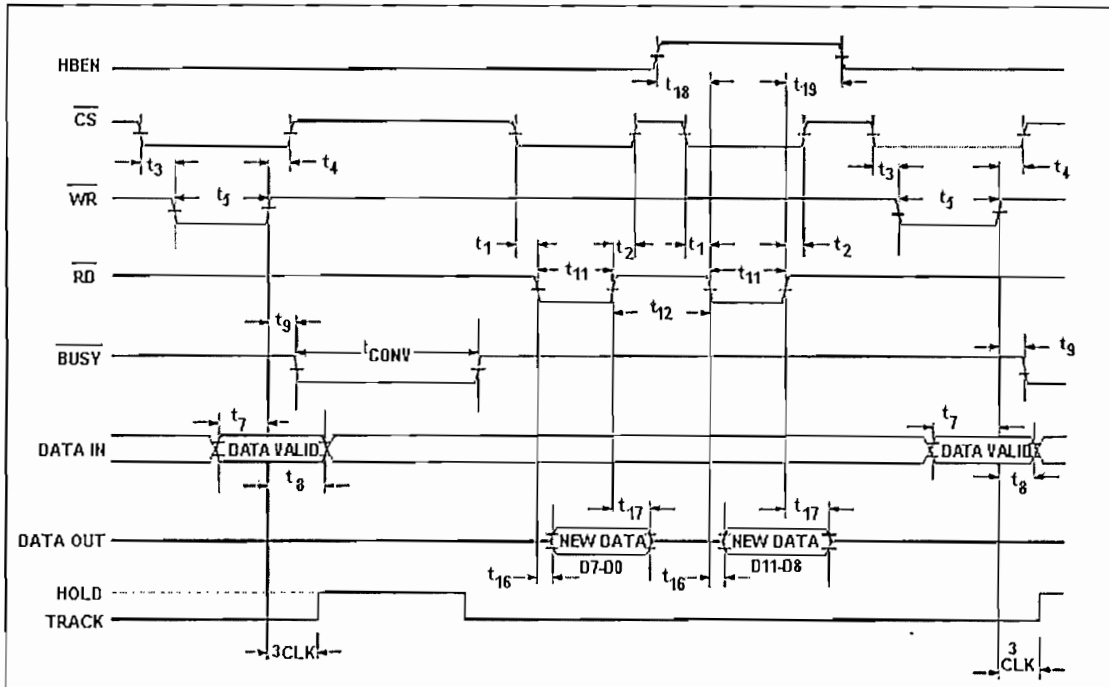


Figura 1.31. Diagrama de tiempo de la lectura de 2 bytes en el Modo 1

Una transición negativa del nivel del pin BUSY indica que una conversión ha comenzado. El multiplexor conecta el canal de entrada seleccionado al T/H, durante 3 ciclos de reloj y 6 la conversión de analógico a digital empieza, 15 ciclos de reloj luego de que el pin WR se puso en alto, el pin BUSY se coloca en alto y los resultados de la conversión aparecen almacenados en los buffers de salida de tres estados. El microprocesador puede luego acceder a los resultados de la conversión mediante una simple instrucción de lectura. Para la operación del bus de 16 bits, se tiene que HBEN=0, y los 12 bits resultantes pueden leerse directamente. Para la operación del bus de 8 bits, HBEN=0 durante la conversión y la instrucción de lectura entrega los 8 bits menos significantes, siendo necesaria una segunda lectura con HBEN=1 para leer los 4 bits más significantes. (En cualquier modo si HBEN=1 se inhabilita el inicio de la conversión).

El registro interno del MAX 180 posee 5 bits: 3 bits para direccionar los canales analógicos, 1 bit para la operación simple o diferencial del mux y un bit para la operación unipolar o bipolar del conversor análogo digital.

Además existen otros dos modos de operación del interfaz del MAX 180: *Modo = 0*, que sirve para comunicarse con periféricos de baja velocidad (modo de memoria lenta), también usado para conectar una memoria ROM con estados de espera del microprocesador (Modo ROM) dentro del que se puede tener una lectura de 12 bits o 2 bytes; el último modo denominado *Modo Asynchronous Hold (Modo = Open)* sirve para comunicarse con un dispositivo que requiera alta precisión o sincronización del tiempo de muestra, difiriendo con el *Modo = 1* en que se requieren dos instrucciones de lectura.

## **1.5.2 INTERFAZ RS 232/WLAN EZL-300W**

### **1.5.2.1 Introducción**

Con el desarrollo de la tecnología del Internet, la demanda para la comunicación de datos a través del Web tiende a aumentar. Los usuarios deben utilizar el protocolo básico de comunicaciones denominado protocolo TCP/IP. Es decir para conectar un dispositivo al Internet, el usuario debe implementar el protocolo TCP/IP, ya sea en el puerto de salida TCP/IP o usar un Sistema Operativo. Sin embargo, estos métodos resultan incómodos en términos de tiempo, costo, y tecnología.

La serie de ezTCP fabricada por Sollae Systems que trata la conversión Serial a protocolos TCP/IP, proporciona funciones de comunicación TCP/IP solamente a través de conexión del puerto serial, realizando el procesamiento del protocolo TCP/IP en los datos recibidos del puerto serial, y los envía al Internet. Análogamente lo hace de manera inversa donde realizan el procesamiento del protocolo de TCP/IP en los datos recibidos del Internet y los envía al puerto serial.

Ezl-300w soporta la red de infraestructura a través del Punto de Acceso (AP), también proporciona una función de soporte para redes peer to peer "ad hoc" sin necesidad de un Punto de Acceso. Ezl-300w proporciona DHCP así como TCP/IP/UDP, además puede ser aplicable a redes cableadas.



### 1.5.2.2 Funcionamiento

Ezl-300w, es un producto de ezTCP que permite enlazar un puerto serial a una WLAN (Red de Área Local Inalámbrica) a través de un Punto de Acceso (AP).

Ezl-300w provee comunicación TCP/IP a través de IEEE 802.11b (LAN Inalámbrica). Es decir como otros productos de ezTCP, los datos recibidos en el puerto serial del Ezl-300w son procesados y enviados a la red LAN Inalámbrica. Y los datos recibidos de la red LAN Inalámbrica al Ezl-300w son procesados y enviados al puerto serial.

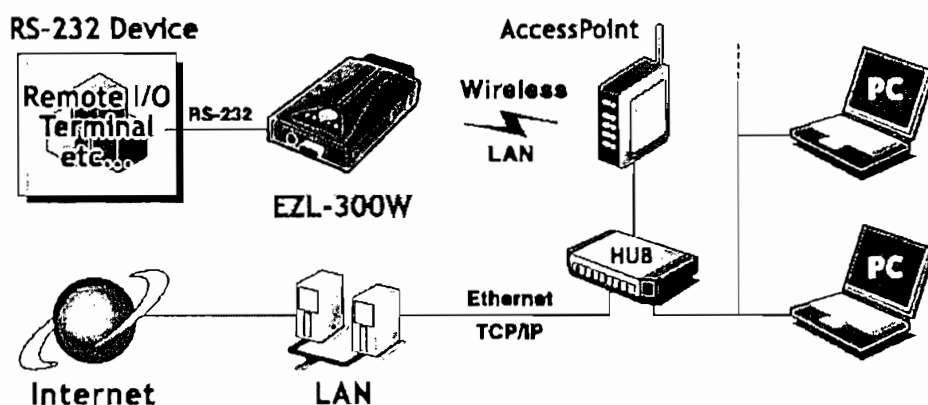


Figura 1.32. Conexión del EZL-300W con una red inalámbrica de infraestructura

### 1.5.2.3 Aplicaciones

- Sistemas de monitoreo remoto.
- Sistemas de meteorología remoto.
- Sistemas contra incendios remoto.
- Control remoto de circulación de aire.
- Sistemas ATM.
- Sistemas de chequeo de salud.
- Sistemas de seguridad.
- Sistemas de muestras de tráfico.

### 1.5.3 PROCESAMIENTO Y ADQUISICIÓN DE LOS DATOS

Los datos obtenidos o adquiridos a través del MAX 180 se tienen que transferir a un microcontrolador de la familia 8051 para su procesamiento, una vez hecho esto se transfieren a la red LAN inalámbrica.

A continuación se detallan varias características y funcionamiento del microcontrolador 8051, así como la configuración de la Red inalámbrica a través del EZL 300W para lograr la comunicación entre cualquier dispositivo de la red y la lectura de los datos de la aplicación.

#### 1.5.3.1 Microcontroladores de la familia 8051

El microcontrolador 8051 está basado en los microprocesadores de 8 bits, contiene internamente un CPU de 8bits, 3 puertos de entrada/salida paralelos, un puerto de control, un puerto serial, dos entradas para Timer/Contador de 16 bits, dos entradas para interrupciones externas, las señales de RD y WR para la toma o almacenamiento de datos externos en RAM, la señal de PSEN para la lectura de instrucciones almacenadas en EPROM externa. Gracias a estas tres señales el microcontrolador 8051 puede direccionar 64 Kbytes de programa y 64 Kbytes de datos separadamente, es decir un total de 128Kbytes.

Además cuenta con 128 bytes de memoria RAM interna. En la figura 1.32 se puede observar la distribución de pines de este dispositivo.

Además este microcontrolador puede generar la frecuencia (Baud Rate) de Transmisión/Recepción de datos por el puerto serial de manera automática partiendo de la frecuencia del oscilador general, por medio de la programación del Timer 1. Dicha frecuencia de transmisión puede ser cambiada en cualquier momento con solo cambiar el valor almacenado en el control o también se puede duplicar o dividir la frecuencia con solo escribir sobre el bit 7 (SMOD) del registro de control (PCON).

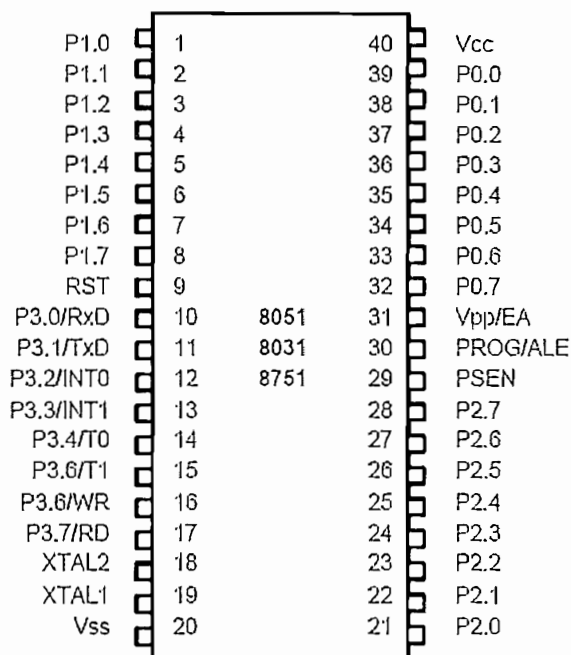


Figura 1.32. Distribución de pines del microcontrolador 8051

La memoria del sistema del 8051 se clasifica en tres partes fundamentales como muestra la figura 1.33: la primera, llamada memoria de programa, en donde se encuentran todas las instrucciones que van a ser ejecutadas por el microcontrolador, Algunas versiones del 8051 cuentan con memoria de programa interna (de 2 a 4 Kbytes).

Cuando se requiere trabajar con una localidad por encima de ésta, la memoria del programa (externa) es seleccionada mediante la activación de la señal PSEN (estado bajo). El máximo espacio de memoria de programa que se puede acceder es de 64 KBytes. Se accede al segundo espacio de memoria denominado memoria de datos mediante la activación de las señales RD y WR, durante la lectura o escritura de datos respectivamente. En este espacio el microcontrolador toma todos los valores que se encuentran en memoria como DATOS, es decir, no puede ejecutar ninguna instrucción que se encuentre aquí almacenada.

El 8051 puede direccionar también 64 KBytes de memoria de datos. El tercer espacio de memoria es denominado como memoria RAM interna, el cual se subdivide en 128 bytes de memoria de la parte baja y en 128 bytes de memoria de la parte alta.

En los primeros 128, se encuentran 4 bancos de 7 registros cada uno. Estos registros son de gran ayuda para la simplificación de los programas, debido a que cada uno de ellos permite almacenar datos momentáneamente y realizar un vasto número de instrucciones del 8051. También dentro de este espacio, se encuentran 16 bytes (del 20H al 2FH) que pueden ser direccionados directamente por bit.

En la parte alta de la memoria RAM interna, se encuentra el contenido de los Registros de Funciones Especiales, formado por Puertos, Registros de Control, Acumuladores, Registros de interrupción, etc.

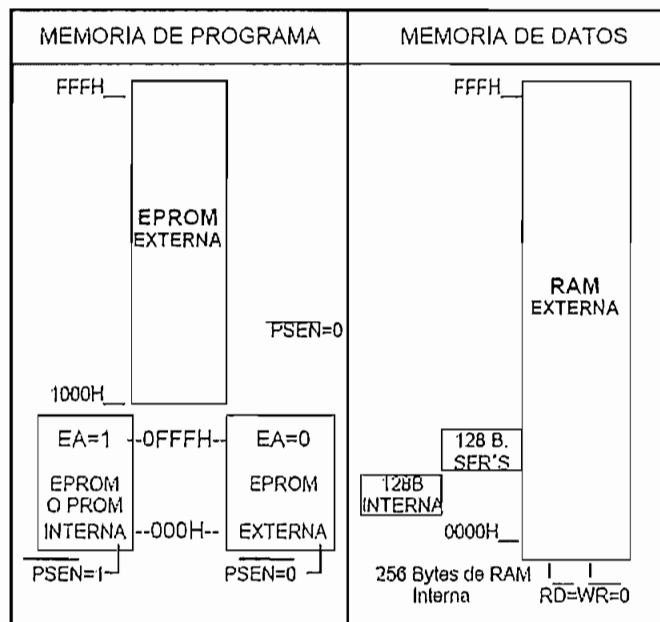


Figura 1.33. Distribución de las localidades de memoria en el microcontrolador 8051

### 1.5.3.2 Configuración de la Red Inalámbrica con el EZL-300W

**Verificación del Ambiente de Comunicaciones.** Antes de instalar EZL-300W, el usuario debe chequear las siguientes condiciones de la red:

- Tipo de red Wireless LAN (infraestructura/ad-hoc), para propósitos de este proyecto, únicamente se utilizará la opción de topología infraestructura.
- El SSID (Service Set Identification ) y el canal Wireless LAN
- Si se está usando WEP y sus parámetros de funcionamiento (el número de bits y los valores clave).
- Protocolo de Autenticación para acceso Wireless LAN .
- Direcciones IP (IP Local , subnet mask, gateway, etc.).
- Elementos del Puerto Serial de los dispositivos conectados (baudrate, databit, parity, y stop bit).
- Protocolo del programa de la aplicación a usar (TCP/UDP, server/client, etc.).

**Configuración de Wireless LAN** Antes de usar el ezTCP, el usuario deberá fijar los elementos relacionados a wireless LAN ya sea vía ezSerialConfig o consola en un modo de Consola.

Para ingresar al modo de consola la tarjeta PCMCIA no debe estar insertada, en este modo el usuario puede no solamente fijar los parámetros de WLAN, también puede fijar los todos los elementos de instalación del ezTCP.

**Conexión a la Red** Para la conexión a la red se inserta la tarjeta de PCMCIA sin proveer energía. Entonces, se conecta un Dispositivo con el ezTCP usando el cable RS232 de acuerdo a las especificaciones, y a la energía de la fuente. Después de que se provea la energía, el LED se debe encender.

Cuando el LED está encendido, significa que un enlace se ha establecido entre el AP y el dispositivo de LAN inalámbrico o entre los dispositivos de LAN inalámbricos para habilitar comunicación entre ellos.

**Configuración de Variables** Una vez que una conexión a la red se ha establecido, se fijan las variables de entorno - variables referentes a la dirección IP, el puerto serial, y al modo de comunicación usando el ezConfig que es un programa de la configuración de la variable de entorno.

**Instalación del Access Point (AP)** Se conecta el AP con la PC a través del cable cruzado LAN. Después de instalar el cable, se energiza al AP, y si el LED de enlace se enciende indica que hay una conexión exitosa entre el Access Point y la PC de la red.

**Instalación del EZL-300W** Para la conexión, se debe enlazar el sistema de adquisición de datos con Ezi-300w usando el cable RS232, se inserta la tarjeta de PCMCIA. Cuando se provee la energía usando el adaptador de energía diseñado para el Ezi-300w, el equipo procura hacer una conexión con el Access Point que contestó primero. Una vez que Ezi-300w esté conectado con el AP, el LED de enlace esta encendido por lo tanto una comunicación con la LAN Inalámbrica es posible.

**Configuración de la Red Inalámbrica** Al configurar la LAN inalámbrica, el usuario debe comprobar el tipo de aplicaciones y la seguridad de la red que es utilizada actualmente o va a ser instalada en el futuro.

**Parámetros a observar en una Red Inalámbrica** La infraestructura es un modo de conexión de red que permite la comunicación entre los dispositivos sin hilos de la LAN o entre la LAN inalámbrica y la LAN cableada (Ethernet) a través del punto de acceso (AP). Cuando un tipo de la red se fija a la infraestructura, la comunicación con la LAN cableada vía el AP es posible, ya que permite una comunicación entre una LAN inalámbrica y una cableada.

La red ad hoc se diseña para comunicarse entre LANs inalámbricas sin ningún AP. Puesto que la comunicación se establece sin ningún AP, el usuario no puede tener acceso a una red externa o al Internet. Esta configuración también se llama modo par-a-par.

## **CAPÍTULO 2**

### **DESCRIPCIÓN Y ANÁLISIS DE LOS INTERFACES PCI Y PCMCIA**

El propósito principal de este capítulo es mostrar las características físicas, eléctricas, funcionales y describir brevemente el procedimiento de transferencia de información que componen dos de los interfaces utilizados típicamente como puertos externos en computadores portátiles para expandir sus capacidades: PCI y PCMCIA.

PCI es el Componente de interconexión Física más conocido como un bus de expansión de datos interno en los computadores de escritorio para interconectar componentes periféricos, pero también muy utilizado en buses externos de computadores portátiles.

PCMCIA es la Asociación Internacional de Tarjetas de Memoria de Computadoras Personales, que es la encargada de la normalización de las tarjetas de memoria o PC Card's, así como del interfaz y procedimiento para acceder a ellas, principios utilizados también en dispositivos como interfaces de red para computadores portátiles dadas sus prestaciones y opciones de bajo consumo.

## 2.1 LOCAL PCI (Peripheral Component Interconnect)

### 2.1.1 INTRODUCCIÓN

El bus local PCI es un bus de alto rendimiento para interconectar chips, tarjetas de expansión y subsistemas de memoria y procesador.<sup>8</sup> En 1990 Intel diseña este bus como un método estándar de interconexión de tarjetas y chips, luego fue adaptado por la industria como un estándar administrado por el PCI Special Interest Group o también llamado PCI SIG, bajo esta definición el bus PCI fue extendido como una definición estándar de buses de expansión para tarjetas adicionales.

PCI está enfocado a interconectarse con el hardware de las desktop, pero posee soporte para ser incluido también en computadoras notebook y laptop. Actualmente existe en el mercado varios diseños de laptop, que incluyen bus local de video, disco de E/S e interfaces de red.

Existen dos tipos de dispositivos PCI: Master (Maestro) y Slave (Esclavo). El dispositivo master es más inteligente, puede conducir el procesamiento independiente del bus o de otros dispositivos.

El master comparte el bus con el procesador y con los dispositivos slave, pudiendo hacer las veces de slave en el caso de que exista otro master conectado en el bus. El dispositivo slave acepta comandos y responde a peticiones del master.

Las tarjetas de expansión PCI trabajan eficientemente en todos los sistemas y pueden ser intercambiadas de cualquier manera. Solamente los controladores de dispositivo deben naturalmente ser ajustados al sistema anfitrión (host) es decir a su correspondiente CPU.<sup>9</sup>

---

<sup>8</sup> <http://www.monografias.com/trabajos11/basdat/basdat.shtml>

<sup>9</sup> <http://cpvs.iespana.es/hardware/7.pdf>



## 2.1.2 DESCRIPCIÓN GENERAL

El bus local PCI es un bus de alto desempeño que requiere un mínimo de 47 pines para un dispositivo slave y 49 pines para un master; incluyendo manejo de datos, líneas de dirección, control de interfaz, arbitraje y función del sistema. Se especifica hasta 120 conexiones para una tarjeta PCI de 32 bits y 184 para una tarjeta PCI de 64 bit estándar, basándose en un diseño multiplexado de líneas de datos y direcciones, permitiendo utilizarla como mecanismo de interconexión entre controladores de periféricos altamente integrados, placas periféricas de expansión y sistemas procesador/memoria.

El bus PCI soporta los requerimientos tanto para sistemas móviles (Laptops) como para servidores, estableciendo voltajes de 3,3 V para entorno móvil y para sistemas de escritorio de 5 V.

Los componentes e interfaces agregables PCI son independientes del tipo de microprocesador, permitiendo una eficiente transición a futuras generaciones de procesadores y al uso de arquitecturas multi-procesador.<sup>10</sup> Esta independencia permite optimizar el bus local PCI para funciones de E/S, habilitando la operación recurrente del bus local con el sub-sistema procesador/memoria y maneja múltiples periféricos de altas prestaciones (video, LAN, SCSI, FDDI, HDD, etc) y otras E/S de gran ancho de banda continuarán incrementando los requerimientos de ancho de banda para el bus local.

En función de esto, una extensión transparente a 64 bits de los buses de datos y direcciones de 32 es precisa, duplicando el ancho de banda del bus y ofreciendo compatibilidad hacia adelante y atrás con periféricos PCI de 32 y 64 bits.

Para proveer una rápida y fácil transición de 5 V a 3,3 V, el estándar PCI define tres tipos de conectores, uno para cada voltaje y otro "universal" que soporta los dos.

---

<sup>10</sup> <http://betoe.imgoingtobed.com/informatica/PCII.doc>.

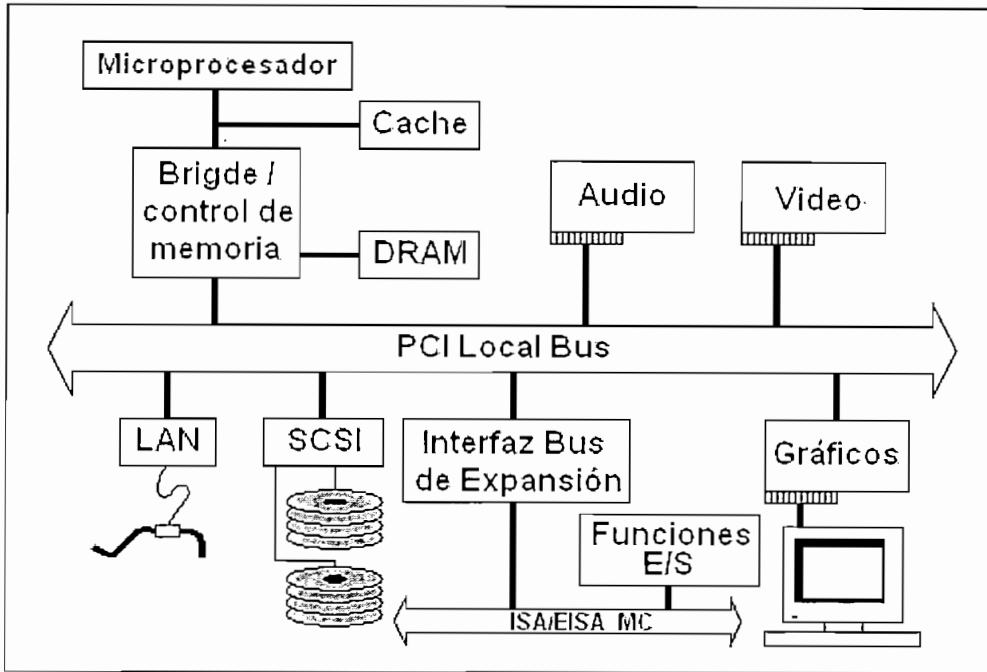


Figura 2.1. Diagrama de un bus PCI típico con sus conexiones.

En la figura 2.1 se muestra la arquitectura de un bus local PCI típico. Se observa que el sistema procesador-caché-memoria se conecta al bus PCI mediante un "Puente PCI". Este puente provee un camino de baja latencia y gran ancho de banda, para que el procesador pueda acceder a cualquier dispositivo incluido en memoria o en los espacios para E/S, permitiendo que cualquier dispositivo PCI Master tenga acceso directo a memoria.

Aplicaciones usuales del bus local PCI soportan hasta cuatro conectores para placas agregables con conectores tipo Micro Channel.

### 2.1.3 CARACTERÍSTICAS FÍSICAS

Actualmente, existen tres tamaños de tarjetas PCI: corta, larga y variablemente corta (no es necesario que los sistemas soporten los tres tipos de tarjetas). Las tarjetas largas incluyen una extensión ISA/EISA para permitir que sean utilizadas en este tipo de sistemas.

El bus PCI emplea un conector estilo Micro Channel de 124 pines para 32 bits (188 en caso de una implementación de 64 bits) pero únicamente 47 de estas conexiones se emplean en una tarjeta de expansión (49 en caso de que se trate de un adaptador bus-master); la diferencia se debe a la incorporación de una línea de alimentación y otra de tierra. Cada una de las señales activas del bus PCI está bien junto o frente a una señal de alimentación o de tierra, una técnica que minimiza la radiación.<sup>11</sup> Las diferencias físicas entre estas tarjetas se muestran en la figura 2.2.

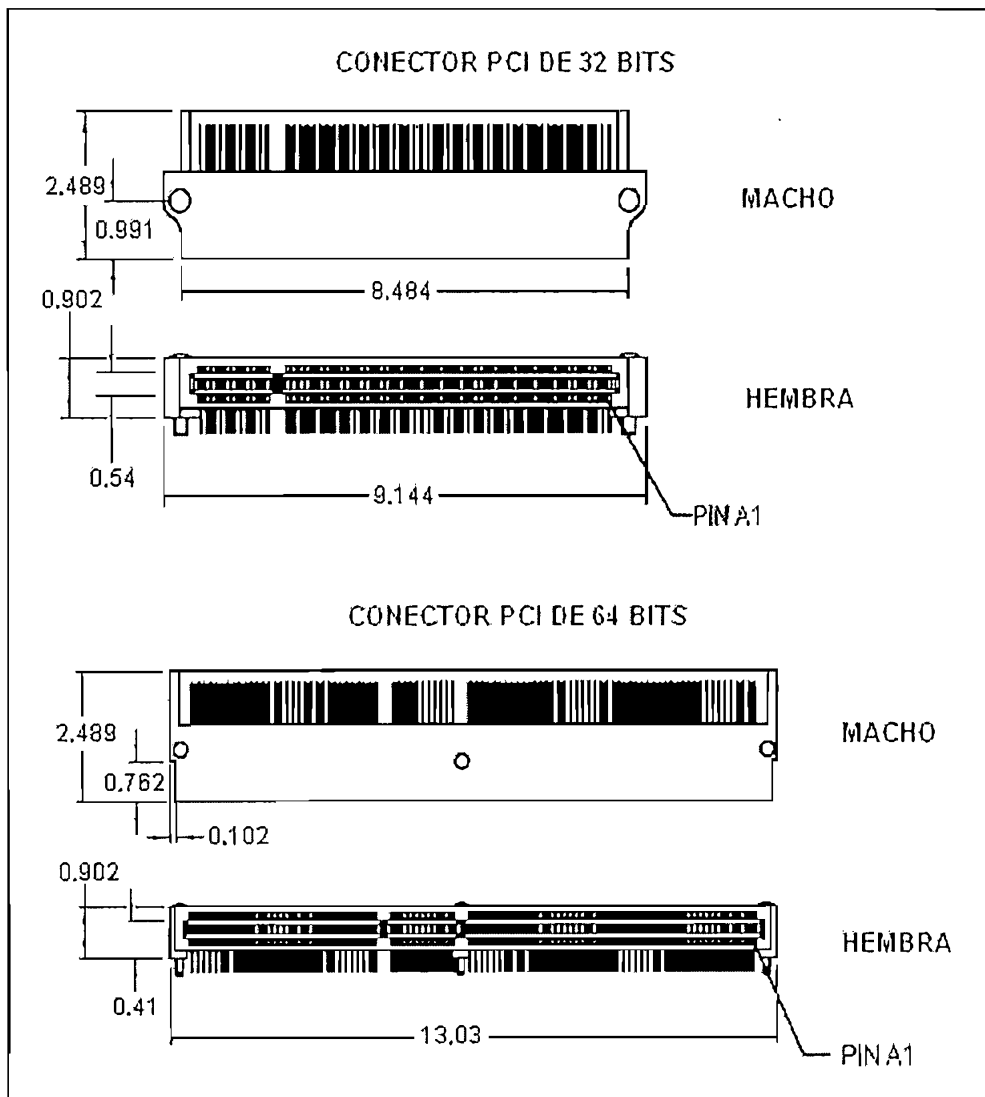


Figura 2.2. Diagramas de un conector PCI de 32 bits y de un conector PCI de 64 bits.<sup>12</sup>

<sup>11</sup> <http://www.mailxmail.com/curso/informatica/arquitecturaordenadores/capitulo12.htm>

<sup>12</sup> [www.meritec.com/Pages/pdf/pci.pdf](http://www.meritec.com/Pages/pdf/pci.pdf)

El límite práctico en la cantidad de conectores para buses PCI es de tres; ya que más conectores aumentarían la capacitancia del bus y las operaciones a máxima velocidad resultarían menos confiables. PCI soporta un riguroso proceso de autoconfiguración.

Cada dispositivo PCI incluye un conjunto de registros de configuración que permiten la identificación de cada tipo de dispositivo (SCSI, video , Ethernet , etc) y la compañía que los ha producido.

Otros registros permiten la configuración del dispositivo como los aspectos de las direcciones de Entrada/ Salida, direcciones de memoria, niveles de interrupción, etc.

#### 2.1.4 CAPACIDADES DEL BUS LOCAL PCI

Las especificaciones del bus local PCI proveen una serie de cualidades que permiten alcanzar altas prestaciones. Las capacidades se resumen en:

- Un bus PCI de 32 bits trabajando a 33 MHz presenta una velocidad de transferencia de datos de 132 MB/s y un bus PCI de 64 bits una velocidad de transferencia de datos de 264 MB/s; con una extensión del bus para funcionar a 66 MHz duplica su velocidad de transferencia de datos.
- Bus sincrónico con operación a 33 o 66 MHz.
- Accesos aleatorios de baja latencia (60 nseg para 33 MHz y 30 nseg para 66 MHz de latencia de acceso para escritura desde un master que ocupa el bus y un registro esclavo).<sup>13</sup>
- Optimizado para interconexión directa (no necesita lógica de conexión).
- La arquitectura multiplexada reduce el número de pines (47 para esclavos y 49 para maestros), las placas de expansión simples PCI trabajan en sistemas ISA, EISA y MC. (Con cambios mínimos a los diseños de chasis existentes).

---

<sup>13</sup> <http://betoe.imgoingtobed.com/informatica/PCI1.doc>

- Soporta direccionamiento de 64 bits y tiene conectores de expansión tipo MC.
- Capacidad total multi-maestro, permitiendo que cualquier maestro PCI pueda acceder puerto a puerto con cualquier esclavo PCI.
- Un slot compartido se acopla tanto a tarjetas estándar ISA, EISA o MC como a tarjetas de expansión PCI.
- Provee paridad tanto en datos como en direcciones, y permite la implementación de plataformas robustas.

La figura 2.3 muestra un diagrama funcional, para una combinación maestro/esclavo, las líneas pueden ser de distinto tipo, a saber, las de entrada son estándar, las de salida son tipo totem pole, tri-state, sustained tri-state u open drain.

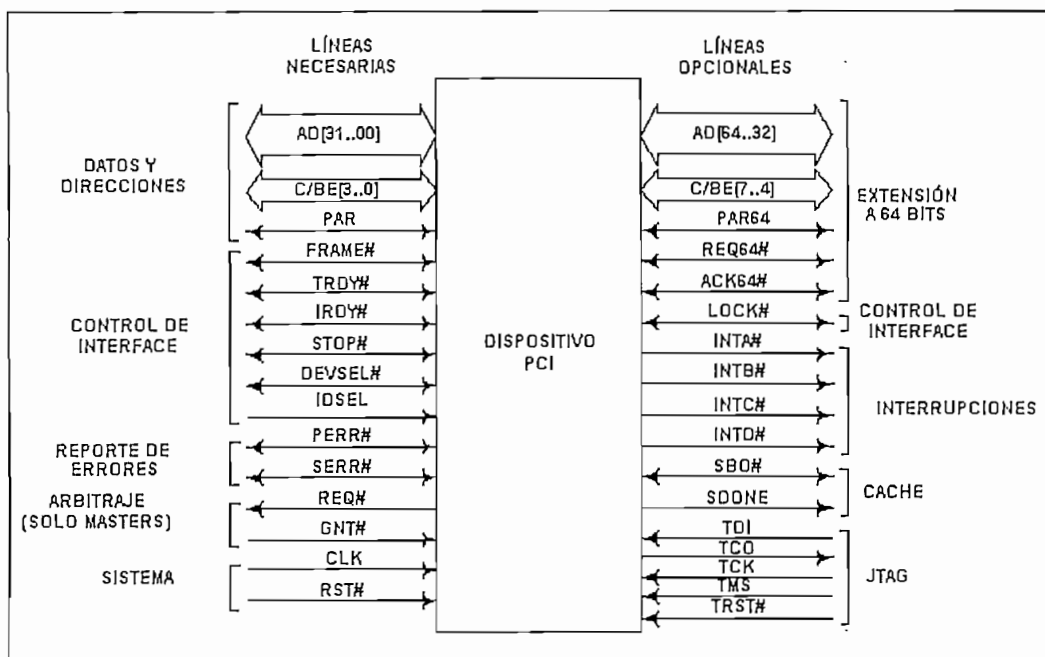


Figura 2.3. Diagrama Funcional de las conexiones de pines PCI

## 2.2 PCMCIA (Personal Computer Memory Card International Association)

### 2.2.1 INTRODUCCIÓN

El crecimiento de la industria de los microcomputadores en los años 80 y la popularidad de los PC portátiles y los notebooks, provocó la necesidad de diseños con pequeñas dimensiones, ligeros y con bajo consumo de energía, pero el principal problema era el de sustituir a los discos de almacenamiento de información, incluyendo la característica de que fueran transferibles a otros sistemas y rápidamente accesibles para la lectura y escritura de ficheros. La opción en un principio fue la de un subsistema de floppy disk con una pequeña batería incorporada, pero la aparición de Flash memory, que no requerían batería hizo que las metas se trazaran en esta dirección. Numerosas empresas produjeron tarjetas de memoria con diferentes características físicas y eléctricas, siendo la compatibilidad un obstáculo más en las necesidades de la industria. Un estándar para el encapsulado físico, una interfaz eléctrica y un conector fue necesario para la compatibilidad de las tarjetas de memoria.

Varias empresas se reunieron en el verano de 1988 para investigar la posibilidad de formar una organización estándar y normar las tarjetas de memoria y su interoperabilidad. Un año más tarde la Asociación Internacional de Tarjetas de Memoria de Computadoras Personales (PCMCIA) fue fundada, y la primera PCMCIA estándar (Versión 1.0) fue introducida en Septiembre de 1990. Este estándar especificaba el diseño de las tarjetas de memoria (comúnmente llamadas PC Cards) y su conector para ser implementado como disco virtual.

PCMCIA fue formado para promover la estandarización e interoperabilidad de tarjetas de PC. Inicialmente, su primer objetivo fue definir un sistema de tarjetas PC estándar compatible con IBM PC (basado en DOS) y en mayor medida permitir una variedad de tipos de computadoras y no de productos de estas para

así poder intercambiar libremente tarjetas de PC. Con estas metas en mente la PCMCIA definió el estándar para PC Cards.

La Asociación de Desarrollo de Industrias Electrónicas Japonesas (JEIDA) empezó a trabajar en tarjetas de memoria y su estandarización en 1985. En 1989 PCMCIA adoptó el conector de 68 pines de la JEIDA como conector de interfase. Para poder llegar a la meta marcada de compatibilidades e interoperatividad JEIDA y PCMCIA empezaron a trabajar conjuntamente para así asegurar la compatibilidad entre ambos estándares. En 1990 PCMCIA dio a conocer su primer estándar (versión 1.0) y JEIDA lanzó su cuarto estándar (versión 4.0). En las siguientes versiones JEIDA y PCMCIA siguieron trabajando muy estrechamente para así soportar otros estándares.

El entorno de computadoras móviles también necesita la estandarización de pequeños diseños de entrada-salida que pudieran ser añadidos a estos sistemas como dispositivos de expansión. La versión 2.0 de PCMCIA incluyó el soporte de dispositivos E/S que pudieran ser insertados dentro de un conector PCMCIA. Igual que en las tarjetas de memoria estos dispositivos son diseñados para ser detectados automáticamente por el sistema cuando son insertados, configurándose automáticamente, pudiendo ser insertadas en un conector PCMCIA después de que el sistema haya sido encendido y esté operando.

Las tarjetas PCMCIA recientemente se enfocan al desarrollo de estándares para periféricos modulares, debido a que proporcionan interoperabilidad no solo en computadores móviles, si no en diversos productos como cámaras digitales, palms, televisión por cable y automóviles. PCMCIA ha añadido las especificaciones del factor de forma a las tarjetas de PC pequeñas y ahora publica y mantiene el estándar de las tarjeta miniatura. También publicará el estándar de la tarjeta SmartMedia, el cual proporciona soluciones de memoria en uno de los factores de forma de periféricos modulares más pequeños en la actualidad.

## 2.2.2 CARACTERÍSTICAS FÍSICAS

El estándar PCMCIA define las dimensiones físicas, forma y clase de conectores para los 6 tipos de tarjetas especificadas: 3 ejemplares de tarjetas PC Card de tamaño completo: Tipo I, Tipo II y Tipo III, y 3 tipos de tarjetas PC Card pequeñas, que son también: Tipo I, Tipo II, y Tipo III.<sup>14</sup>

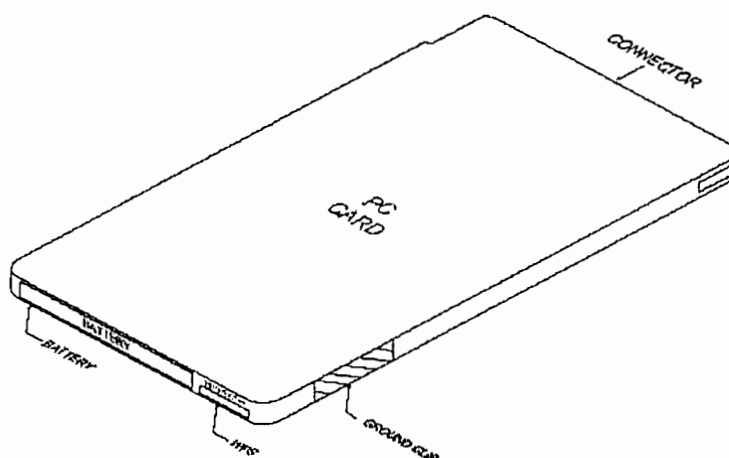


Figura 2.4. Tarjeta PC Card Tipo I

La figura 2.4 muestra una tarjeta PC Card Tipo I, de 3 mm de espesor, lo suficiente para la expansión de memoria y nada más, por tanto no incluye la circuitería de entrada y salida ni conectores requeridos.

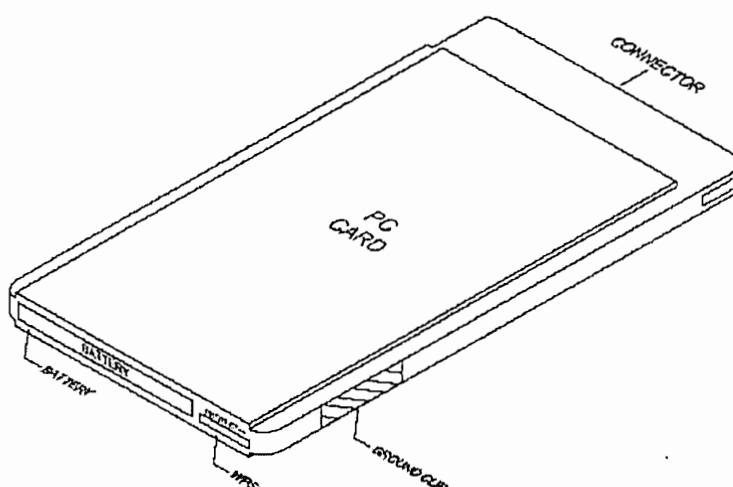


Figura 2.5. Tarjeta PC Card Tipo II

<sup>14</sup> 1999 PCMCIA/JEIDA STANDARD



La tarjeta PC Card Tipo II fue lanzada al mercado un año más tarde que su predecesora, ofreciendo más funcionalidades y resolviendo las necesidades de soporte de entrada y salida así como de memoria, pero manteniendo compatibilidad con aquella, a pesar de que su espesor aumentó a 5 mm. La tarjeta PC Card Tipo III mostrada en la figura 2.6, posee un espesor de 10.5 mm, proporcionando el suficiente espacio como para incluir un disco duro removible dentro de la tarjeta.

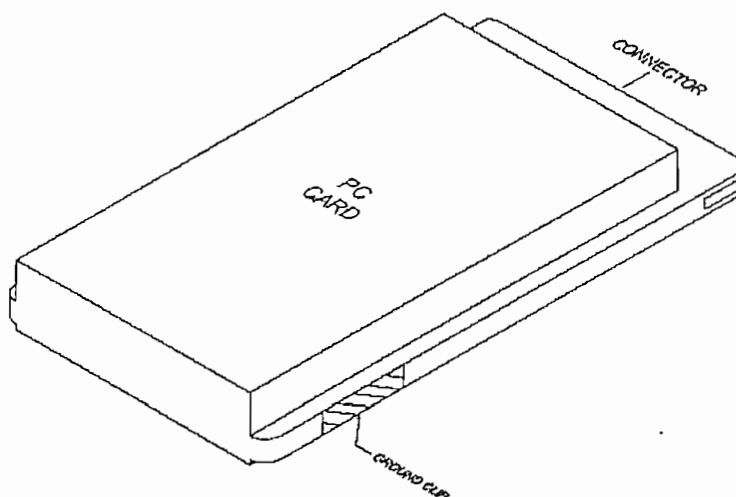


Figura 2.6. Tarjeta PC Card Tipo III

### 2.2.2.1 Zócalos y Conectores

Los dispositivos PCMCIA utilizan una interfaz de software llamada socket services para enlazar a la tarjeta de expansión con dispositivos de arquitectura Intel. Este software puede estar integrado en el ROM-BIOS de la computadora o añadirse a los dispositivos utilizados, mediante un manejador de software. Los zócalos y conectores de las tarjetas PC Card están diseñados de tal forma que no pueden ser conectados de forma incorrecta, como se muestra en la figura 2.7.

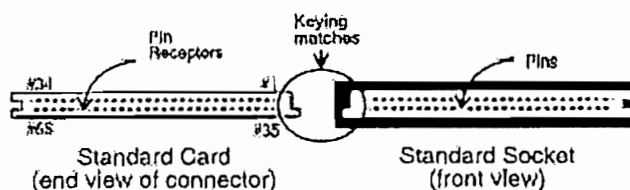


Figura 2.7. Zócalo y conector PC Card.

Las tarjetas PC Card Tipo II pueden soportar voltajes duales, normalmente para funcionamiento con 5 v, pero en operación de baja potencia utiliza 3.3 v, también existiendo versiones diseñadas solo para funcionar con voltaje de 3.3 v, sin necesidad de soportar los dos valores de voltaje, utilizándose la forma de los conectores para evitar insertar una tarjeta de operación en bajo consumo de potencia en una que funciones con 5 v o voltaje dual y viceversa.

Las tarjetas PCMCIA poseen 3 longitudes de pines diferentes según su función:

- Pines de alimentación ( GND y VCC ), 2.5 mm.
- Pines generales de interfaces ( dirección, datos y control ), 2.1 mm
- Pines de detección de tarjeta, 1.5 mm

Cuando una tarjeta es insertada en el socket PCMCIA, los pines de alimentación hacen el primer contacto, seguidamente por los pines generales de interfaces y luego los pines de detección de tarjeta.

#### 2.2.2.2 Características Ambientales<sup>15</sup>

Para conectores estándares normales:

- Rango de temperatura de operación: -20°C a 60°C.
- Rango de temperatura de almacenamiento: -40°C a 70°C.
- Humedad relativa: 95%.
- Número de inserciones y extracciones: 10.000 para ambientes de oficina y 5.000 para ambientes contaminados.
- Específicamente el estándar PCMCIA define:
- Rango de temperatura de operación: 0°C a 55°C.
- Rango de temperatura de almacenamiento: -20°C a 65°C.

---

<sup>15</sup> 1999 PCMCIA/JEIDA STANDARD

### 2.2.3 CARACTERÍSTICAS ELÉCTRICAS

Las especificaciones eléctricas de este estándar describen los pines de salida del conector, los protocolos de interfaz, señalización, el modelo de sincronizaciones del interfaz, programación, aspectos de inserción y extracción de la tarjeta, energización, y configuración.<sup>16</sup>

El estándar original PCMCIA define un zócalo de interfaz solo para tarjetas de memoria, pero versiones posteriores agregaron nuevos interfaces como:

- Solo Memoria.
- Memoria o Entrada/Salida
- ATA (accesorios AT para drives IDE)
- AIMS (auto indexación para almacenamiento en masa)
- DMA con PC Card.

Cada uno de estos interfaces emplea un conector de 68 pines, sin embargo la asignación de los pines cambia con cada tipo de interfaz. A continuación se detallan los dos primeros interfaces mencionados por ser los más relevantes y en los que se basan los siguientes.

#### 2.2.3.1 El interfaz de Memoria

Todas las tarjetas PC Card de 16 bits deben inicialmente operar como dispositivos de solo memoria. Existen dos tipos de espacios para direcciones de memoria dentro de una PC Card:

- Memoria Común: Es el espacio de memoria de trabajo típicamente para almacenar datos y ejecutar archivos.
- Memoria de atributo: Es usada para configurar información y contiene el CIS (Card Information Structure) y configuración de registros.

---

<sup>16</sup> 1999 PCMCIA/JEIDA STANDARD

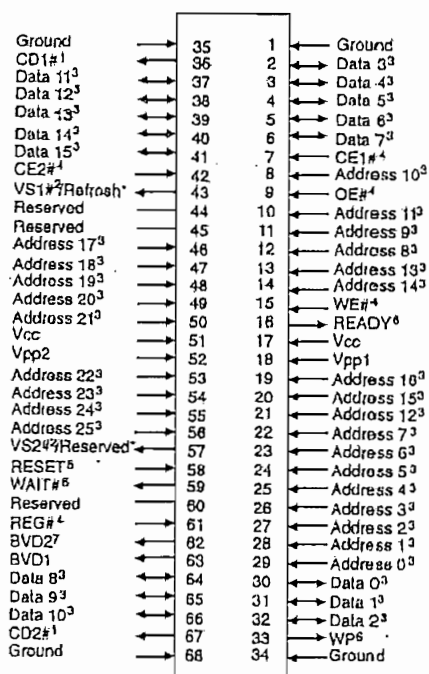


Figura 2.8. Zócalo de interfaz para memoria PCMCIA.

### 2.2.3.2 Interoperabilidad

El tipo de conexión de PCMCIA puede ser diseñado dentro de una gran variedad de arquitectura de buses, permitiendo la interoperabilidad entre PC Cards. Esta conexión entre el bus del sistema y una conexión PCMCIA es facilitada a través de un adaptador al bus del sistema principal (HBA).

El adaptador actúa como un puente, ubicando transacciones del bus del sistema principal en la tarjeta PC instalada en el conector PCMCIA. El adaptador debe estar programado por el sistema para poder ingresar a la tarjeta PC. La manera en la que el adaptador del bus debe ser programado es determinada por los requerimientos de la PC Card instalada en el conector de control.

### 2.2.3.3 Configurando la Tarjeta

Las PC Card requieren un software de configuración que detecte su presencia, determine sus requerimientos de configuración y las programe para operar dentro del sistema. Adicionalmente, los programadores deben determinar el recurso de sistema requerido de su PC Card y asignarlo a la tarjeta. Los

recursos que estos asignan para sus PC Card no deben todavía asignarse a otros dispositivos en el sistema.

En el pasado, los programadores tuvieron que de algún modo determinar qué recursos de sistema estarían disponibles para ser asignados, o probar esperando que los recursos estuvieran alrededor de lo asignado. Estos aspectos y otros colocaron una carga pesada sobre el programador para asegurar que sus tarjetas trabajan en un determinado sistema.

Afortunadamente, PCMCIA introdujo otra capa de software llamada servicio de tarjeta que libera las sobrecargas en el programador. Los servicios de tarjetas posibilita una capa de software consistente de funciones de alto nivel que el programador puede llamar para acceder a la tarjeta, determina sus requerimientos de configuración, e instalar los recursos del sistema requeridos.

Una de las funciones principales del servicio de tarjeta es la asignación de recursos en el sistema. Los servicios de tarjetas mantienen una base de datos de los recursos del sistema disponibles para asignarlas a las PC Card. Primero la capacidad de los requerimientos de configuración de las tarjetas para la lectura del CIS, esto demanda que esos recursos sean asignados a las tarjetas. Si los recursos están disponibles, el servicio de tarjeta devuelve "éxito" para la habilitación y el recurso es entonces asignado.

Si el recurso ya ha sido usado, entonces el servicio de tarjeta devuelve "fallo" y otra opción de configuración debe ser leída del CIS e intentada de nuevo. Este proceso continúa hasta que una de las opciones de configuración de la tarjeta satisface la instancia o hasta que todas las configuraciones son probadas y entonces la tarjeta no puede ser configurada.

El servicio de tarjeta también notifica la posibilidad de inserción, extracción de tarjeta y otros cambios de estado. Esto elimina la necesidad de hacer un sondeo para ver si alguno de los estados ha cambiado.

#### 2.2.3.4 El Metaformato

PCMCIA ha definido un software estructurado llamado metaformato que define el soporte que puede ser suministrado con la tarjeta. El metaformato está en un modelo software de cuatro capas que abarca el CIS.

La única capa requerida por todas las tarjetas es el CIS (primera capa), la cual contiene información necesaria para configurar la tarjeta dentro del sistema. Las otras capas están pensadas para tarjetas de memoria que son usadas como discos virtuales. Las tarjetas de memoria son accedidas como un disco lógico vía el sistema principal de ficheros del sistema operativo y en unión con dispositivos drivers que tienen los conocimientos de como los arreglos de memoria de las PC Card son organizados. Las capas adicionales suministran información que puede ser usada por los ficheros principales, como son utilidades y software que requieren conocimientos de las características de los arreglos de las tarjetas de memoria.<sup>17</sup> Pocas PC Card hoy en día contienen información para las capas de metaformato.

A continuación se describe cada una de las capas del metaformato:

- **Capa 1:** Capa de compatibilidad, esta capa incluye la información necesaria para que una tarjeta sea reconocida y configurada. Esta parte del estándar de la tarjeta es comúnmente referida como Estructura de Información de la Tarjeta (CIS).
- **Capa 2:** Capa de formato de grabación de datos, cuya información especifica cuantos datos son grabados en los arreglos de memoria de la PC Card, y especifica que capacidad de detección de errores posee.
- **Capa 3:** Capa de organización de datos, define la organización lógica de datos dentro de una partición en una tarjeta de memoria.
- **Capa 4:** Capa específica de sistema, posee información que ciertas aplicaciones requieren para un determinado entorno operativo.

---

<sup>17</sup> [http://diea.ulpgc.es/users/aurelio/int\\_equipos/trab9899/pcmcia\\_2/](http://diea.ulpgc.es/users/aurelio/int_equipos/trab9899/pcmcia_2/)

## 2.2.4 ADAPTADOR DE BUS DEL HOST PC CARD

El adaptador del bus del host PCMCIA (HBA) reside físicamente entre el bus del host PCMCIA (usualmente un bus de expansión como un ISA, EISA, PCI) y los zócalos PCMCIA,<sup>18</sup> tal como se ilustra en la figura 2.9 en la que además se puede apreciar el software que se dirige desde y hacia el HBA:

- Configuración y notificación de eventos (específicamente PCMCIA).
- Run-time (flujo de programas y ejecuciones normales).

El software PCMCIA ingresa inicialmente al HBA y lo programa para reconocer los accesos hechos a la PC Card. Una vez programado, el HBA reconoce las direcciones dentro de la PC Card y traslada la transacción del bus hacia el zócalo de la tarjeta.

El HBA además monitoriza los eventos de cambio de estado y genera una interrupción para informar a los servicios de la tarjeta estos eventos.

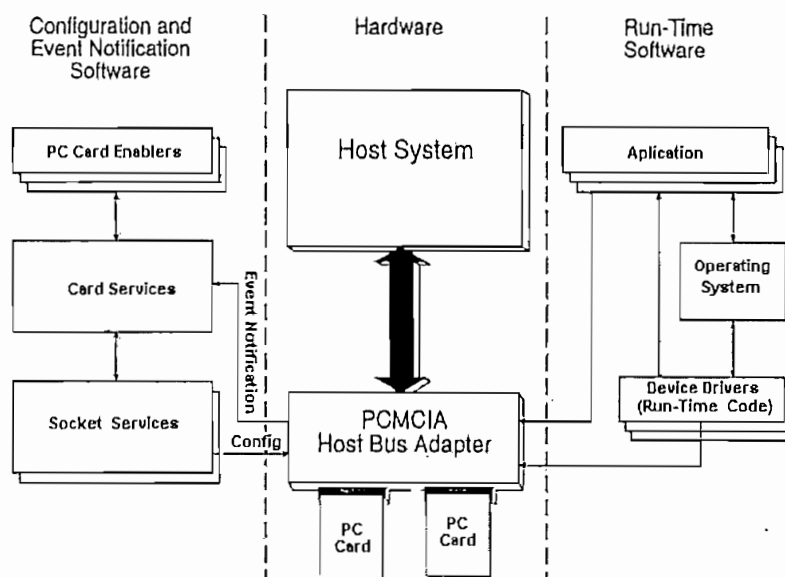


Figura 2.9. Ambiente PCMCIA

<sup>18</sup> 1999 PCMCIA/JEIDA STANDARD

### 2.2.5 ESTRUCTURA DE LA INFORMACIÓN DE LA TARJETA (CIS)

Las PC Card incluyen una estructura de datos llamada *Card Information Structure* (CIS) que esta almacenada en una memoria no volátil.

El CIS proporciona al software un método para determinar qué clase de PC Card está instalada, junto con su velocidad, tamaño, y los recursos de sistema requeridos por la tarjeta.

Después de haber determinado esta información, el HBA del PCMCIA se puede programar para permitir el acceso a la PC Card, y la propia tarjeta puede ser configurada, escribiendo en sus registros de configuración.

Los registros de configuración son necesarios para las tarjetas E/S, pero son opcionales para las tarjetas de memoria. Los registros de configuración y los registros de la CIS están establecidos en el espacio de la memoria atribuida.

Como se muestra en la figura 2.9 el driver cliente de la tarjeta PC lee la CIS mientras se inicia la tarjeta para determinar las opciones de configuración que soporta la tarjeta. El cliente de la PC Card accede a la CIS a través de los servicios de la tarjeta y los servicios de los conectores.

Después de que la CIS haya leído las necesidades de recursos y el tipo de tarjeta, entonces el driver cliente programa el HBA y configura la tarjeta PC de nuevo a través de los servicios de conectores y de la tarjeta.

Normalmente no se puede hacer otro acceso a la CIS, después de que la tarjeta se haya inicializado. La memoria o el dispositivo E/S pueden ser accedidos a través del bus host de expansión. Hay que tener en cuenta que la CIS solo puede ser accedida por programas que conozcan PCMCIA.



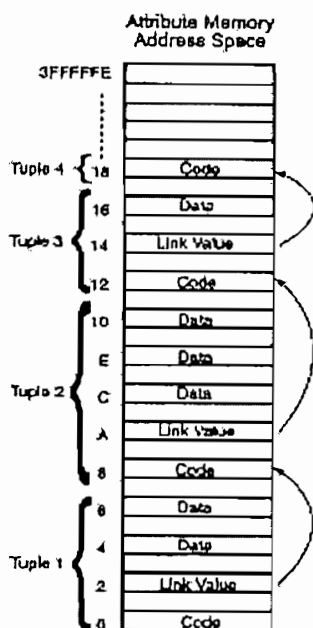


Figura 2.10. Esquema del CIS que consiste en una lista encadenada de cuatro tuples.

La CIS está mapeada dentro del espacio de dirección de la memoria atribuida y comienza en la dirección cero, como se muestra en la figura 2.10. La CIS es una lista enlazada de datos en bloque, o "tuples", que describen la función y las características de la tarjeta PC. La configuración del software accede a estos datos para determinar las características y las necesidades de configuración de una tarjeta PC dada.

Los tuples son identificados por un único código que se encuentra en el primer byte de cada tuple. Como los datos CIS solo se encuentran en las localizaciones impares dentro del espacio de direcciones de la memoria atribuida, esto significa que la información solo retorna por la ruta de datos inferior (D7:D0). Esto simplifica el diseño de las tarjetas para acomodar sistemas host de 8-bits que solo conecta con la ruta de datos inferior.

### 2.2.5.1 Tuples o Bloques de Datos

Un tuple es un set de accesos de datos que caracteriza un cierto aspecto de una PC Card. El estándar PCMCIA especifica los tuples que son utilizados por los diseñadores de PC Card para proporcionar la información sobre su dispositivo.

Los tuples proporcionan la información de velocidad y tamaño del dispositivo. La información del tuple es utilizada a menudo por la configuración software para determinar los requisitos de la configuración de la tarjeta. Sin embargo, otros tuples proporcionan la información que se puede utilizar por programas utilitarios, y para comprobar las capacidades adicionales de la tarjeta.

- **Formato del Tuples.**

Todos los tuples tienen un formato general definido por PCMCIA.

Byte	Standard Tuple Format	
0	TPL.CODE	Tuple type code (XXh).
1	TPL.LINK	Link to next tuple (number of bytes remaining in tuple).
n	TPL.DATA	Tuple specific data block (definition, format and length defined by individual tuples).

Tabla 2.1. Formato Básico de un Tuple.

El CIS consiste en una lista enlazada de tuples. Cada uno especifica un valor de acoplamiento que identifica el comienzo del tuple siguiente. El sistema exacto de tuples incorporados en el CIS depende sobre todo del tipo de tarjeta y de sus capacidades.

#### 2.2.5.2 Configuración de la Tabla

Los dispositivos de E/S necesitan que el CIS contenga una tabla de configuración que no requiera de una tarjeta de memoria.

Esta tabla consiste de múltiples entradas, cada una describe un sistema de opciones de configuración que la PC Card necesita para la operación normal. Cada entrada de la tabla de la configuración refleja las combinaciones posibles de recurso que la PC Card puede configurar para:

- Configuración de entrada de Tuple.
- Interpretación de la Tabla de Configuración.
- Múltiples Funciones de la PC Card.

La memoria ROM y FLASH se utilizan comúnmente para poner al CIS en ejecución. La ventaja de utilizar una memoria FLASH es que el CIS puede ser actualizado fácilmente.

#### 2.2.5.4 Los Registros de Configuración

Cada función de E/S de las PC Card debe implementar registros de configuración. El estándar de la tarjeta PC define los siguientes registros de configuración:

- Registro de configuración de opciones de todas las funciones E/S

### 2.2.5.3 Dispositivos Comúnmente usados por el CIS

La memoria ROM y FLASH se utilizan comúnmente para poner al CIS en ejecución. La ventaja de utilizar una memoria FLASH es que el CIS puede ser actualizado fácilmente.

### 2.2.5.4 Los Registros de Configuración

Cada función de E/S de las PC Card debe implementar registros de configuración. El estándar de la tarjeta PC define los siguientes registros de configuración:

- Registro de configuración de opciones de todas las funciones E/S (obligatorio).
- Registro de configuración y estado (opcional).
- Registro de reemplazo de pin (opcional).
- Registro de conector y copia (opcional).
- Registro de estados extendido (opcional).
- Registro de direcciones E/S base, obligatorio para PC Card multifunción.
- Registro límite E/S (opcional).

El diseño debe constar básicamente de:

- Conversor analógico digital.
- Microcontrolador.
- Conversor de interfaz TTL a RS232 y viceversa.
- Transceiver.

La figura 3.1 muestra el diagrama funcional del diseño a realizar.

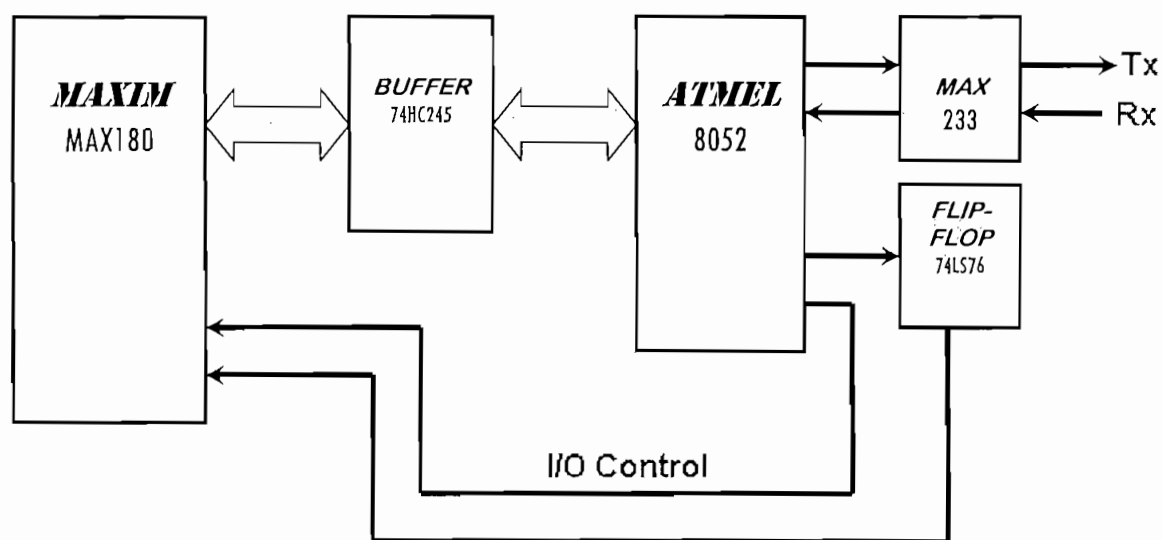


Figura 3.1. Diagrama funcional del Diseño

## 3.2 CONFIGURACIÓN DEL HARDWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS

### 3.2.1 MICROCONTROLADOR

Un Microcontrolador es un circuito integrado (Un circuito integrado es un pastilla o chip en la que se encuentran los componentes electrónicos, que permiten realizar alguna función, éstos poseen transistores en su mayoría, también se encuentran en éste resistencias, diodos, etc.) el cual, de no ser programado no realizará tarea alguna.

Contiene tres unidades básicas que lo identifican como tal y son CPU (Unidad central de proceso) para procesar la información, Memoria (de datos para guardar información y de Programa para almacenar las instrucciones) y las líneas de entrada y salida.

### **3.2.1.1 CPU**

La Unidad Central de Proceso es el corazón del Microcontrolador y es aquí donde todas las operaciones aritméticas y lógicas son realizadas y se controla el funcionamiento de los distintos componentes. Es decir, es la unidad que ejecuta todas las operaciones que están almacenadas en la memoria de programa.

### **3.2.1.2 Memoria de Programa**

Contiene las instrucciones organizadas en una secuencia particular para realizar una tarea. Típicamente es denominada memoria de sólo lectura (ROM) o también OTP, EPROM o FLASH que son memorias que una vez programadas retienen la información aunque el sistema no tenga energía.

### **3.2.1.3 Memoria de Datos**

Esta es una memoria que puede ser escrita o leída según sea requerido por el programa. Tiene las funciones de almacenamiento de datos y almacenamiento de variables. Este tipo de memoria es usualmente llamada memoria RAM Cada localidad de memoria tiene una dirección única con la cual la CPU encuentra la información necesaria.

Los microcontroladores actuales contienen ambas memorias (Datos y Programa) incluidas dentro del circuito integrado.

Por otro lado, resulta necesario contar con otras unidades que hacen posible el funcionamiento mínimo de un Microcontrolador que son Circuitería de Temporización y Entradas/Salidas.

#### 3.2.1.4 Circuitería de Temporización

Los Microcontroladores usan señales de temporización llamadas señales de Reloj que proveen una referencia de tiempo para la ejecución de un programa. Esta señal determina en qué momento los datos deben ser escritos o leídos de la memoria. Así mismo, provee el sincronismo con los dispositivos conectados al Microcontrolador (Periféricos).

#### 3.2.1.5 Entradas/Salidas

Los Microcontroladores requieren una interfaz para comunicarse con la circuitería externa. Esta Interfaz es denominada comúnmente Puerto (Por ejemplo puerto serial, paralelo, etc.). Existen Puertos de Entrada y Salida los cuales permiten que las señales (o datos) sean leídos del exterior o enviados al exterior del Microcontrolador. Los Puertos están formados por pines, (terminales del circuito integrado) los cuales, dependiendo de la aplicación, son conectados a un sin fin de dispositivos como teclados, interruptores, sensores, servo-motores, etc.

#### 3.2.1.6 Puertos

Un puerto es un grupo de pines utilizado para enviar o recibir información, puede consistir únicamente de salidas, entradas o incluso una combinación de entradas y salidas. Actualmente la mayoría de los puertos son bidireccionales, es decir pueden ser configurados como pines de entrada o salida dependiendo de los requerimientos del usuario.

Usualmente cada puerto tiene asignada una dirección como si fuera un registro de memoria. La escritura a una dirección asignada a un puerto ocasiona que los pines asociados con la dirección del puerto sean forzados a un estado alto o bajo de acuerdo al valor escrito. Si los puertos no son mapeados en memoria, se tendrán instrucciones especiales de Entrada/Salida para acceder a ellos.

### 3.2.1.7 Características

Se están lanzando al mercado microcontroladores muy diversos con características particulares, que los diferencian de forma muy apreciable. Incluso un mismo fabricante pone a disposición de los usuarios varios modelos de microcontroladores, que basados en un mismo concepto tecnológico, disponen de muy diferentes posibilidades, pudiendo, de este modo el usuario elegir el modelo que mejor se ajuste a sus necesidades, no solo en cuanto a servicio o utilidad técnica, sino también económicamente.

Ahora se mencionarán algunas características generales de los microcontroladores:

- Disponen de circuito de reloj incorporado en el propio microchip.
- Poseen memoria de datos (RAM).
- Poseen memoria de programa (PROM, EPROM, EEPROM).
- Disponen de elementos de Entrada/Salida (puertos) para comunicarse con el exterior.
- Pueden colocarse en estado de bajo consumo (power down).
- Tienen un conjunto de instrucciones que facilitan la programación y confieren mayor rapidez en la ejecución del programa.
- Disponen de posibilidad de protección del programa.
- Poseen temporizadores programables.
- Trabajan con una o más interrupciones con diferentes niveles de prioridad.

El Microcontrolador seleccionado para esta aplicación es uno de la Familia ATMEL 8051, el AT89C52 ya que los puertos, la memoria de programa (ROM), la memoria RAM y la velocidad de funcionamiento de este dispositivo supera en mucho los requerimientos para la presente aplicación.

Para un funcionamiento adecuado se escogió un cristal que provee una frecuencia de oscilación de 11.059 MHz.

### 3.2.2 MAX 180<sup>19</sup>

El MAX 180 presenta varias características que lo hacen superior a los conversores tradiciones, tales como:

- Ocho canales para entrada de señales analógicas, con varias opciones: unipolar (0 – 5 V) o bipolar (-2.5 – 2.5 V) controlado con el pin BIP (0 y 1 lógico respectivamente), simple o diferencial (canales 3 – 4, 5 – 6 y 7 - 8) controlado con el pin DIFF (0 y 1 respectivamente).
- Posee un circuito de Track and Hold, lo que lo hace más preciso al momento de la conversión.
- La resolución digital de salida es de 12 bits.
- Bajo consumo de potencia (110 mW).

#### 3.2.2.1 Descripción de los componentes del MAX 180

El MAX 180 es un sistema de adquisición de datos de gran ancho de banda, cuya composición interna se detalla en el diagrama de bloques de la figura 3.2

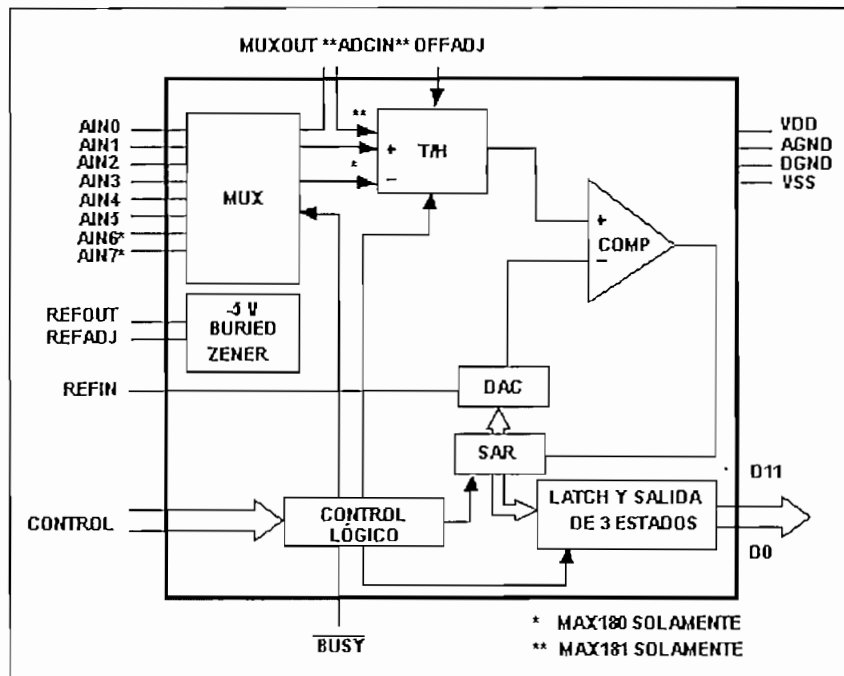


Figura 3.2. Diagrama de Bloques del MAX 180.

<sup>19</sup> <http://maxim-ic.com>



A continuación se describen los principales componentes internos del MAX 180:

**Multiplexer Analógico:** Este bloque permite elegir externamente uno de los ocho canales de entrada analógicos mediante los pines A0, A1, A2 y es habilitado internamente por la señal BUSY. La tabla 3.1. muestra los valores de estos pines y su respectiva habilitación:

**Circuito Track and Hold y Comparador:** En la figura 1.28 se ilustra la arquitectura de muestreo del comparador analógico del convertidor analógico digital (CAD), la capacitancia de entrada actúa como un capacitor retenedor del valor de entrada para cada conversión, la capacitancia se carga a través de un resistor interno de 1 k $\Omega$  en serie con la entrada.

Cuando la conversión inicia el C<sub>HOLD</sub> desconecta de la entrada positiva el T/H, entonces se inicia el muestreo de la señal, cuando el switch se cierra, la conversión termina y el capacitor C<sub>HOLD</sub> se reconecta nuevamente a la entrada.

El T/H comienza el rastreo cuando el CAD se deshabilita (BUSY = nivel lógico alto).

El modo Hold o Retención comienza 3 ciclos de reloj luego de que la conversión se inicia todos excepto en el modo Asynchronous Hold. Una variación en el retardo del modo Retención de una conversión a la siguiente (Jitter) es menor de 100 ps.

El tiempo requerido por el T/H para adquirir una señal de entrada está en función de qué tan rápido la capacitancia de entrada se carga.

Si la impedancia de la fuente de entrada es alta, el tiempo de adquisición se alarga y más tiempo transcurrirá entre conversiones.

A2	A1	A0	SE/DIFF	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7
0	0	0	0	+							
0	0	1	0		+						
0	1	0	0			+					
0	1	1	0				+				
1	0	0	0					+			
1	0	1	0						+		
1	1	0	0							+	
1	1	1	0								+
0	0	0	1	+	-						
0	0	1	1	-	+						
0	1	0	1			+	-				
0	1	1	1			-	+				
1	0	0	1					+	-		
1	0	1	1					-	+		
1	1	0	1							+	-
1	1	1	1							-	+

Tabla 3.1. Direccionamiento y Canal Seleccionado del MAX 180.

**Conversor Analógico Digital:** La operación de conversión utiliza el método de aproximaciones sucesivas y circuitería de entrada tipo Track and Hold, para convertir una señal analógica de entrada a una digital de salida de 12 bits. El control lógico permite al microcontrolador utilizar pocos componentes pasivos para la mayoría de aplicaciones. El T/H no requiere de un capacitor externo para su funcionamiento. Sin importar el modo o selección del interfaz, la secuencia es la misma y es la siguiente:

- El dato de entrada configura el latch del SAD y el interfaz indica al microcontrolador que la conversión ha iniciado.
- El Mux direcciona la señal de entrada seleccionada a la entrada del T/H.
- Un retardo permite al T/H tomar la señal. En todos los modos excepto en el Asynchronous Hold, este retardo es de tres ciclos de reloj. En el modo de excepción, el microcontrolador controla este retardo
- Los switches T/H se ubican en el modo Hold. La salida del T/H entrega una señal muestreada estable.
- Empieza el proceso de aproximaciones sucesivas, en donde se prueba y se establece cada uno de los 12 bits, desde el más significativo hasta el menos

significativo. La selección del bit ocurre en la transición negativa del CLKIN, para un total de 12 ciclos de reloj.

- f) El dato de salida es ubicado en los registros de salida, y el interfaz indica al microcontrolador que la conversión se ha completado y el dato está disponible.

### 3.2.2.2 Descripción de los Pines

PIN	NOMBRE	FUNCIÓN
1 – 8	AIN0 – AIN5	Entradas Analógicas al Mux: unipolar: 0V + 5V, bipolar: -2.5V +2.5V
9	REFIN	Entrada de Referencia
10	AGND	Tierra Analógica
11	REFOUT	Salida de Referencia -5V
12	REFADJ	Ajuste de Referencia en -5V, Conectar a $V_{DD}$ si no se requiere
13	OFFADJ	Ajuste del Offset, Conectar a $V_{DD}$ si no es requerido
14	MODE	Selección del modo del Interfaz
15	$V_{SS}$	Fuente negativa: -1.5 V o -12 V
16 – 19	D11 – D8	Salida de Datos de tres estados, MSB = D11
20	DGND	Tierra Digital
21 – 28	D7 – D0	Salida de Datos de tres estados, LSB = D0
29	CLKIN	Reloj de entrada compatible con TTL/CMOS
30	HBEN	Entrada de habilitación del Byte mayor
31	RD'	Entrada de Lectura
32	WR'	Entrada de escritura (MODE = 1 o abierto), Salida Lista (MODE = 0)
33	CS'	Entrada para selección del chip
34	BUSY'	Salida de indicación de chip ocupado
35	DIFF	Selección del modo simple (DIFF=0), o modo diferencial (DIFF =1)
36	BIP	Selección del modo unipolar (BIP=0), o modo bipolar (BIP =1)
37 – 39	A0 – A2	Entrada de las direcciones de selección del canal del mux, A0=LSB y A2=MSB
40	$V_{DD}$	Alimentación positiva: +5V de entrada.

Tabla 3.2. Pines del MAX 180 y su función<sup>20</sup>

### 3.2.3 MAX 233

Las líneas de transmisión y recepción del Microcontrolador tienen como salida y entrada una señal TTL, por lo que resulta necesario el uso de un circuito que permita manejar niveles RS-232 y hacerlo compatible con el puerto serial del Interfaz RS232/PCMCIA.

Las opciones analizadas permitieron seleccionar el dispositivo MAX233 porque es un circuito integrado que soluciona los problemas de niveles de voltaje

<sup>20</sup> <http://maxim-ic.com>

cuando se requiere enviar señales digitales con niveles RS-232, ya que solamente necesita una fuente de +5 V para su operación, después un elevador de voltaje interno convierte el voltaje de +5 V al de doble polaridad de  $\pm 10$  V. Como la mayoría de las aplicaciones de RS-233 necesitan de un receptor y un emisor, este dispositivo incluye en un solo empaque 2 parejas completas de transmisor y receptor.

La diferencia principal respecto al MAX232 es que ya no requiere de capacitores electrolíticos que son utilizados por el elevador de voltaje y el inversor de voltaje internos para obtener los voltajes con niveles RS-232.

En la tabla 3.3. se describen las principales características eléctricas.

Voltaje de Alimentación ( $V_{CC}$ ):		5 V $\pm$ 5%
Corriente de Alimentación:	Típica:	5 mA
	Máxima:	10 mA
Número de puertos de Recepción:		2
Número de puertos de Transmisión:		2
Velocidad de Transferencia:		120 kbps
Capacitores Externos:		No necesita (posee capacitores internos)
Fuentes de voltaje internas:	$V_+$	Min= ( $V_{CC} + 0.3$ V), Máx = +14 V
	$V_-$	Min=+0.3 V, Máx = -14 V
Voltajes de Entrada:	TTL	Min= -0.3 V, Máx = ( $V_{CC} + 0.3$ V)
	RS232	$\pm 30$ V
Voltajes de Salida:	TTL	Min= -0.3 V, Máx = ( $V_{CC} + 0.3$ V)
	RS232	Min= ( $V_+ + 0.3$ V), Máx = ( $V_- - 0.3$ V)

Tabla 3.3. Características Eléctricas del MAX 233

Para que un sistema de comunicación funcione correctamente, tanto el transmisor como el receptor deben tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y número de bits de parada.

El microcontrolador envía los datos serialmente con una velocidad de 19200 Baudios, 8 bits de datos, sin paridad y con un bit de parada.

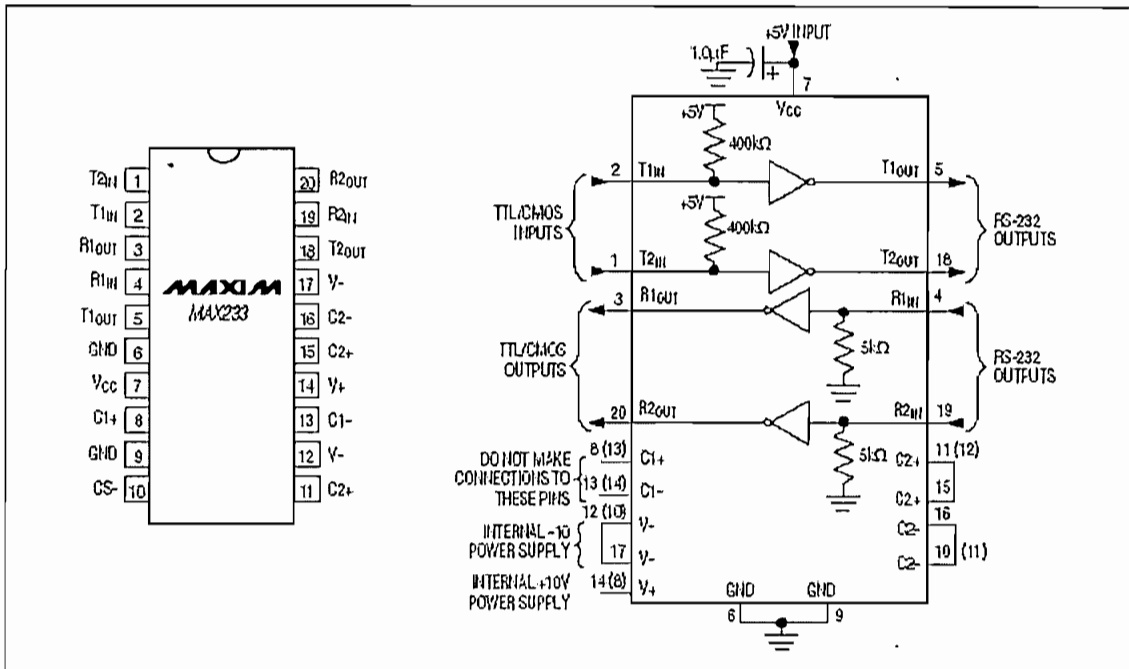


Figura 3.3. Distribución de pines y configuración interna del MAX 233

### 3.2.4 DIVISOR DE FRECUENCIA

El Sistema de Adquisición de Datos MAX 180 cuando se conecta directamente a un microcontrolador, como es el caso, tiene la opción de escoger el modo de operación denominada Interfaz Digital (MODE = nivel lógico alto), pero dadas las prestaciones de este dispositivo, el fabricante aconseja que la señal de reloj externa sea un divisor de la frecuencia con la que funciona el microcontrolador para evitar problemas de sincronismo.

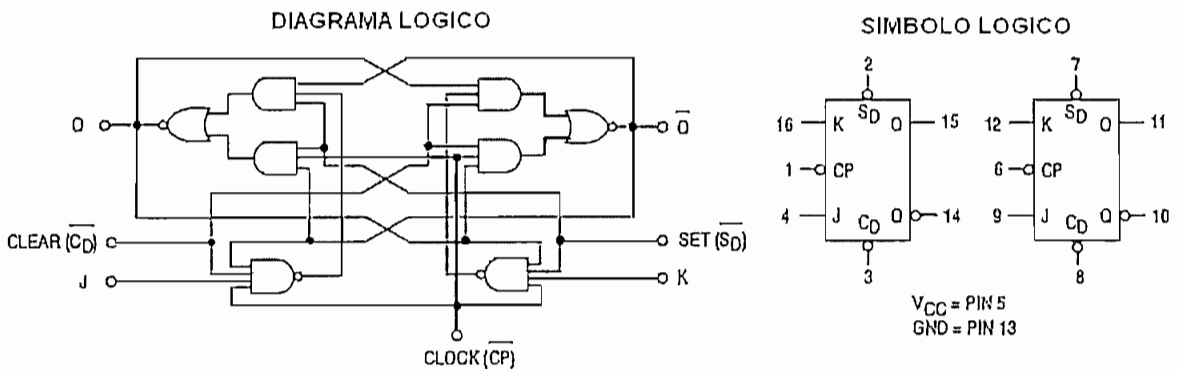


Figura 3.4. Diagrama y Símbolo Lógico del 74LS76

Por esta razón se divide la frecuencia entregada por el oscilador de 11.059 MHz por un factor de ocho, obteniéndose una frecuencia de 1.382 MHz con un ciclo de trabajo del 50%. Con este propósito se optó por un grupo de 3 flip-flops JK (el 74LS76<sup>21</sup>, que contiene un par de flip-flops tipo JK como se muestra en la figura 3.4.), que hagan las veces de un contador módulo 8, debido a la escalabilidad y flexibilidad para escoger un factor de división diferente si se requiriese. La figura 3.5. muestra la configuración utilizada.

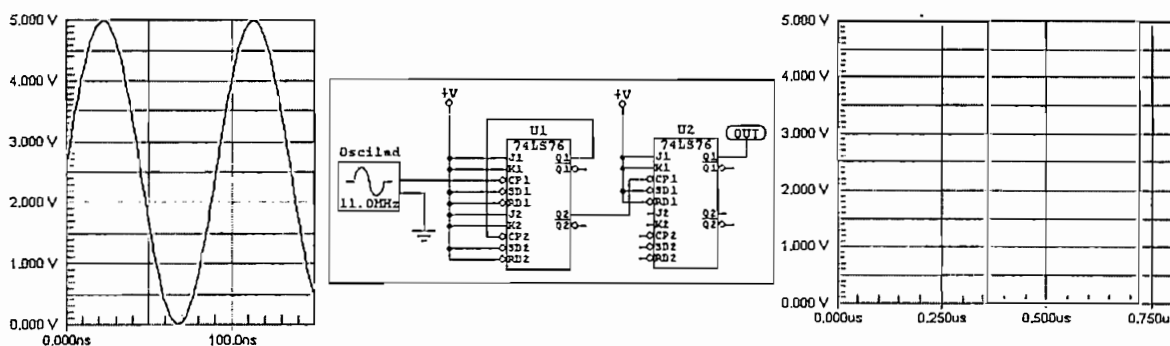


Figura 3.5. Configuración para obtener una frecuencia de 1.382 MHz

### 3.2.5 TRANSCEIVER

La configuración del Sistema de Adquisición de Datos requiere de un dispositivo controlador de la comunicación bidireccional entre el microcontrolador y el MAX 180, es decir que cuando el microcontrolador necesite realizar un proceso de escritura, la dirección sea llevada del microcontrolador al MAX 180 y el dato sea retenido en este dispositivo y enviado desde el microcontrolador al MAX 180; y en un proceso de lectura, que la dirección sea llevada del microcontrolador al MAX 180 y el dato sea leído del MAX 180, retenido en este dispositivo y enviado hacia el bus del microcontrolador.

El dispositivo más idóneo para desempeñar esta tarea es el Bus-Transceiver Octal 74LS245<sup>22</sup>, que es un transmisor-receptor para la comunicación

<sup>21</sup> <http://onsemi.com>

<sup>22</sup> [www.fairchildsemi.com](http://www.fairchildsemi.com)

de dos vías asincrónica entre dos buses de datos. La implementación de la función de control minimiza los requerimientos de temporización externos.

El dispositivo permite la transmisión desde el bus A hasta el bus B o del bus B hacia el bus A por medio del nivel lógico de entrada del control de dirección (DIR). La entrada de habilitación (G) puede ser usada para habilitar o deshabilitar al dispositivo y permitir la comunicación entre buses o aislarlos eficazmente. La tabla 3.4 ilustra lo descrito además la figura 3.6 muestra el diagrama lógico y distribución de pines del dispositivo.

Enable G	Dirección Control DIR	Operación
L	L	B Dato a A Bus
L	H	A Dato a B Bus
H	X	Aislado

H = Nivel ALTO  
L = Nivel BAJO  
X = Irrelevante

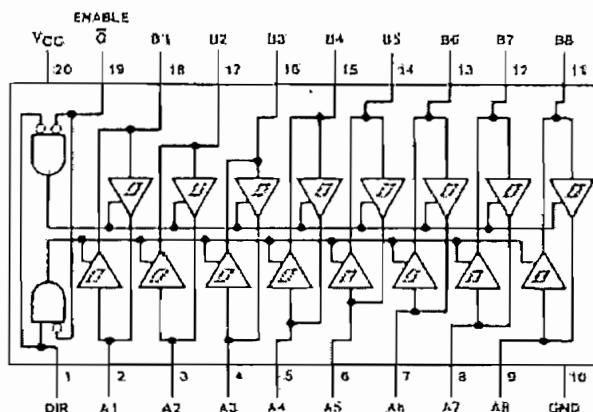


Tabla 3.4. Habilitaciones del 74LS245

Figura 3.6. Distribución de pines del 74LS245.

### 3.2.6 Alimentación de los Circuitos

La fuente de alimentación es una parte determinante para el buen funcionamiento del sistema, ya que sin una alimentación correcta y suficiente, el rendimiento o eficiencia del sistema estará muy por debajo del que normalmente debe ofrecer. Por esta razón, es conveniente contar con una fuente de alimentación para proporcionarle a los elementos o circuitos de cada uno de ellos, el voltaje y la corriente necesaria para su operación. La tabla 3.5. muestra los requerimientos en voltaje y corriente de los dispositivos utilizados en el diseño del Sistema de Adquisición de Dato

Dispositivo	Voltaje de Alimentación	Nivel de Consumo
MAX 180	+5V, -12V	110 mW
MAX 233	+5V	300 mW
74LS76 (x2)	+5V	50 mW
74LS245	+5V	50 mW
AT89C52	+5V	125 mW

Tabla 3.5. Voltaje de alimentación y Potencia consumida por cada dispositivo.

Por tanto para este proyecto se necesitan dos fuentes de polarización: +5V y -12V. La fuente de +5V demanda una potencia máxima de 700 mW, además de los leds de señalización, alimentación para las aplicaciones y demás requerimientos, por lo que el regulador de voltaje LM7805<sup>23</sup> que provee de hasta 1A y +5V (es decir 5W) es más que suficiente para alimentar este circuito, entonces la única preocupación es encontrar un transformador de voltaje que suministre la potencia necesaria, pero una potencia de este valor se obtiene fácilmente con un transformador de tamaño pequeño. Por otro lado, el voltaje de -12V, necesario para el funcionamiento del MAX 180 es únicamente un voltaje de referencia, por lo que el regulador de voltaje LM7912<sup>24</sup> provee este voltaje sin problema, debiéndose tener en cuenta que a la salida del transformador debe tenerse al menos 12 V (lo sugerido es 15V).

### 3.3 APLICACIONES

El Sistema de Adquisición de Datos está provisto de ocho entradas analógicas normales o de 4 entradas diferenciales, por lo que cualquier señal que sea adaptada al rango del voltaje de entrada (Unipolar: 0 – 5V y Bipolar: -2.5 - +2.5V), por lo que puede ser alimentado por la salida de cualquier transductor, teniéndose cualquier variable para sensar: temperatura, presión, luminancia, fuerza, velocidad, humedad, peso, etc.; pero el objetivo de este proyecto no es realizar un diseño complejo de una aplicación específica, sino más bien comprobar el funcionamiento del SAD, por lo que se implementa una aplicación básica: medición de temperatura.

<sup>23</sup> <http://www.national.com>

<sup>24</sup> <http://www.national.com>



### 3.3.1 MEDICIÓN DE TEMPERATURA

Para este efecto existen dispositivos termoelectricos tales como: termocuplas, resistencias detectoras de temperatura, termistores, sensores embebidos en un circuito integrado, y cualquier elemento que transforme un cambio de temperatura en un flujo de corriente eléctrica o un nivel de voltaje, siendo comúnmente escogidos los materiales termoelectricos que presenten una relación temperatura – voltaje o corriente lo más cercana a la linealidad.

Los sistemas de medición de temperatura más utilizados son: grados Fahrenheit (°F) y grados Celsius (°C); en donde el punto de solidificación del agua (y de fusión del hielo) es de 32 °F, y su punto de ebullición es de 212 °F, en escala centígrada o Celsius, se asigna un valor de 0 °C al punto de congelación del agua y de 100 °C a su punto de ebullición.

En ciencia, a nivel experimental, la escala más empleada es la escala absoluta o Kelvin en donde el cero absoluto, que está situado en -273,15 °C, corresponde a 0 K, y una diferencia de un kelvin equivale a una diferencia de un grado en la escala centígrada.<sup>25</sup>

Las fórmulas de conversión son las siguientes:

$$^{\circ}\text{C} = 5/9 (^{\circ}\text{F} - 32)$$

$$^{\circ}\text{F} = 9/5 ^{\circ}\text{C} + 32$$

$$^{\circ}\text{C} = ^{\circ}\text{K} - 273.15$$

Donde:

°C → Temperatura en Grados Celsius

°F → Temperatura en Grados Fahrenheit

°K → Temperatura en Grados Kelvin

A continuación se describen brevemente varios transductores:

---

<sup>25</sup> <http://es.encarta.msn.com>

### 3.3.1.1 Termistores

Los termistores, o resistores térmicos, son dispositivos semiconductores que se comportan como resistencias con un coeficiente de temperatura de resistencia alto y, generalmente negativo. En algunos casos, la resistencia de un termistor a temperatura ambiente puede disminuir hasta un 6% por cada 1°C que se eleve la temperatura. Dada esta alta sensibilidad al cambio de temperatura hacen al termistor muy conveniente para mediciones, control y compensación con precisión de la temperatura. El uso de termistores está muy difundido en tales aplicaciones, en especial en el rango más bajo de temperatura de -100°C a 300°C.<sup>26</sup>

Los termistores se componen de una mezcla sintética de óxidos de metales, como manganeso, níquel, cobalto, cobre, hierro y uranio. Su rango de resistencia va de 0.5 ohms. a 75 ohms y están disponibles en una amplia gama de formas y tamaños.

### 3.3.1.2 Termocuplas

Una termocupla o termopar es un dispositivo de estado sólido que consta de dos metales diferentes empalmados en una junta, siendo utilizados para la fabricación de termopares, materiales como: hierro y constantano, cobre y constantano o antimonio y bismuto. Los termopares comerciales se designan por letras (T,E,J,K,R) que identifican los materiales que los componen y se especifican generalmente por su sensibilidad o coeficiente térmico mV/°C.

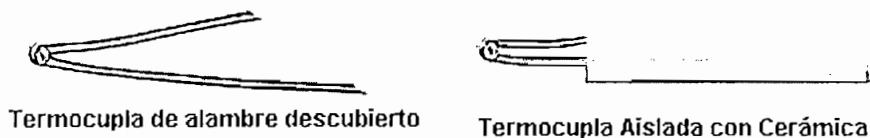


Figura 3.7. Aspecto Físico de una Termocupla.

<sup>26</sup> <http://proton.ucting.udg.mx/temas/control/memo/MEMO.html>

Calentando la zona de unión se produce una transferencia de electrones de un metal al otro (uno de ellos siempre tiene más tendencia a captar electrones que el otro), si se conecta los extremos no unidos a un circuito externo circulará por él, continuamente, una corriente en un sentido, siendo más alta la corriente que circula cuanto más se calienta la unión. Para que exista una diferencia de potencial debe haber entre los metales dos soldaduras: una fría y otra caliente.<sup>27</sup>

### 3.3.1.3 Resistencias Detectoras de Temperatura (RTD)

Las RTD son dispositivos más eficientes que otros sensores de temperatura debido a que su respuesta es más lineal eliminando así redes correctivas y errores de interpretación, operan en rangos de temperatura  $-400\text{ }^{\circ}\text{F}$  a  $1700\text{ }^{\circ}\text{F}$ , tienen un coeficiente positivo de temperatura lo que significa que al aplicar calor a la RTD la resistencia de ésta aumenta por lo que la corriente disminuye, al quitar el calor la resistencia de RTD disminuye y la corriente aumenta.

El instrumento de medida puede ser calibrado para que los cambios de corriente representen cambios de temperatura.

Para la fabricación de las RTD se toma en cuenta una característica térmica, conocida como alfa ( $\alpha$ ), que no es más que el coeficiente positivo de temperatura cuyo valor varía según el conductor (cobre, platino, tungsteno, níquel), siendo el material más apropiado para la fabricación de las RTD el platino con  $\alpha = 0.0039$ , eliminando así las precauciones que se toman con otros tipos de sensores.

Una RTD está compuesta de un sensor, hecho de alambre de platino envuelto en un núcleo de cerámica y está montado en el interior de una vaina de acero inoxidable brindando buena transferencia térmica y protección contra la humedad y por medio del cual se mide la temperatura.

---

27

[http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicaInteractiva/sacaleE\\_M2/termoelectricidad/Termoelectricidad.htm](http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicaInteractiva/sacaleE_M2/termoelectricidad/Termoelectricidad.htm)

Los valores de resistencia estándar para las RTD son pequeños y están desarrollados por la industria a temperaturas específicas, por lo tanto se debe tomar en cuenta el efecto de la impedancia de los hilos de medición que puede ser de algunos ohmios o decenas de ohmios, esta pequeña impedancia podría contribuir a un error significativo en la medición de la temperatura.

El método clásico para evitar este problema ha sido el uso de puentes.

#### 3.3.1.4 Sensores de temperatura embebidos en un Circuito Integrado

Son dispositivos que proporcionan un voltaje de salida en función de la temperatura que se aplica sobre ellos y el voltaje de alimentación.

Los más comúnmente utilizados operan como un zener típico de dos terminales, consiguiendo que el voltaje de ruptura sea directamente proporcional a la temperatura.

La familia de sensores LM135 son un ejemplo de estos dispositivos: voltaje de ruptura proporcional a la temperatura absoluta con una variación de +10 mV por cada grado kelvin de variación, opera sobre un rango de corriente de 400  $\mu$ A a 5 mA virtualmente sin cambios en su desempeño, poseen una salida lineal, lo que hace especialmente fácil el interfaz de su lectura y diseño de los circuitos de control.

Típicamente entrega el valor de temperatura con un error menor a 1°C en un rango de temperatura de 100°C, Los sensores que conforman esta familia son: LM135 que trabaja en un rango de temperatura que va de -55°C hasta +150°C, el LM235 cuyo margen de trabajo va desde -40°C hasta +125°C y el LM335 cuyos límites son: -40°C a +100°C. <sup>28</sup>

---

<sup>28</sup> <http://www.national.com>

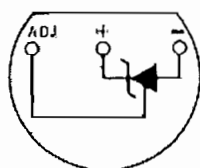
### 3.3.2 Diseño del Sensor de Temperatura

Para este proyecto, la opción más idónea a utilizar para diseñar e implementar una aplicación para medición de temperatura es en base a un sensor embebido en un circuito integrado, ya que no requiere circuitos ni sistemas de compensación, calibraciones complejas, como se mencionó anteriormente, presenta una respuesta lineal de voltaje en función a la temperatura, es de muy bajo consumo de potencia, lo que lo hace muy atractivo.

El integrado escogido es el LM335, cuyas características se enuncian a continuación:

- Directamente calibrado en grados kelvin.
- Dispone inicialmente de una precisión de 1°C.
- Trabaja desde 400  $\mu$ A hasta 5 mA.
- Presenta una impedancia dinámica inferior a 1  $\Omega$ .
- Sensa desde -40°C hasta +100°C

En la figura 3.8. se presenta un gráfico de este circuito integrado (en su configuración física plástica tipo TO-92<sup>29</sup> )



Vista Inferior

Figura 3.8. Configuración física y distribución de pines del LM335.

El fabricante propone circuitos básicos como los de la figura 3.9. Pero para evitar cualquier problema (principalmente que el cable con el que se llega al sensor sea demasiado largo, por tanto la caída de voltaje que presenta no sea despreciable o se puedan introducir señales en ambientes ruidosos que alteren la señal entregada) se optó por una alimentación para el sensor en una configuración como fuente de corriente, como detalla la figura 3.10.

<sup>29</sup> <http://www.national.com>

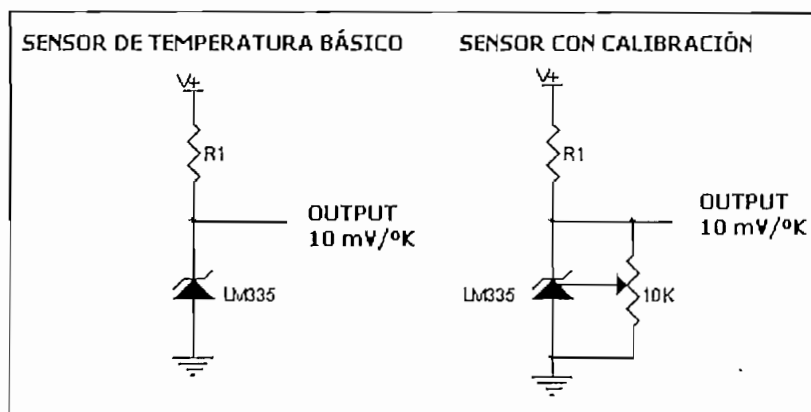


Figura 3.9. Configuraciones Básicas del Sensor LM335

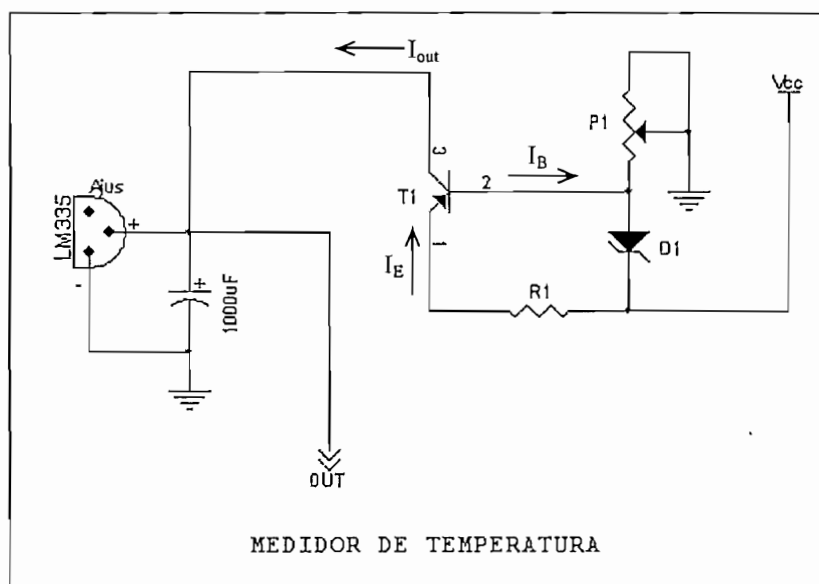


Figura 3.10. Sensor de temperatura alimentado con una fuente de corriente.

Por lo que para el diseño, lo fundamental es la corriente de alimentación del sensor o  $I_{out}$ , que puede variar desde  $400 \mu\text{A}$  hasta  $5 \text{ mA}$ , entonces se escoge un valor intermedio con el que no se tenga ninguna complicación de sobrecarga o subfuncionamiento del sensor:  $1 \text{ mA}$ .

Dado que para este caso no se necesita de un transistor con características especiales, sino más bien uno de propósito general que maneje fácilmente una corriente de colector de esta magnitud se escoge el transistor 2N3702, que soporta idealmente hasta  $300 \text{ mA}$ , con lo que satisface este requerimiento.

El diodo escogido para estabilizar los voltajes de polarización es en este caso el NTE5007A, que es un zéner de 3.9 V, con lo que se tienen las siguientes relaciones:

$$V_Z = 3.9 \text{ V}$$

$$V_B = V_{CC} - V_Z$$

$$V_B = 5\text{V} - 3.9\text{V}$$

$$V_B = 1.1 \text{ V}$$

$$V_E = V_B + V_{BE}$$

$$V_E = 1.1\text{V} + 0.6\text{V}$$

$$V_E = 1.7 \text{ V}$$

$$I_E \approx I_C$$

$$I_E = V_E / R_1$$

$$R_1 = V_E / I_E$$

$$R_1 = 1.7 \text{ V} / 1 \text{ mA}$$

$$R_1 = 1.7 \text{ k}\Omega$$

Finalmente el potenciómetro P1 simplemente sirve para regular la corriente de base, tanto para estabilidad de polarización como también para que, de ser necesario, se compense la corriente de emisor y la corriente de colector sea la requerida:

$$I_E = I_C + I_B$$

Por lo tanto

$$I_C = I_E - I_B$$

### 3.4 CONFIGURACIÓN DEL SOFTWARE DEL SISTEMA DE ADQUISICIÓN DE DATOS

Los programas realizados tanto para el microcontrolador en lenguaje ensamblador como en el computador para la visualización de los datos con una herramienta computacional para este propósito (en este caso Visual Basic), tienen como objetivo principal que el microcontrolador adquiera el dato en forma de

lectura de Memoria Externa desde el MAX 180 (bus paralelo) y lo envíe transformado en un caracter serial por el puerto del microcontrolador hacia el computador, en formato en el que se obtuvo originalmente (hexadecimal). Esta lectura tiene que hacerse a los 8 canales analógicos (ya que en este caso se tienen entradas unipolares no diferenciales) y mostrarlos eligiendo el o los canales a sensar y la forma de presentación: decimal o hexadecimal, lectura única o de forma continua.

El diagrama que se muestra en la figura 3.11. detalla las directrices que se siguieron para la elaboración de los programas que en conjunto muestran los datos leídos del MAX 180. Para el microcontrolador de la familia 8051, se consideró una velocidad de transmisión serial de 19200 bps, con ocho bits de datos, sin bits de paridad y un solo bit de parada; no se utiliza la opción de detección de velocidad ya que la transmisión inalámbrica posterior se hace a alta velocidad (1 MBps) , y no se escoge un valor de velocidad estándar mayor para evitar errores en la transmisión dadas las limitaciones del microcontrolador y su oscilador.

A continuación se presenta la subrutina más importante del microcontrolador: la implementación de la lectura de los datos del MAX 180.

#### 3.4.1 LECTURA DEL MAX 180

Esta subrutina lo único que hace es enviar las señales de control necesarias para la lectura: Habilitación del chip CS en nivel lógico bajo e igualmente HBEN (para el caso del primer byte) y el valor correspondiente a DIFF, BIP y el número de canal a sensar ( $DIFF = BIP = 0_L$ ) como si se diese la dirección de una memoria externa para escribir el dato 00H, luego se espera que se termine la conversión sensando el bit BUSY y se lee el dato igualmente como si fuese una memoria externa.

Para la lectura del segundo byte se cambia el pin HBEN a nivel lógico alto y se lee el dato. El diagrama de la figura 3.12. muestra lo descrito.



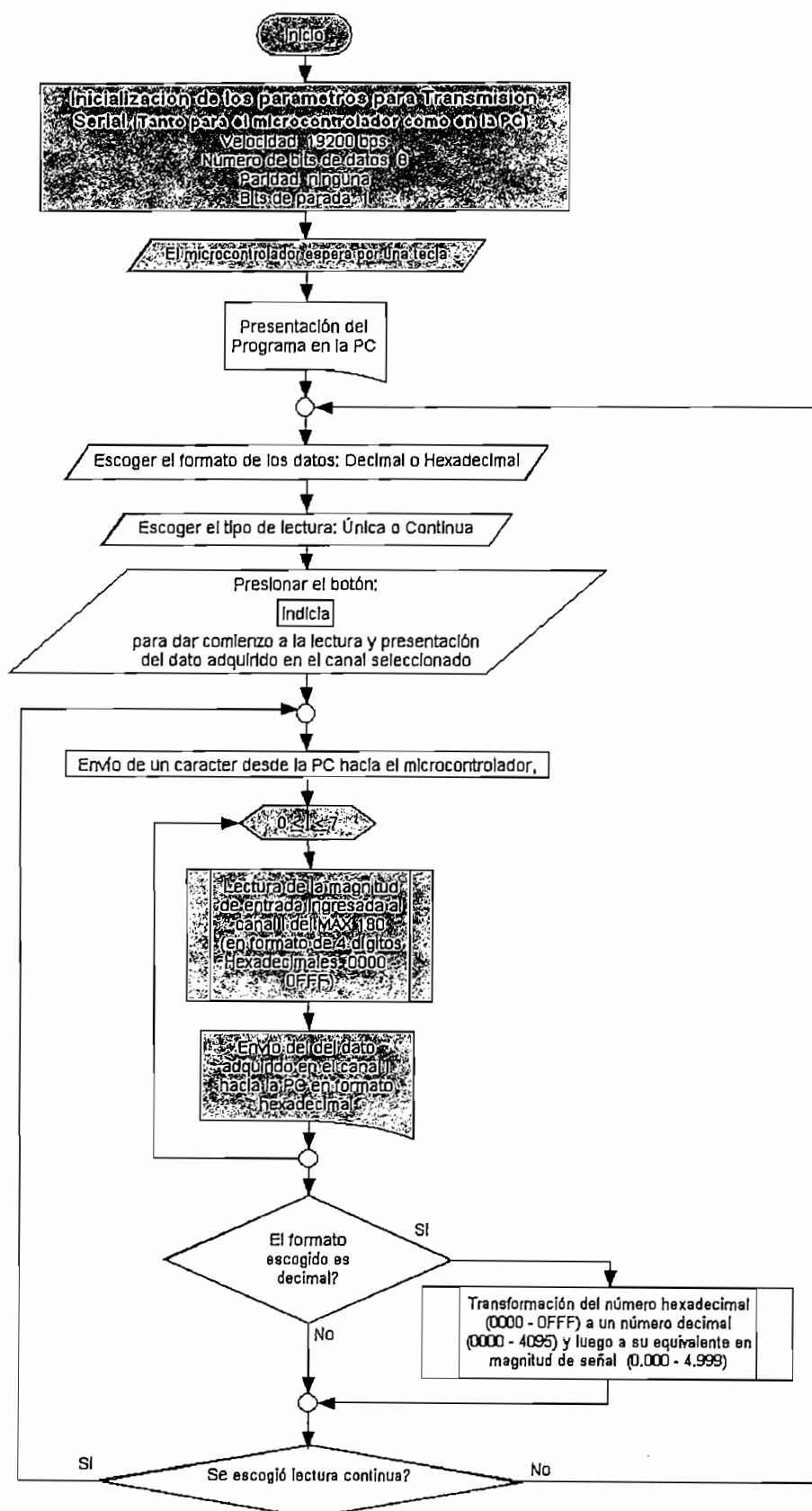


Figura 3.11. Diagrama de Flujo del Software implementado.

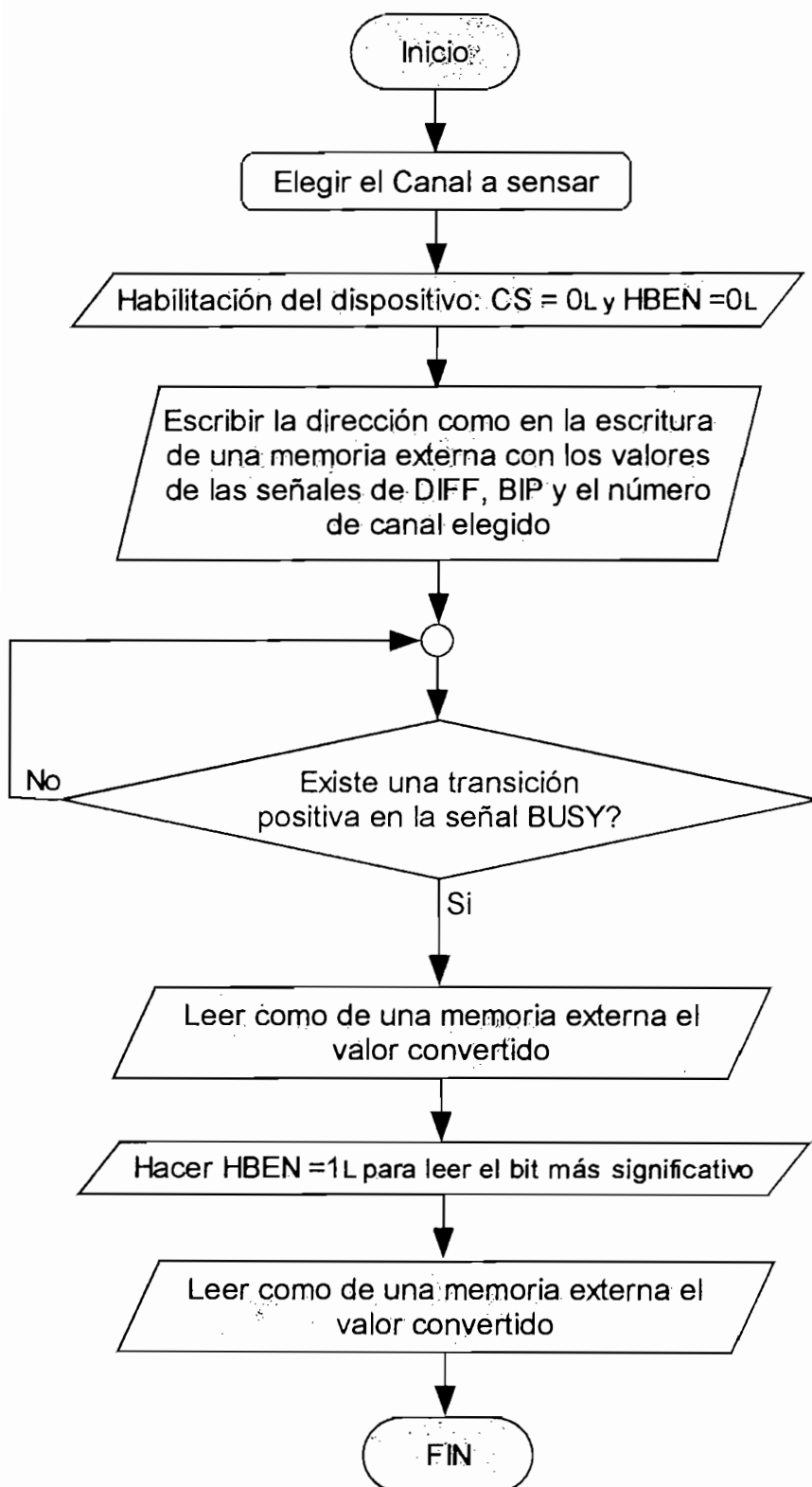


Figura 3.12. Diagrama de flujo de la lectura de un dato del MAX 180

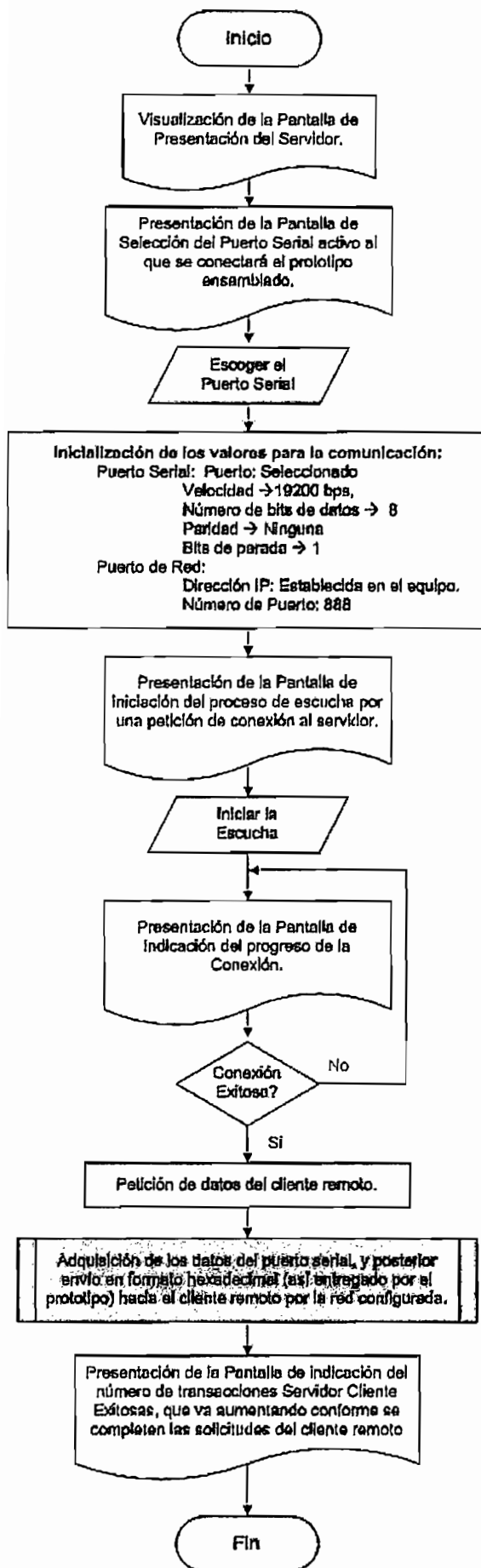


Figura 3.13. Diagrama de flujo del Software Servidor

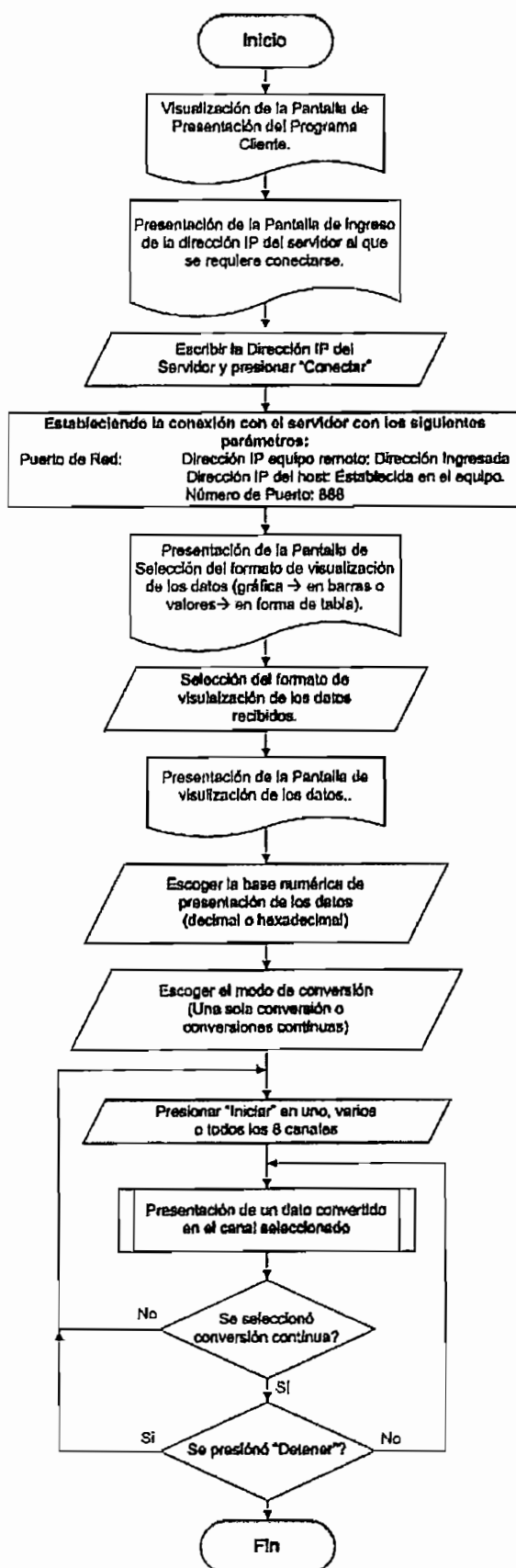


Figura 3.14. Diagrama de flujo del Software Cliente

### **3.4.2 SOFTWARE IMPLEMENTADO EN EL SERVIDOR**

La figura 3.13. muestra el diagrama de flujo del software en el Servidor para iniciar el paso de adquisición de datos, seleccionando el puerto serial activo que iniciará el proceso de escucha por una petición de conexión para luego adquirir los datos del puerto serial y enviarlo hacia el cliente remoto por la red.

Además el programa muestra en número de transacciones Servidor/Cliente exitosas que aumenta según se satisfagan las peticiones de datos del cliente.

### **3.4.3 SOFTWARE IMPLEMENTADO EN EL CLIENTE**

La figura 3.14. muestra el diagrama de flujo del software en el Cliente para visualizar la adquisición de datos, ingresando la dirección IP del servidor a conectarse. El programa da la opción de visualizar los datos en forma gráfica o en valores que pueden ser escogidos en una base numérica decimal o hexadecimal.

## **CAPÍTULO 4**

### **DISEÑO E IMPLEMENTACIÓN DEL SISTEMA, PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS.**

En este capítulo se describe la implementación total y los resultados obtenidos en cada una de las etapas de las pruebas del Sistema desarrollado. Al final de las pruebas se espera conseguir la visualización de las mediciones del valor de voltaje y temperatura adquiridos a través del prototipo desarrollado, en un punto remoto de la red inalámbrica a configurar, preferentemente utilizando un equipo que permita el acceso al sistema de distribución AP (Access Point).

Al inicio de este capítulo se muestra la interconexión de los elementos que integran el sistema completo, posteriormente se presentan los datos obtenidos del Sistema de Adquisición de Datos, así como las tablas de resultados obtenidas durante la medición de temperatura y voltaje desde un terminal remoto, generado en diferentes condiciones de prueba tanto para un sistema cliente/servidor entre PC como para el dispositivo EZLW300,

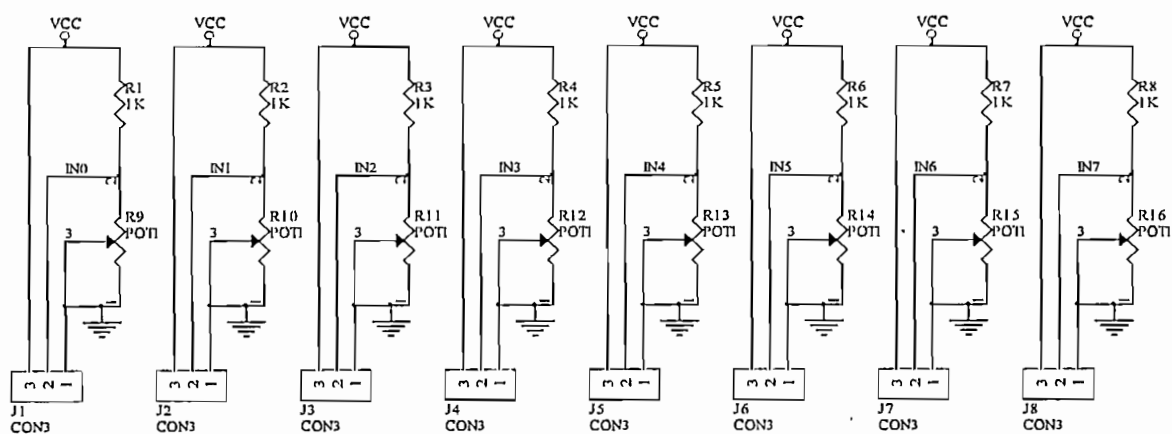
Sin duda alguna, los resultados que se obtienen (tiempos de respuesta, retardos, datos adquiridos, datos visualizados, etc.) y las conclusiones que éstos producen es lo más importante al finalizar el diseño, la implementación y las pruebas correspondientes de este proyecto.

## 4.1 DESCRIPCIÓN DE LAS TARJETAS Y DISTRIBUCIÓN DE LOS ELEMENTOS QUE CONFORMAN EL PROTOTIPO.

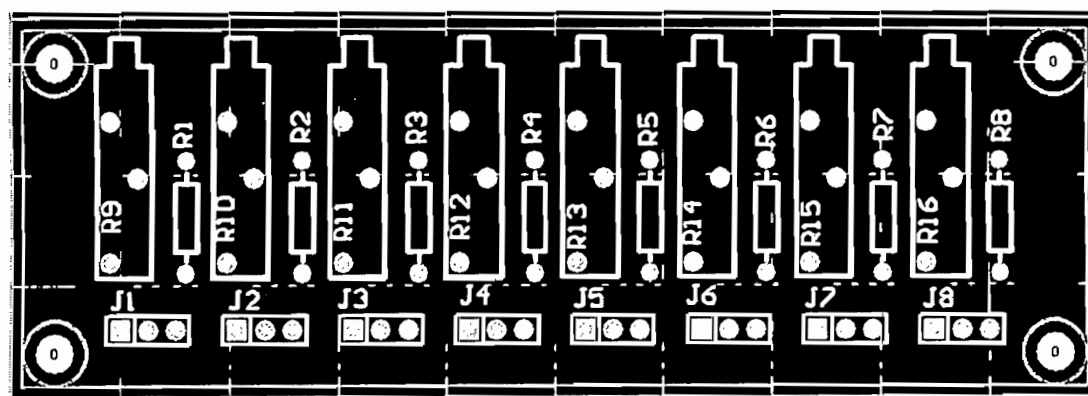
El sistema o prototipo está compuesto por cuatro tarjetas de circuito impreso que se describen a continuación:

### 4.1.1 TARJETA DE VARIACIÓN DEL VALOR LAS SEÑALES DE ENTRADA.

En la figura 4.1. se muestran los esquemas del circuito para la simulación de variación de las señales analógicas de entrada, conformado por ocho potenciómetros (10 K) con sus respectivas resistencias (1K) que permiten variar el voltaje (0 - 5 V).



a) DIAGRAMA ESQUEMÁTICO

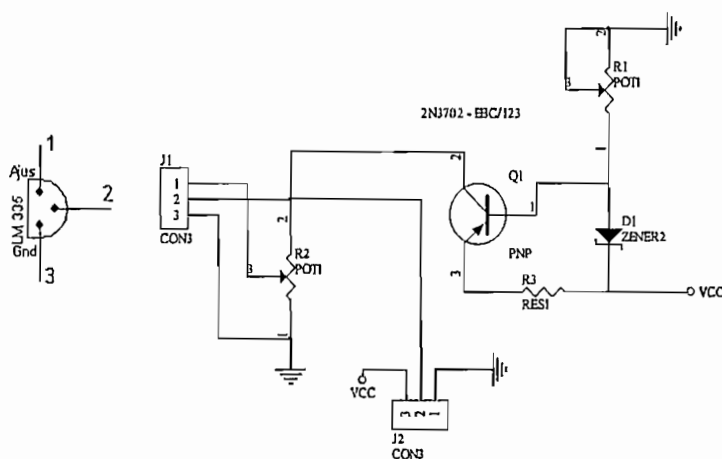


b) PLACA DE CIRCUITO IMPRESO (PCB)

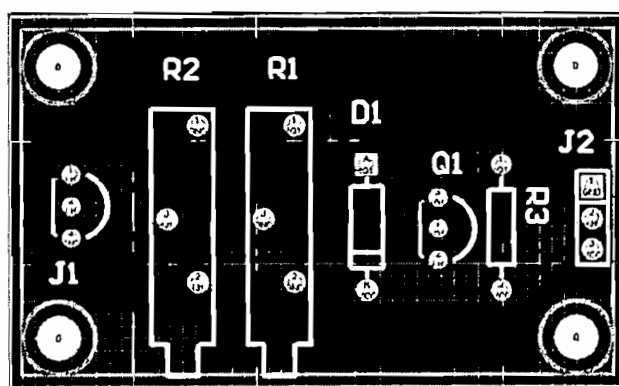
Figura 4.1. Tarjeta de canales variables.

#### 4.1.2 SENSOR DE TEMPERATURA

En la figura 4.2. se ilustra los esquemas del circuito del sensor de temperatura que permite obtener la variación de la temperatura a partir del voltaje, esto se logra gracias al sensor LM335 que complementa el diseño con dos potenciómetros (1 y 10 K), un diodo zener NTE5007A, un transistor 2N3702 y una resistencia de 120 ohmios.



a) DIAGRAMA ESQUEMÁTICO



b) PLACA DE CIRCUITO IMPRESO (PCB)

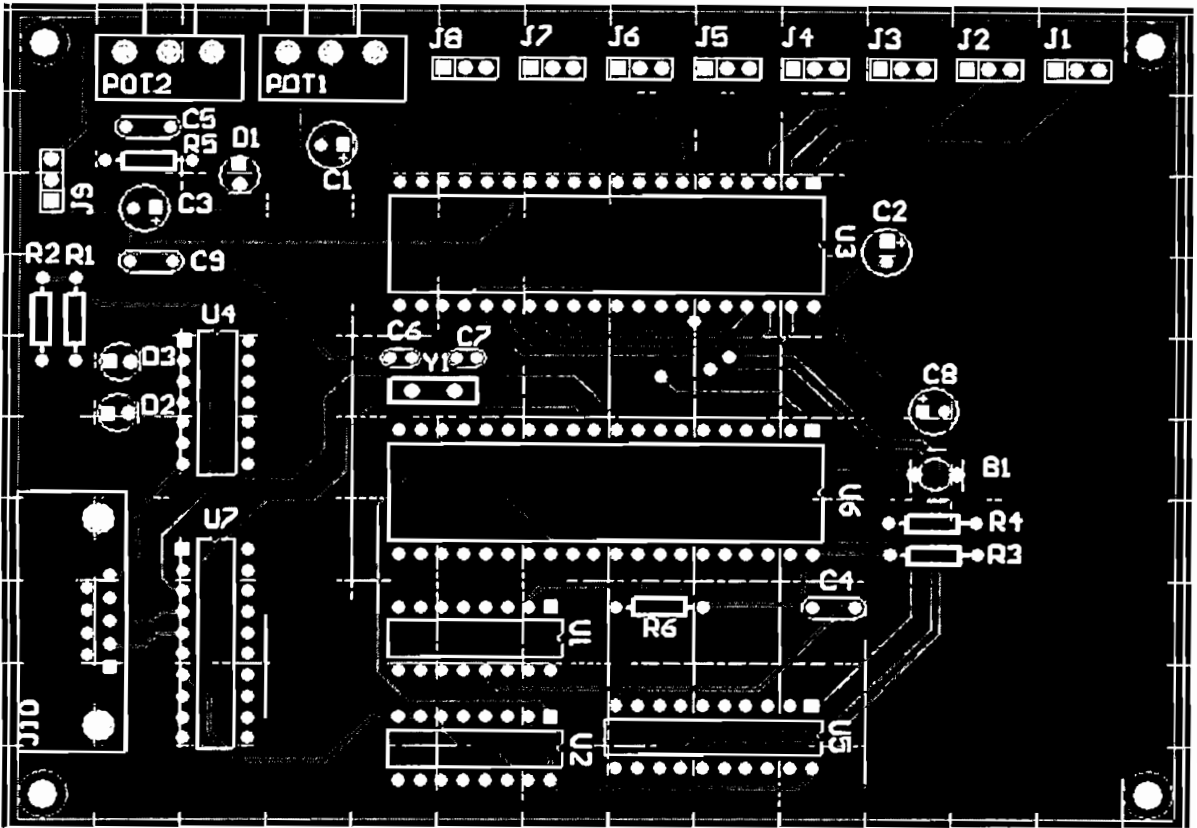
Figura 4.2. Tarjeta del Sensor de Temperatura.

#### 4.1.3 SISTEMA DE ADQUISICIÓN DE DATOS

En la figura 4.3. se ilustra la tarjeta de circuito impreso del subsistema que permite la captura de datos y está compuesto principalmente por los circuitos integrados: MAX180, encargado de la adquisición de datos, el microcontrolador







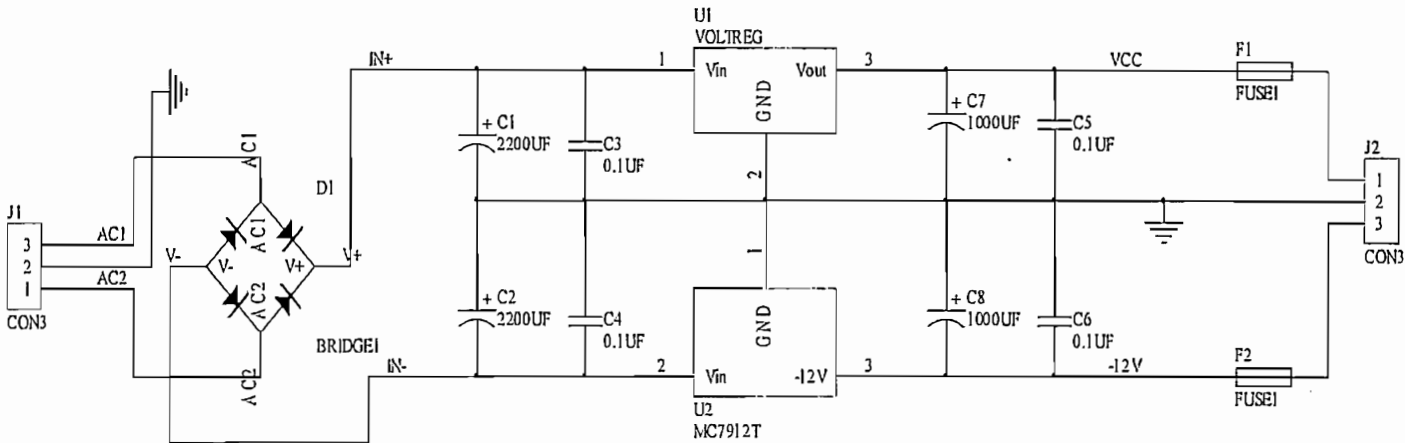
b) PLACA DE CIRCUITO IMPRESO (PCB)

Figura 4.3. Tarjeta del Sistema de Adquisición de Datos.

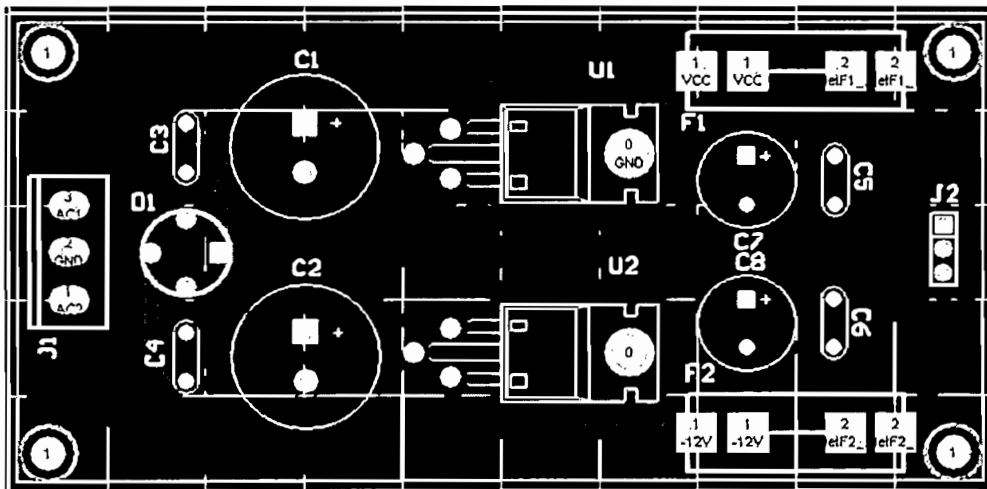
#### 4.1.4 TARJETA DE ALIMENTACIÓN

En la figura 4.4. se ilustra la fuente de alimentación del sistema y sus conexiones.

- Entrada de la señal rectificadora. Son dos señales que provienen de los circuitos rectificadores LM7805 y LM7912 que a su vez están alimentadas por un transformador de 120V/2A, además esta un circuito de protección para esta fuente de alimentación.
- Salida de voltaje regulado. Estos conectores proporcionan un voltaje de -12V/1A al MAX 180 y 5V/1A al resto del circuito.



a) DIAGRAMA ESQUEMÁTICO.



a) PLACA DE CIRCUITO IMPRESO (PCB)

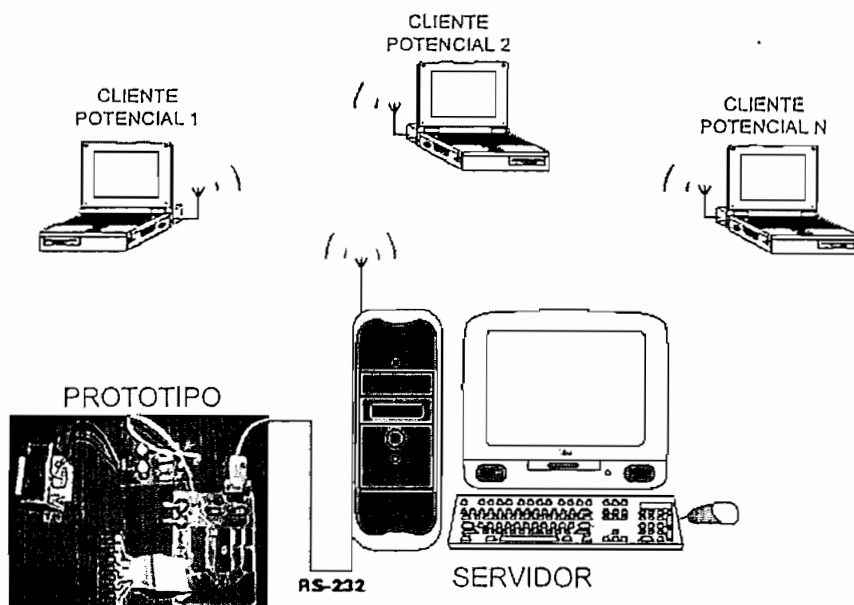
Figura 4.4. Sensor de Temperatura.

## 4.2 INSERCIÓN DEL PROTOTIPO EN LA RED INALÁMBRICA.

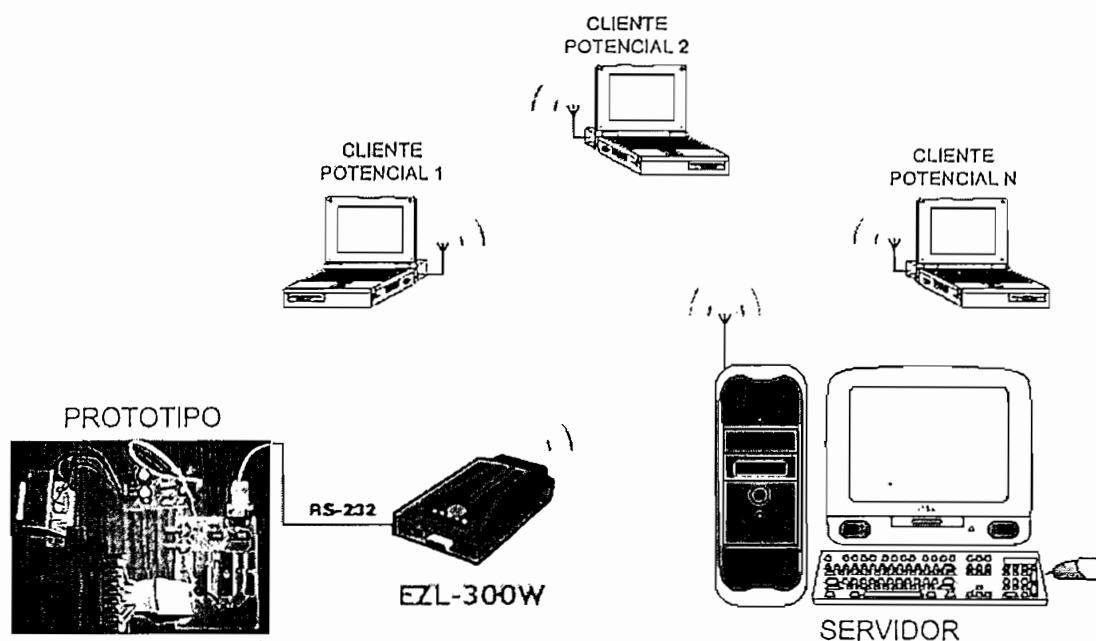
### 4.2.1 Configuración del Sistema

El prototipo se comunica mediante su puerto serial con un computador servidor, que opera como Access Point y hace las veces de nexo entre el prototipo y los demás equipos que conforman la red inalámbrica y son potenciales clientes o

receptores de la información adquirida por el prototipo. En la figura 4.5. se muestra de una forma explícita las configuraciones a implementar.



a) CONEXIÓN DIRECTA DEL PROTOTIPO CON EL SERVIDOR



b) CONEXIÓN DEL PROTOTIPO CON EL EZL300W

Figura 4.5. Configuraciones de la conexión del prototipo con la red inalámbrica.

Cabe señalar que debido a las limitaciones de la herramienta utilizada para interconectar el prototipo con el cliente (Winsock), la conexión es uniusuario.

A continuación se describen los elementos constitutivos de la red.

#### 4.2.1.1 Servidor

El equipo servidor puede ser un Access Point o, como en este caso, un PC con las suficientes prestaciones como para soportar el software para convertirlo en Access Point y todas las demás herramientas requeridas (servicio de DHCP de ser necesario, tarjeta de red inalámbrica cuyo transmisor-receptor entregue una potencia apropiada, ubicación adecuada dentro de la red a la que pertenece, etc.).

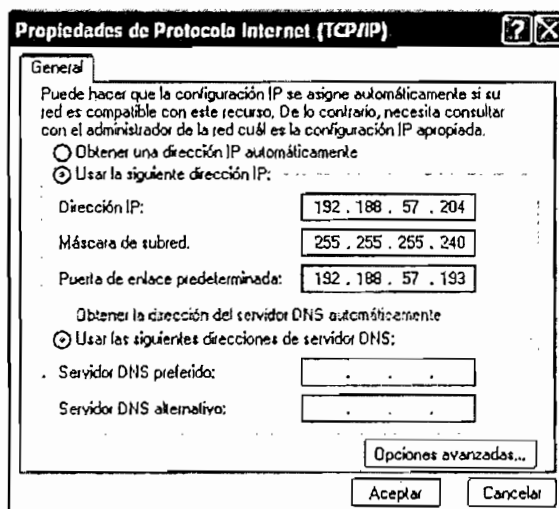


Figura 4.6. Configuración de la tarjeta de red en el Servidor.

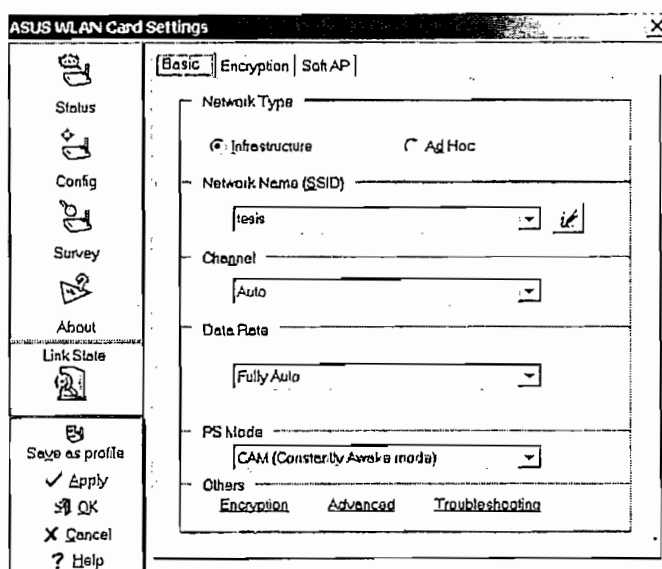


Figura 4.7. Configuración del software en el Servidor.

Una vez establecido el PC como Access Point, se procede a modificar la configuración de su tarjeta inalámbrica, para poder formar la red con los clientes potenciales del prototipo. En las figuras siguientes se muestra la configuración utilizada en esta prueba, tanto de la tarjeta de red inalámbrica como del software utilizado.

#### 4.2.1.2 Clientes Potenciales

Puede ser cualquier computador con una tarjeta de red inalámbrica ubicada dentro de la zona de cobertura del servidor, configurada adecuadamente dentro de la red. En esta prueba se utilizó un computador portátil o Laptop con una tarjeta de red PCMCIA (la dirección IP respecto a la del servidor, únicamente cambia en el conjunto de dígitos correspondiente al Host).

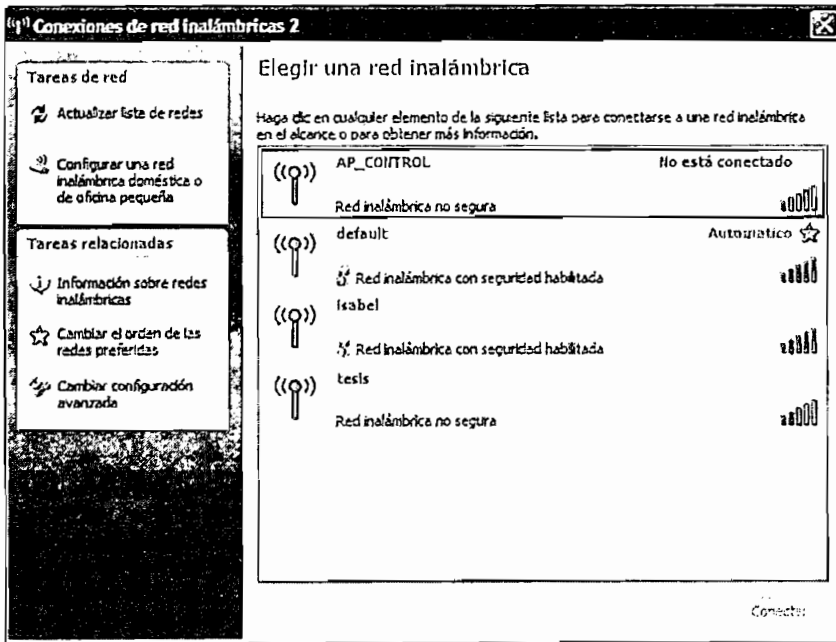


Figura 4. 8. Redes en la zona de recepción de la tarjeta inalámbrica

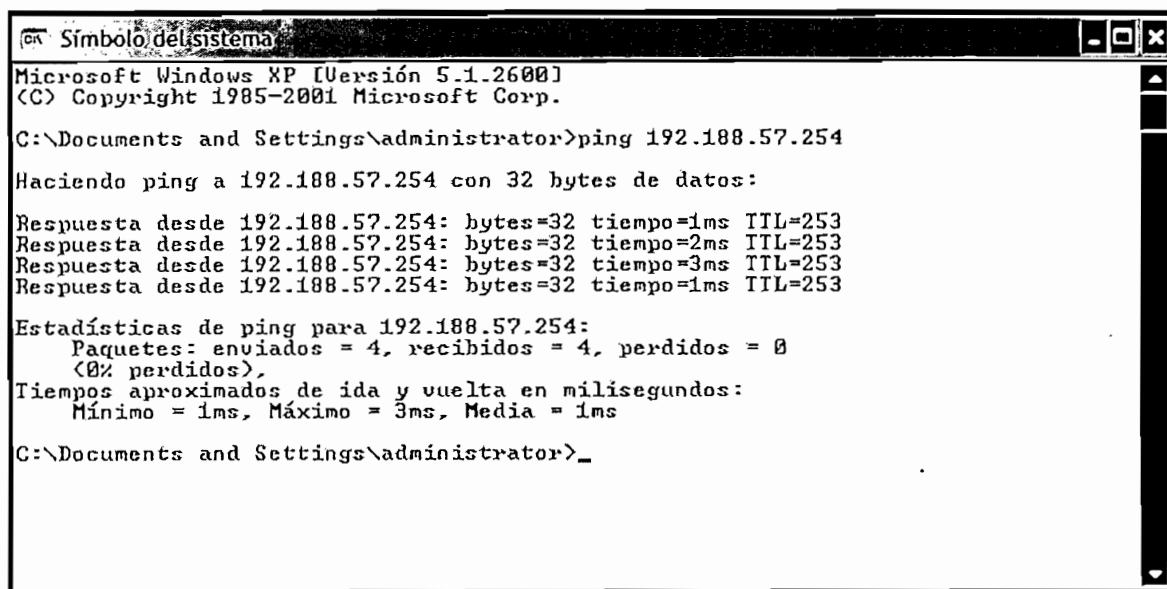
La dirección asignada corresponde a una dirección clase C la misma que ha sido dividida en 16 subredes, de las cuales se ha utilizado la subred 192.88.57.192 con dirección de broadcast 192.188.57.207 y máscara de red 255.255.255.240.

La dirección del gateway corresponde a 192.188.57.193 quedando por tanto disponibles para asignar a los hosts el rango de direcciones comprendido entre 192.188.57.194 y 192.188.57.206.

Luego de configurar la tarjeta de red con los parámetros expuestos (si es el caso que el servidor no posee servicio de DHCP), se visualiza que red a la que se pretende conectar aparezca en la zona de recepción de la tarjeta de red inalámbrica, como se muestra en la figura 4.8.

#### 4.2.2 COMPROBACIÓN DEL ENLACE INALÁMBRICO

Una vez realizados los pasos para la inserción del sistema de adquisición de datos a la red, se procede a verificar la comunicación entre el cliente y el servidor del sistema, es decir comprobar el enlace entre el cliente y la máquina que realiza el nexo entre el peticionario de datos y el prototipo, utilizando un comando para la verificación de la conexión entre dispositivos: PING, como se muestra en la figura 4.9.



```
Símbolo del sistema
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\administrator>ping 192.188.57.254

Haciendo ping a 192.188.57.254 con 32 bytes de datos:

Respuesta desde 192.188.57.254: bytes=32 tiempo=1ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=2ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=3ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=1ms TTL=253

Estadísticas de ping para 192.188.57.254:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 3ms, Media = 1ms

C:\Documents and Settings\administrator>_
```

Figura 4.9. Ejecución del comando PING en el cliente.

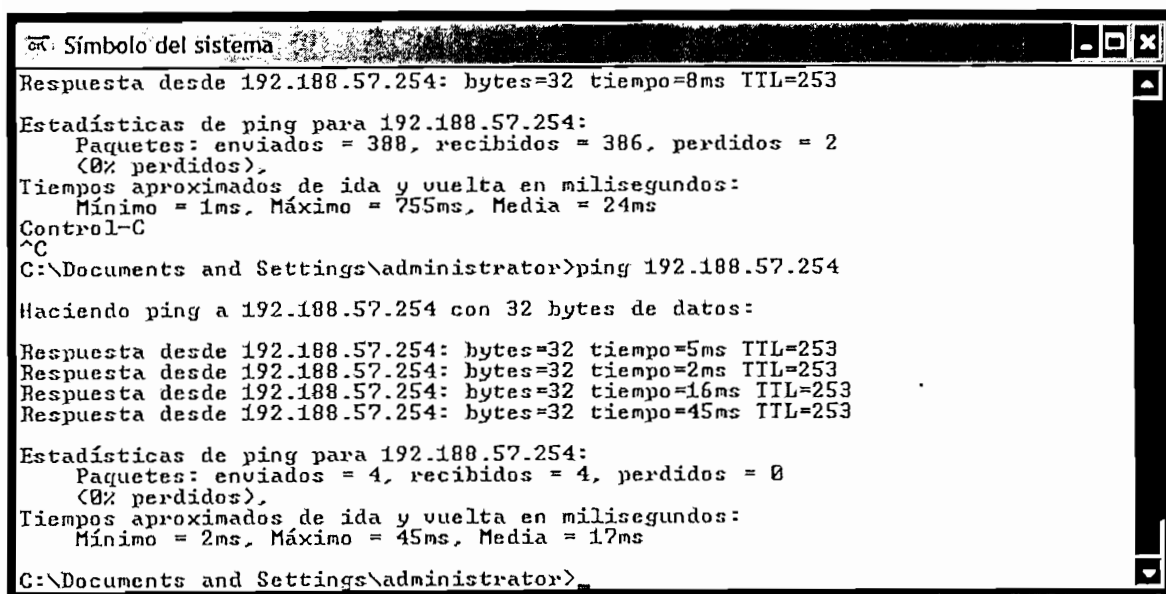
### 4.3 PRUEBAS DEL PROTOTIPO CON PC

La primera configuración utilizada para la realización de estas pruebas fue utilizando un computador como servidor de datos (Figura 4.1.a), por lo que el prototipo implementado se conecta a través del puerto serial RS-232 con el computador, y este con la red inalámbrica a través de una tarjeta de red.

#### 4.3.1 PRUEBAS DE CONEXIÓN

La verificación de la conexión cliente – servidor como se describió en apartados anteriores, se la realiza a través de la ejecución del comando PING.

Esta comprobación se la estableció a diferentes distancias y con múltiples obstáculos, llegando hasta una distancia aproximada de cien metros, observando un ligero debilitamiento de la señal, y un considerable retardo, que en las posteriores pruebas no se perciben claramente debido a que los tiempos de recepción de los paquetes de datos a través de la red está en el orden de los milisegundos y en la máxima distancia experimentada, varias decenas de milisegundos, como ejemplo se presenta la ejecución del comando PING en la distancia máxima.



```
Símbolo del sistema
Respuesta desde 192.188.57.254: bytes=32 tiempo=8ms TTL=253
Estadísticas de ping para 192.188.57.254:
  Paquetes: enviados = 388, recibidos = 386, perdidos = 2
  (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 755ms, Media = 24ms
Control-C
^C
C:\Documents and Settings\administrator>ping 192.188.57.254

Haciendo ping a 192.188.57.254 con 32 bytes de datos:

Respuesta desde 192.188.57.254: bytes=32 tiempo=5ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=2ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=16ms TTL=253
Respuesta desde 192.188.57.254: bytes=32 tiempo=45ms TTL=253

Estadísticas de ping para 192.188.57.254:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
  (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 2ms, Máximo = 45ms, Media = 17ms

C:\Documents and Settings\administrator>
```

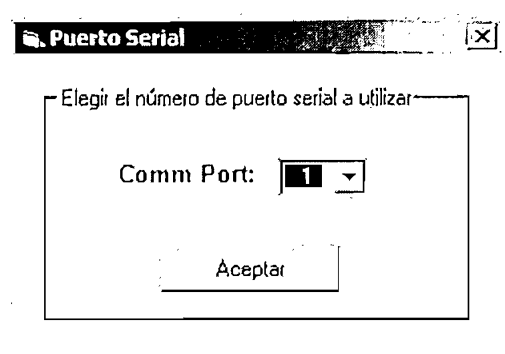
Figura 4.10. Ejecución del comando PING en el cliente.



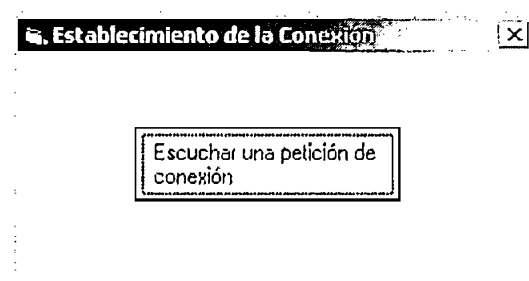
### 4.3.2 ACCESO REMOTO

El acceso remoto al equipo se realiza ejecutando software tanto en el servidor, que primero solicita elegir el puerto serial activo en el equipo y luego se mantiene esperando por una petición de conexión directa por parte de algún elemento de la red; así como en el cliente, que realiza una solicitud de conexión ingresando la dirección IP del servidor para luego poder recibir los datos adquiridos por el sistema. En los gráficos siguientes se muestra el proceso de conexión en los dos equipos.

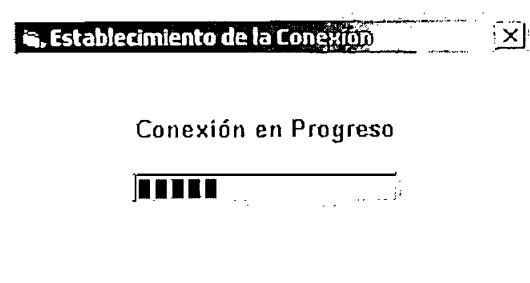
a) Elección del puerto serial activo



b) Escucha de una petición



c) Estableciendo la conexión



d) Una petición realizada exitosamente

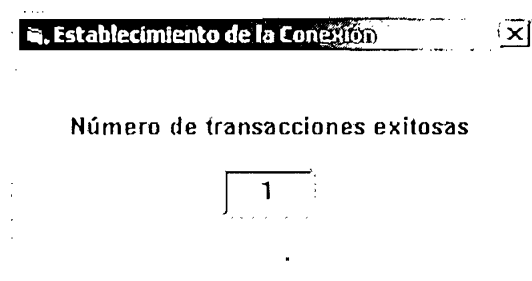
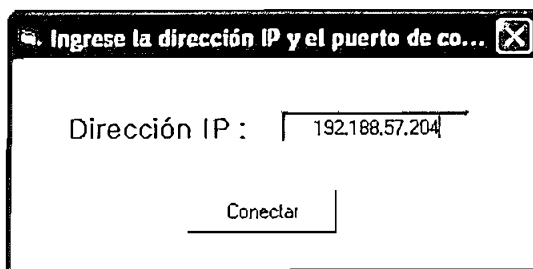


Figura 4.11. Proceso de conexión desde el servidor.

a) Ingreso de la dirección IP del servidor



b) Elección del formato de visualización de los datos adquiridos

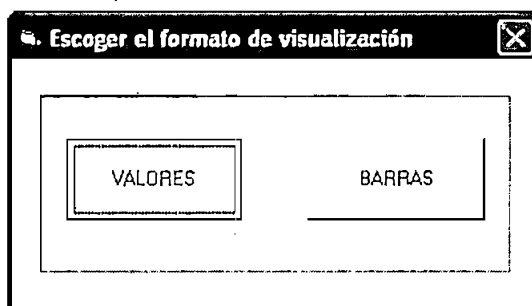


Figura 4.12. Proceso de conexión desde el cliente.

Luego del establecimiento de la conexión al escoger el tipo de visualización de los datos, se muestran en el siguiente formato:

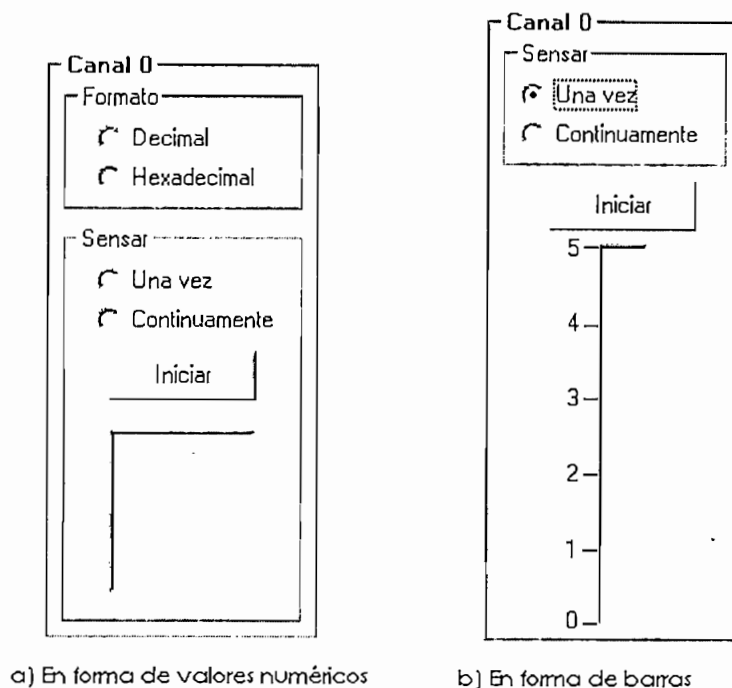


Figura 4.13. Visualización de los datos adquiridos y recibidos

### 4.3.3 PRUEBAS DE OPERACIÓN

Una vez configurada adecuadamente toda la red, además de realizadas todas las pruebas para comprobar el funcionamiento apropiado del enlace inalámbrico entre el cliente y el servidor y una vez establecida la conexión entre éstos, el cliente se procede a invocar los datos adquiridos por el sistema presionando en las pantallas de visualización el botón "Iniciar" en uno, varios o todos los ocho canales mostrados para este efecto (la capacidad en canales del sistema los da el MAX 180), acción con la que se da comienzo al proceso de envío de la petición al servidor, este a su vez solicita los datos adquiridos por el MAX180, datos que son enviados por el microcontrolador hacia el servidor y este los entrega en el formato escogido al cliente, todo en formato hexadecimal. Si el formato escogido es hexadecimal, el equipo cliente no realiza ninguna acción además de mostrar los valores recibidos, pero si el formato elegido es decimal, la conversión total se la realiza en el equipo cliente, ya que de esta manera, el

tráfico cursado por la red debido al envío de los datos adquiridos siempre será el mismo (cuatro dígitos hexadecimal por canal), evitando innecesariamente aumentar el número de bits al enviar en formato decimal con un dígito entero además de los dígitos decimales requeridos y el punto de separación entre ellos. Como se mencionó en el capítulo anterior, para mostrar la utilidad práctica de este sistema, además de mostrar la adquisición de valores de voltaje, se incluye la medición de una magnitud real: La temperatura. Para este efecto, como se observa claramente en las pantallas de visualización, se ha designado el último canal con este propósito, estando la precisión de la medición dada por la tolerancia del sensor: un grado centígrado como error extremo. Al igual que para los demás canales, la transmisión del valor adquirido en el canal designado para la temperatura, se la hace en formato hexadecimal, siendo transformado al formato de presentación (grados centígrados), en el equipo cliente, con la conversión señalada en la hoja de especificaciones del sensor.

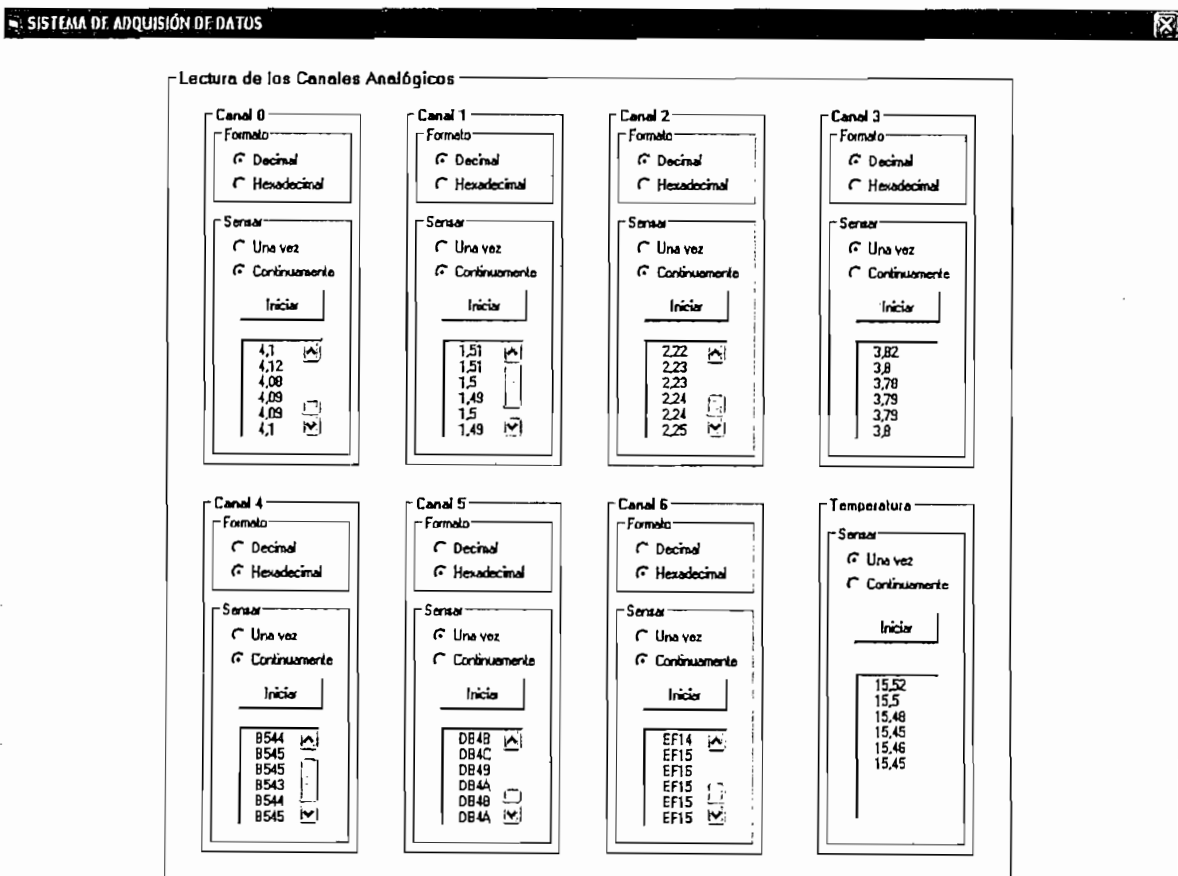


Figura 4.14. Visualización de los datos en formato de valores numéricos.

La figura 4.14. presenta la pantalla de visualización en formato de datos numéricos, con todos los canales utilizados.

En el formato gráfico o de presentación de los valores en barras, para una visualización adecuada y para no distorsionar la medida de alguna forma, se utiliza únicamente el formato decimal para la presentación de los datos adquiridos, como se muestra en la figura siguiente, el canal asignado para la medición de temperatura sigue siendo el mismo, al igual que su formato.

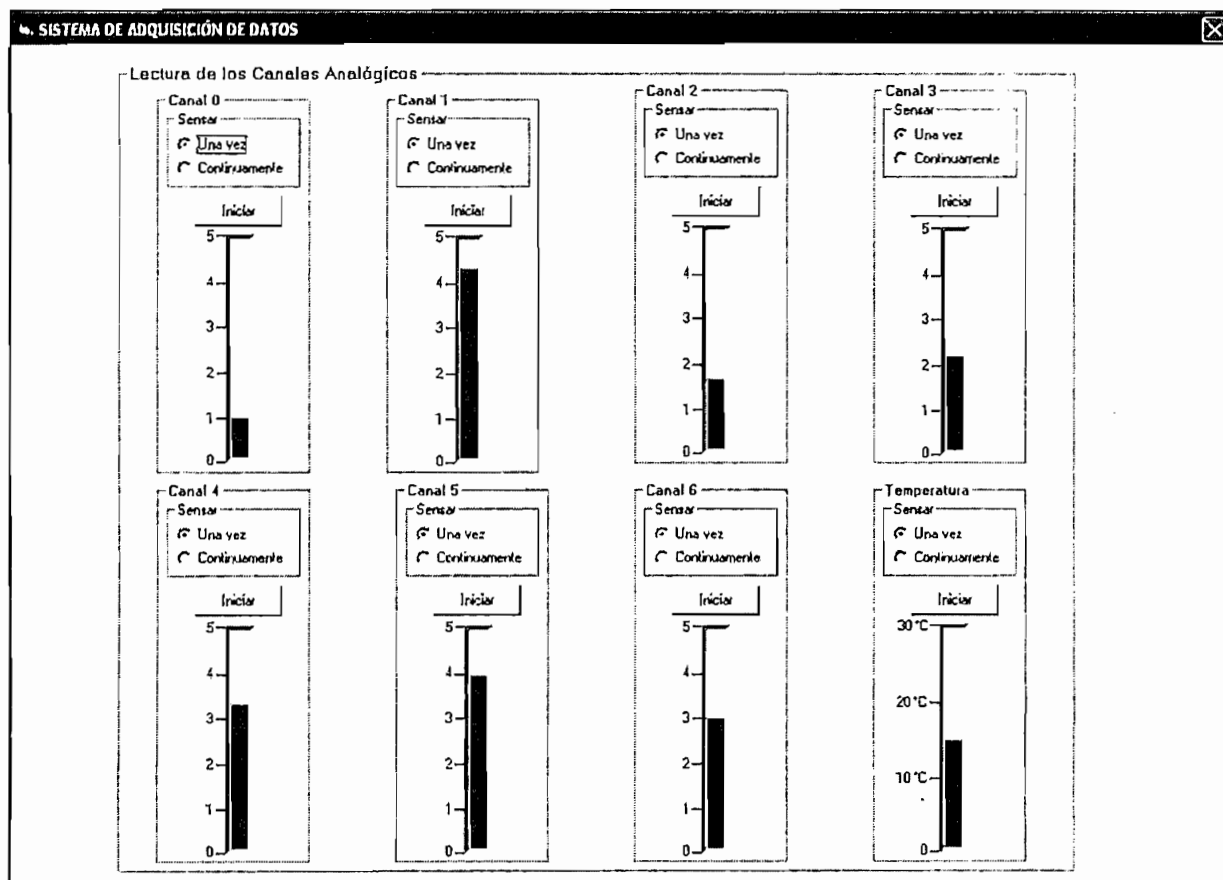


Figura 4.15. Visualización de los datos en forma de barras.

#### 4.3.4 ANÁLISIS DE RESULTADOS

Los resultados obtenidos están dentro del contexto esperados, ya que en pruebas preliminares en redes de alta velocidad alámbricas, se obtuvo la misma respuesta en la obtención de los resultados de la adquisición de datos, ya que el

retardo introducido por la red (al incrementar la distancia y también los obstáculos al alejarse del Access Point), es imperceptible al momento de presentar los datos, ya que para hacerlos moderadamente rápidos para una visualización adecuada de la variación se introduce un retardo de 300 milisegundos en cada aparición de un nuevo valor adquirido, recibido y procesado. A pesar de que este efecto no se advierte, al ejecutar el comando PING en distintos lugares y distancias del cliente respecto al servidor, se obtuvieron los tiempos de respuesta detallados a continuación:

Tiempos de respuesta aproximados (ida y vuelta) [ms]			Distancia [m]
Mínimo	Máximo	Media	
1	3	1	5
2	45	17	20
3	46	21	100

Tabla 4.1. Valores resultantes de las pruebas de conexión cliente – servidor.

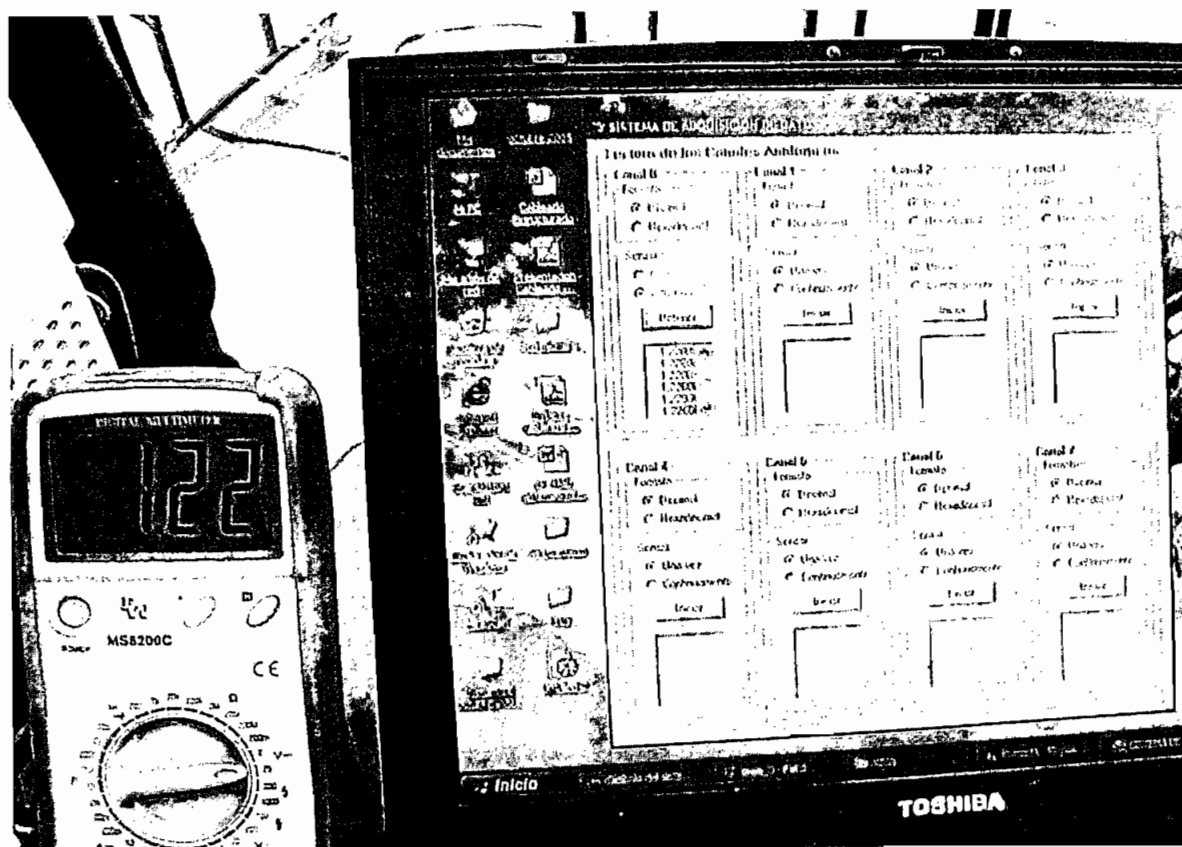


Figura 4.16. Comparación del valor adquirido y la medición de voltaje con un multímetro digital.

Además de estas pruebas, se realizó una contrastación del valor adquirido mediante el sistema a través de una red inalámbrica, ubicando al equipo portátil cliente en la cercanía del prototipo, con el valor entregado en la medición directa de la magnitud por medio de un multímetro digital, mostrándose el resultado obtenido en la figura 4.16.

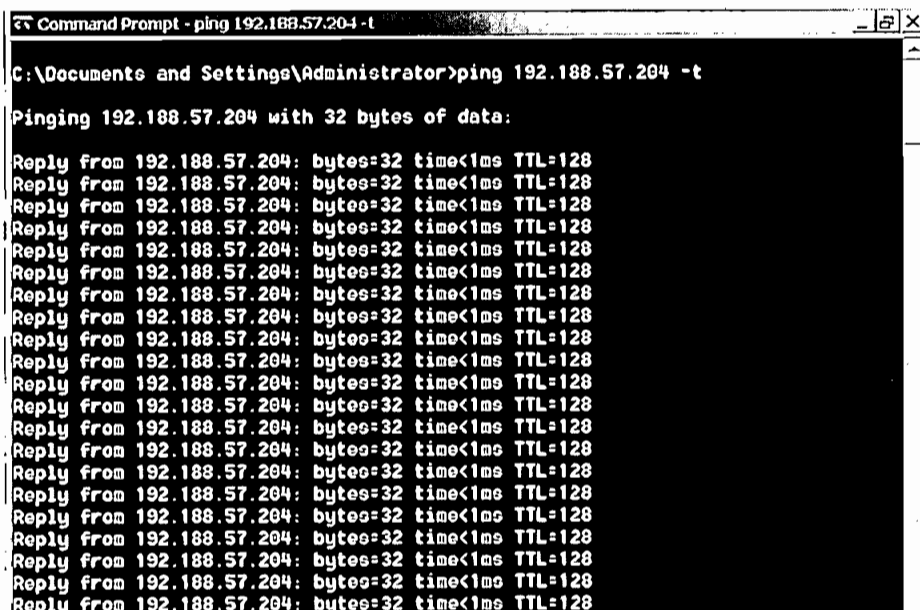
## 4.4 PRUEBAS DEL PROTOTIPO CON EZL W300

### 4.4.1 CONFIGURACIÓN

En este caso se utilizó el prototipo EZL W300, al mismo que se le asignó la dirección IP de prueba 192.188.57.204.

### 4.4.2 PRUEBAS DE CONEXIÓN

Se realizó las pruebas de conexión, de forma análoga a la descrita en el apartado anterior (con PC), obteniéndose la siguiente respuesta.



```
Command Prompt - ping 192.188.57.204 - t
C:\Documents and Settings\Administrator>ping 192.188.57.204 -t
Pinging 192.188.57.204 with 32 bytes of data:
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
Reply from 192.188.57.204: bytes=32 time<1ms TTL=128
```

Figura 4.17. Ejecución del comando PING en el cliente.

Luego se realizaron las pruebas del sistema de adquisición de datos y se obtuvieron resultados similares a los mostrados en el apartado anterior.

Es importante indicar que la diferencia con las pruebas anteriores es que en este caso el EZL W300 reemplaza al PC, permitiendo la posibilidad de movilidad e independencia.

#### 4.5 COSTOS DE LA IMPLEMENTACIÓN

Los costos que se presenta para la implementación de este proyecto fluctúan dependiendo la manera de configuración que se desee realizar, esto es si utilizamos una PC o un prototipo EZL300, cabe anotar que se trajo del extranjero el prototipo EZL300 y el MAX180, los costos de envío y el IVA de todos los elementos están ya incluidos, a continuación un análisis presupuestario:

Descripción	Cantidad	Elementos	Valor unit.	Total en \$
<b>Elemento de la Red</b>	1	EZL300	340	340
<b>Elementos de la Tarjeta de Adquisición de Datos</b>	1	MAX180	75	75
	1	RS233	6	6
	1	Micro 8052	7	7
	1	74HCT245	0,6	0,6
	1	74LS04	0,35	0,35
	2	74LS76	0,4	0,8
	2	DB9 (macho, hembra)	1	2
	1	Oscilador	5	5
	1	Pulsador	0,25	0,25
	2	LED	0,3	0,6
	6	Resistencias	0,1	0,6
	2	Potenciómetros	0,6	1,2
	4	Capacitores Electroífticos	0,15	0,6
	5	Capacitores Cerámicos	0,15	0,75
	1	Construcción	50	50
<b>Elementos del Sensor de Temperatura</b>	1	LM335	0,6	0,6
	2	Potenciómetros de Precisión	1,5	3
	1	NTE5007A	0,3	0,3
	1	2N3702	0,6	0,6
	1	Construcción	10	10
<b>Elementos de la Tarjeta de Potenciómetros</b>	8	Potenciómetros de Precisión	1,5	12
	8	Resistencias	0,1	0,8
	1	Construcción	20	20
<b>Elementos de la Fuente de Alimentación</b>	1	Bornera	0,25	0,25
	1	Puente Rectificador	0,3	0,3
	2	Fusibles	0,15	0,3
	4	Capacitores Cerámicos	0,15	0,6
	4	Capacitores Electroífticos	0,15	0,6
	1	LM7805	0,6	0,6
	1	LM7912	0,6	0,6
	1	Construcción	10	10
<b>Otros</b>	25	Conectores MOLEX	0,5	12,5
		Otros	10	10
<b>Total costos Tesis</b>				

Tabla 4.2. Costos de la Implementación

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

Este capítulo reúne las conclusiones obtenidas a lo largo de la consecución de este proyecto, referente al Sistema de Adquisición de datos y su conexión inalámbrica a la red, además de las recomendaciones que se dan a unas futuras implementaciones que tengan semejanzas respecto a este proyecto.

Con el desarrollo de este proyecto se logró implementar un sistema de adquisición de datos, que posee una interfaz de uso basada en Internet, de modo que, permita capturar señales analógicas, y que la conexión física distancia deje de ser el factor importante. Esto se logró construyendo un prototipo en el cual la parte electrónica se basa en un subsistema microprocesado, con un interfaz RS232 que está conectado a un dispositivo inalámbrico (PC con su tarjeta WLAN o módulo EZL W300), que por medio de un software de conexión, podemos lograr visualizar los datos vía inalámbrica, en principio, dentro de una red WLAN.

Al unir estos tres temas principales (RS-232, Comunicación Inalámbrica y Microcontroladores) se podrá realizar diferentes aplicaciones, que permiten transmitir información por radio frecuencia hacia uno o más microcontroladores por medio del puerto serial, es decir enviar datos desde un Sistema mediante el puerto RS-232 a un microcontrolador el cual recibirá la información y la enviará inalámbricamente hacia los diferentes puntos de una red, los cuales ejecutarán el software mencionado y visualizarán la información capturada.



La comunicación inalámbrica está extendiéndose de manera muy fuerte en el mundo, debido a que las personas tienen una mayor movilidad en sus trabajos, esto quiere decir que no solo trabajan en sus oficinas o en sus laboratorios sino además cuando viajan de una ciudad a otra o de un país a otro. Este medio de comunicación es mucho más flexible que la comunicación por medio de cables, porque puede llegar a lugares en donde los cables no pueden y a un precio que no es elevado, ya sea para personas, como para empresas u otro tipo de negocio, además permite una reestructuración mucho más ordenada y dinámica, a la hora de agregar más equipos o cambiarlos de ubicación.

Por otra parte las comunicaciones inalámbricas tienen una gran variedad en la naturaleza misma de su propagación, como por ejemplo: Radio Frecuencia, Infrarrojo, Satelital; Esto las hace mucho más interesantes y atractivas, ya que se cuenta con una gran variedad y formas de comunicarse por medio del aire, cada una de ellas, cuentan con sus propias ventajas y desventajas, así como sus propios equipos de transmisión y recepción, por lo que se debe escoger adecuadamente, según los requerimientos de cada diseño, la forma de comunicación inalámbrica.

En conclusión, se puede mencionar que la comunicación inalámbrica está marcando el mundo y que esa es la dirección hacia donde actualmente se está apuntando; además el uso de microcontroladores ha aumentado desde su creación y se ha masificado aunque las personas no se den cuenta, por tanto, éstas son dos grandes tecnologías a las cuales se les puede sacar una gran provecho, utilizándolas adecuadamente, ya que sus aplicaciones son casi ilimitadas.

Debido al gran despliegue de estándares para Comunicación Inalámbrica, de equipos portátiles cada vez más potentes, de eficientes técnicas de compresión y transmisión; el protocolo Mobile IP ha sido considerada la mejor opción para la movilidad de usuarios en Internet, no solo por presentar una solución a nivel de Red, sino también, por representar una alternativa para

interconectar las diferentes tecnologías de redes inalámbricas (IEEE 802.11a, GPRS, HiperLan, etc.), entre si y con las tradicionales redes alambradas.

La velocidad de conexión depende de la tecnología, el enlace y la red utilizados, pero ésta puede variar normalmente desde 1 hasta 54 Mbit/s, este último correspondería como conectar 1000 modems estándar juntos en una conexión. En el caso de una red local normalmente va de 11 hasta 54 Mbit/s que es similar a una red normal por cable.

El alcance de una red Inalámbrica, por ejemplo para una red local (LAN) con computadoras de oficina va de 300 metros (en espacio abierto) a 150 metros. Esto depende de las paredes, instalación del edificio y pisos, En el caso de conexiones entre oficinas ubicadas a kilómetros se pueden hacer conexiones hasta 60 Km de distancia (si ambos puntos están visibles uno del otro) y se puede utilizar repetidores.

El estándar PC Card involucra 2 tipos de tarjetas y 3 tamaños diferentes. Tipo I, Tipo II y Tipo III, hacen todos referencia al tamaño físico de la tarjeta compacta. La mayoría de PC Cards son de Tipo II, como módems, tarjetas de sonido, interfaces de red, adaptadores Compact Flash. Tipo III son exactamente el doble de altura que las de tipo II, y suelen usarse en discos duros portátiles o tarjetas especializadas como la gama Realport de Xircom. Generalmente, las tarjetas de Tipo I son menos comunes, pero caben en un slot Tipo II. Lo segundo que se debe tener en cuenta es que existen 2 tipos de especificaciones: el antiguo estándar PCMCIA de 16 bits y 5 V, y el nuevo CardBus PCI de 32 bits y 3.3 V. En otras palabras, las tarjetas PCMCIA son tarjetas ISA funcionando en un bus ISA, mientras las tarjetas CardBus son tarjetas PCI funcionando en un bus PCI. Hay que tener en cuenta un pequeño detalle: las tarjetas CardBus tienen una pequeña pestaña metálica en una de sus puntas que evita que puedan ser introducidas en un antiguo slot PCMCIA, pero las tarjetas PCMCIA sí que pueden introducirse en un slot PC Card.

El interfaz PCMCIA es el más utilizado para interconectar computadores, laptops, PAD's, Palm's, etc., a través de tarjetas de red inalámbricas, debido a su pequeño tamaño, bajo consumo de energía y a la flexibilidad que presenta en el voltaje de alimentación (La tarjeta PC Card Tipo II soportan dos valores de voltaje: voltaje normal de 5 V o con voltaje de ahorro de energía de 3,3 V). El interfaz o bus PCI típicamente utilizado para interconectar tarjetas de expansión, tarjetas de red, etc. directamente al subsistema memoria – procesador generalmente en un hardware de computadores de escritorio, aunque en ocasiones se lo puede encontrar en dispositivos portátiles.

Una vez concluido el diseño y construcción del sistema desarrollado, se aprecia la importancia y los alcances de un sistema de este tipo para beneficio de cualquier aplicación relacionada con la adquisición de datos, ya que ofrece ventajas con respecto a otros dispositivos gracias a que su diseño se basa en un microcontrolador, lo cual hace posible la modificación de las variables que genera, convirtiéndolo en un sistema adaptable para la adquisición de distintas señales analógicas.

Uno de los puntos más importantes del trabajo, son los microcontroladores. Estos dispositivos electrónicos pueden realizar una gran variedad de instrucciones, son pequeños en tamaño pero con una gran potencia a la hora de emplearlos. Como se menciona, los microcontroladores no son muy conocidos por el común de las personas o la mayoría de ellas piensan que un microcontrolador pertenece a algún robot determinado, pero los microcontroladores están siendo muy utilizados en nuestros días y dadas sus prestaciones, han tenido un despegue muy fuerte y los tenemos desde hace mucho tiempo en nuestros hogares, como por ejemplo en la lavadora automática, en una calculadora, en un microondas, etc. se podría decir que lo que tenemos en nuestros hogares son robots que realizan tareas para nosotros, ya que fueron programados para realizar un trabajo específico.

Existen una gran cantidad de microcontroladores, en donde cada uno de ellos posee ventajas y desventajas, ya sea de su arquitectura física, su circuitería

como también de su precio o nivel de complejidad para utilizarlos. Todos ellos poseen una gran potencialidad, es decir se puede realizar una infinidad de labores con un circuito tan pequeño (Por ejemplo el chip de una calculadora, puede sumar, restar, dividir, etc. a una velocidad extremadamente alta). Para este trabajo se eligió el microcontrolador 8051, debido a que se tiene un conocimiento previo, es de fácil acceso y bajo costo.

El trabajar con microcontroladores es una tarea muy atractiva ya que se pueden realizar múltiples diferentes tipos de aplicaciones, lo primordial de este proyecto es que se pueda seguir desarrollando o fomentar a los alumnos para que sigan desarrollando y poder realizar mejoras para una mayor utilidad.

Al requerir la conversión de varias señales analógicas a formato digital, el circuito integrado MAX 180 presenta una relativa ventaja respecto a conversores analógico-digitales convencionales, respecto a: optimización de espacio físico ocupado, energía de consumo; alta resolución: 12 bits, mínima complejidad de conexión y procesamiento de la información de control enviada y los datos adquiridos; permitiendo, con sencillos cambios en la programación del microcontrolador que gobierna su funcionamiento en las etapas de adquisición de datos, manejar señales unipolares (8), bipolares (4) y/o diferenciales (3), tomando las consideraciones referidas por el fabricante.

Los circuitos electrónicos se componen de subsistemas o circuitos dedicados y se piensa en términos de ENTRADA, PROCESO, SALIDA. Entre estos sistemas se transfieren las distintas señales. Estas señales se transfieren casi siempre como voltajes que cambian. Esto hace inevitable que los circuitos electrónicos incluyan los divisores de voltaje similares a los estudiados, como parte integral de su estructura.

Una desventaja del MAX 180 frente a conversores analógico-digitales tradicionales radica en la falta de flexibilidad de los límites de los voltajes en las entradas analógicas, permitiendo únicamente dos configuraciones: unipolar, con

señales que pueden ir de 0 V a 5V, y bipolar, con señales que van de -2,5V a 2,5V (en el formato de entradas diferenciales, el voltaje resultante de la comparación de las dos señales de entrada no puede exceder los 5V); siendo necesaria una circuitería externa para adaptar las señales a adquirir a las entradas del MAX 180 además de pequeñas modificaciones en el tratamiento de los datos obtenidos a través de la programación del microcontrolador que maneja la presentación final de éstos, para poder trabajar con señales de entrada que fluctúen entre valores límite diferentes a los descritos (lógicamente superiores).

Otra desventaja del MAX 180 respecto a conversores analógico-digitales habitualmente utilizados, se presenta en su alimentación. Este circuito integrado requiere de dos voltajes distintos para su funcionamiento: el de alimentación, cuyo valor debe ser de 5V; y voltaje de referencia negativo que puede variar entre -12V y -15V, aunque no se prevea procesar señales analógicas de entrada negativas.

Un AP es una tarjeta wireless que actúa a modo de Hub, es decir como concentrador de señal, es decir, en una red wireless con AP todos los clientes comunican directamente con el AP y este se encarga de distribuir la señal. Aparte, los AP pueden enlazarse entre sí para configurar una red más extensa al igual que ocurre con un sistema de Hub en cascada. Además, según la marca y modelo del AP, éste tendrá implementado opciones de seguridad, filtrado, router, etc.

El proceso que sigue el Cliente es identificar los puntos de acceso y las redes disponibles. Este proceso se lleva a cabo mediante el control de las tramas de señalización procedentes de los puntos de acceso que se anuncian a sí mismos o mediante el sondeo activo de una red específica con tramas de sondeo.

El Alcance de la red está limitado por las capacidades del punto de acceso, este se encarga de coordinar la transmisión y recepción de múltiples dispositivos inalámbricos dentro de una extensión específica; la extensión y el número de dispositivos dependen del estándar de conexión inalámbrica que se utilice y del

producto. En la modalidad de infraestructura, puede haber varios puntos de acceso para dar más cobertura a una zona grande o un único punto de acceso para una zona pequeña, ya sea un hogar o un edificio pequeño

Un factor importante que el usuario o una estación debe tener en cuenta cuando se desplaza de un punto de acceso a otro punto de acceso, es mantener una asociación entre la tarjeta NIC y un punto de acceso para poder mantener la conectividad de la red. Esto puede plantear un problema especialmente complicado si la red es grande y el usuario debe cruzar límites de subredes o dominios de control administrativo.

Si el usuario cruza un límite de subred, la dirección IP asignada originalmente a la estación puede dejar de ser adecuada para la nueva subred. Si la transición supone cruzar dominios administrativos, es posible que la estación ya no tenga permiso de acceso a la red en el nuevo dominio basándose en sus credenciales.

La sincronización entre las estaciones de la red se controla mediante las tramas de señalización periódicas enviadas por el punto de acceso. Estas tramas contienen el valor de reloj del punto de acceso en el momento de la transmisión, por lo que sirve para comprobar la evolución en la estación receptora. La sincronización es necesaria por varias razones relacionadas con los protocolos y esquemas de modulación de las conexiones inalámbricas.

La limitación de usuarios depende del tipo de servidor que se use, para este caso la disponibilidad de usuarios es uno, debido a que el software implementado tiene la herramienta Winsock, la cual permite tener un enlace cliente/servidor en una modalidad punto a punto (uniusuario)

La velocidad de transmisión depende de la distancia del enlace entre el servidor y el cliente, del tipo de tecnología de red WLAN, y de los obstáculos o niveles de interferencia que existan, a mayor retardo menor velocidad, esto básicamente se puede determinar observando el tiempo de respuesta de ida y vuelta de la petición de conexión entre el cliente /servidor mediante en comando

Pin, estos tiempos varían según como se aleje o acerque el portátil o dispositivo inteligente.

Finalmente, cabe señalar que en revistas de actualidad (diciembre 2005 y enero 2006) se presentan soluciones integradas similares al presente proyecto, ligeramente más económicas y compactas, pero de forma expresa señalan que el tema abordado es de actualidad, ya que en laboratorios extranjeros han estado trabajando en estos dispositivos.

## REFERENCIA BIBLIOGRÁFICAS

- [1] Arturo Gutiérrez Landa.  
[http://www.batihuevo.tripod.com/III\\_ex\\_parcial.doc](http://www.batihuevo.tripod.com/III_ex_parcial.doc).  
 REDES DE ÁREA PERSONAL INALÁMBRICA, WPANS Y APLICACIONES.  
 CARACTERÍSTICAS DE LOS SISTEMAS BLUETOOTH (NORMAS IEEE 802.15) Y  
 HOME RF (proyecto cancelado en 2002).
- [2] Grup Gíreles – UPF, Estudiants Enginyeria Telemàtica, Projecte Mobicat.  
[http://wupf.upf.edu/documents/visualizar/Estudio\\_tecnol\\_inal.htm](http://wupf.upf.edu/documents/visualizar/Estudio_tecnol_inal.htm).  
 ESTUDIO DE TECNOLOGÍAS INALÁMBRICAS.
- [3] Luciano Moreno, Integrante del departamento de diseño web de BJS Software.  
[http://www.htmlweb.net/redes/tcp\\_ip/capa\\_2/acceso\\_red\\_1.html](http://www.htmlweb.net/redes/tcp_ip/capa_2/acceso_red_1.html).  
 LA CAPA DE ENLACE DE DATOS (I)
- [4] Ing. Pepino M. António Prazer Estudiante de la Maestría en Telemática del Instituto Superior Politécnico José Antonio Echeverría.  
<http://www.cujae.edu.cu/revistas/telematica/Articulos/106.htm>.  
 UNA SOLUCIÓN PARA EL ENRUTAMIENTO TRIANGULAR DEL MOBILE IPV4 (MIP).  
 Cuba  
 Mayo 2004.
- [5] Departament d'Enginyeria Química, Universitat Autònoma de Barcelona.  
<http://eq3.uab.es/personal/baeza/comunicaciones/comunica.htm> ,  
 COMUNICACIONES.
- [6] Jordi Mayné, Ingeniero de Aplicaciones, SILICA.  
[http://www.bairesrobotics.com.ar/data/Sistemas\\_Comunicaciones\\_r35\\_Silica.pdf](http://www.bairesrobotics.com.ar/data/Sistemas_Comunicaciones_r35_Silica.pdf)  
 SISTEMAS DE COMUNICACIONES.
- [7] <http://isc.tripod.com.mx/ndis.htm>,  
 NORMAS DE INTERCONEXIÓN EN SERIE  
 Tripod, c2000, DE  
 25 de abril, 2003.
- [8] Walter Navarrete, Departamento de Operaciones Asociación Infocentros,  
<http://www.monografias.com/trabajos11/basdat/basdat.shtml>,  
 BASES DE DATOS  
 El Salvador.



- [9] Ing. Huarache Francia Oswaldo  
<http://cpys.iespana.es/hardware/7.pdf>.  
 BUS: ESTRUCTURAS DE INTERCONEXIÓN DE UN SISTEMA COMPUTACIONAL
- [10] Alberto Enciso Lizárraga, Ingeniero Electrónico en el Itmar Mazatlán.  
<http://betoe.imgoingtobed.com/informatica/PCI1.doc>.  
 BUS LOCAL PCI (PERIPHERAL COMPONENT INTERCONNECT)  
 México.
- [11] <http://www.mailxmail.com/curso/informatica/arquitecturaordenadores/capitulo12.htm>,  
 ARQUITECTURA DE ORDENADORES  
 MAILXMAIL.
- [12] [www.meritec.com/Pages/pdf/pci.pdf](http://www.meritec.com/Pages/pdf/pci.pdf),  
 PCI CONNECTORS AND CARD EDGE CONNECTORS  
 MERITEC  
 Painesville, Ohio, EE.UU.
- [13] Alberto Enciso Lizárraga, Ingeniero Electrónico en el Itmar Mazatlán.  
<http://betoe.imgoingtobed.com/informatica/PCI1.doc>,  
 BUS LOCAL PCI (PERIPHERAL COMPONENT INTERCONNECT).  
 México.
- [14] 1999, PCMCIA/JEIDA STANDARD;  
 PCMCIA, San Jose, California, EE.UU., JEIDA, Tokio, Japón.
- [15] 1999, PCMCIA/JEIDA STANDARD;  
 PCMCIA, San Jose, California, EE.UU., JEIDA, Tokio, Japón.
- [16] 1999, PCMCIA/JEIDA STANDARD;  
 PCMCIA, San Jose, California, EE.UU., JEIDA, Tokio, Japón.
- [17] Vega Martínez, Aurelio, Docente Titular del Departamento de Ingeniería Electrónica y Automática de la Universidad de las Palmas de la Gran Canaria.  
[http://diea.ulpgc.es/users/aurelio/int\\_equipos/trab9899/pcmcia\\_2/](http://diea.ulpgc.es/users/aurelio/int_equipos/trab9899/pcmcia_2/),  
 PCMCIA (2)  
 España.

- [18] 1999, PCMCIA/JEIDA STANDARD.  
PCMCIA, San Jose, California, EE.UU., JEIDA, Tokio, Japón.
- [19] <http://maxim-ic.com>  
COMPLETE 8 CHANNEL, 12-BIT DATA-ACQUISITION SYSTEMS: MAX180/MAX181  
MAXIM Integrated Products  
Sunnyvale, California, EE.UU.
- [20] <http://maxim-ic.com>  
COMPLETE 8 CHANNEL, 12-BIT DATA-ACQUISITION SYSTEMS: MAX180/MAX181  
MAXIM Integrated Products  
Sunnyvale, California, EE.UU.
- [21] <http://onsemi.com>,  
DUAL JK FLIP-FLOP WITH SET AND CLEAR: SN74LS76A.  
ON SEMICONDUCTOR  
Tokio Japón.
- [22] [www.fairchildsemi.com](http://www.fairchildsemi.com)  
3-STATE OCTAL BUS TRANSCEIVER: DM74LS245.  
FAIRCHILD SEMICONDUCTOR Corporation  
South Portland, U.S.A. February 2000.
- [23] <http://www.national.com>  
LM78XX SERIES VOLTAGE REGULATORS  
National Semiconductor Americas Customer Support Center.
- [24] <http://www.national.com>,  
LM79XX SERIES 3-TERMINAL NEGATIVE REGULATORS  
National Semiconductor Americas Customer Support Center.
- [25] <http://es.encarta.msn.com>.  
TEMPERATURA  
Enciclopedia Microsoft® Encarta® Online 2005.
- [26] Ing. Juan Gilberto Mateos Suárez, Docente de la Universidad de Guadalajara.  
<http://proton.ucting.udg.mx/temas/control/memo/MEMO.html>  
SENSORES DE TEMPERATURA  
México.

- [27] José Villasuso Gato, estudiante del IES Monelos de La Coruña.  
[http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicalinteractiva/sacaleE\\_M2/termoelectricidad/Termoelectricidad.htm](http://teleformacion.edu.aytolacoruna.es/FISICA/document/fisicalinteractiva/sacaleE_M2/termoelectricidad/Termoelectricidad.htm)  
 TERMOPAR O PAR TERMOELÉCTRICO.  
 España
- [28] <http://www.national.com>,  
 PRECISION TEMPERATURE SENSORS: LM135/LM235/LM335,  
 LM135A/LM235A/LM335A,  
 National Semiconductor Americas Customer Support Center.
- [29] <http://www.national.com>,  
 PRECISION TEMPERATURE SENSORS: LM135/LM235/LM335,  
 LM135A/LM235A/LM335A,  
 National Semiconductor Americas Customer Support Center.

Bastidas Galarza Alfonso

**“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO PARA MEDICIÓN DE TEMPERATURA UTILIZANDO TERMOCUPLAS”**

Proyecto de Titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Escuela Politécnica Nacional.

Quito, Ecuador.

Febrero 2002.

Rivadeneira Hinojosa Nuria Karina

**“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE UN SERVIDOR WEB BASADO EN MICROCONTROLADOR 8x51”**

Proyecto de Titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Escuela Politécnica Nacional.

Quito, Ecuador.

Febrero 2002.

**8-BIT MICROCONTROLLES WITH 8K BYTES FLASH.**

MAXIM Corporation.

San Jose, California, EE.UU.

1999.

**ezTCP/LAN (EZL – 300W) USER'S MANUAL**

Sollae Systems Co., Ltd.

<http://www.ezTCP.com>

Mano, M. Morris.

**ARQUITECTURA DE COMPUTADORES**

Prentice Hall.

México, México.

1983

Aguilar Carlos, Arias Paúl

**“TECNOLOGIAS DE ACCESO INALAMBRICO Y DISEÑO DE LA RED INALÁMBRICA FIJA PARA EL AREA DE QUITO CENTRO”**

Proyecto de Titulación, Facultad de Ingeniería Electrónica

Escuela Politécnica del Ejército

Junio 2003

## ANEXOS

## **ANEXO 1**

**CÓDIGO FUENTE DE LOS PROGRAMAS  
DESARROLLADOS**



```

Unload Presentación      'Y SE CIERRAN LAS PANTALLAS DE 'PRESENTACIÓN
Unload Me                'Y LA PANTALLA ACTUAL

End Sub

Private Sub Command2_Click()

Barras.Show              'AL PRESIONAR EL BOTON "BARRAS", SE MUESTRA
                          'AUTOMÁTICAMENTE LA PANTALLA "barras"

Unload Presentación      'Y SE CIERRAN LAS PANTALLAS DE PRESENTACIÓN
Unload Me                'Y LA PANTALLA ACTUAL

End Sub

'FORMULARIO QUE MUESTRA LOS DATOS ADQUIRIDOS EN FORMA DE 'VALORES
NUMERICOS

'DECLARACIÓN DE VARIABLES

Option Explicit
Public dirIP, Npuerto As String
Dim det1, u1, det2, u2, det3, u3, det4, u4, det5, u5, det6, u6, det7, u7, det8, u8 As Boolean
Dim Dec(31), i As Integer
Dim enviar, Recep, Aux(31), aux1(8) As String
Dim deci1(8), auxtemp As Single

'INICIALIZACIÓN DE VARIABLES

Private Sub Form_Load()
Inicializar              'SE INVOCA AL EVENTO "Inicializar" PARA UBICAR LOS VALORES
                          'INICIALES A TODAS LAS VARIABLES NECESARIAS.

Winsock1.RemoteHost = IP.dirIP      'SE UBICA LA DIRECCIÓN DEL SERVIDOR INGRESADA
                                     'EN EL CAMPO "RemoteHost"

Winsock1.LocalPort = 888            'SE ASIGNA EL NUMERO DE PUERTO LOCAL

Winsock1.RemotePort = 888          'SE ASIGNA EL NUMERO DE PUERTO REMOTO

Winsock1.Connect                'PETICION DE CONEXION CON EL SERVIDOR REMOTO
    i = 0
End Sub

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 0
Private Sub inicia1_Click()

If Timer1.Enabled = False Then      'SI EL TIMER QUE REALIZA LAS PETICIONES CADA 300
    Timer1.Enabled = True           'MILISEGUNDOS ESTA INACTIVO
    'SE ENCIENDE

End If

u1 = False                          'VARIABLE QUE INDICA SI SE PETICIONÓ SENSAR CONTINUAMENTE EL
det1 = False                          'CANAL (u1 = False→ SE PIDIO SENSAR CONTINUAMENTE)

If Cont1 = True Then                'SI SE ESCOGIO SENSAR CONTINUAMENTE
    If inicia1.Caption = "Iniciar" Then 'Y LA LEYENDA DEL BOTON DICE "Iniciar"
        Cont1.Enabled = False         'VARIABLE QUE INDICA QUE SE PRESIONO "Iniciar"
        inicia1.Caption = "Detener"   'SE UBICA COMO NUEVA LEYENDA "Detener"
    End If
End If

```



```

    det1 = True                'VARIABLE PARA QUE EL TIMER REALICE PETICIONES
                              'CONTINUAS
End If
Else
u1 = True                    'SI NO SE PIDIO SENSAR CONTINUAMENTE EL CANAL, ESTA VARIABLE
                              'SE UBICA EN TRUE
End If

If inicia1.Caption = "Detener" And det1 = False Then    'SI SE PREISIONÓ EL BOTON "Detener"
                                                         'LA LEYENDA DEL BOTÓN
    inicia1.Caption = "Iniciar"                        'CAMBIA A "Iniciar"

    Cont1.Enabled = True    'VALOR DE LA VARIABLE QUE INDICA QUE SE DEBEN DETENER
                              'LAS PETICIONES

'SI SE PRESIONÓ INICIAR PERO UNA SOLA VEZ, SE OBSERVA TODOS LOS CANALES: SI
'NINGUNO PETICIONÓ SENSAR SU CANAL SE DESHABILITA 'EL TIMER PARA LA CONVERSIÓN Y
'SE DETIENE EL PROCESO A LA ESPERA DE UNA TECLA:

    If det2 = False And det3 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub

```

'A continuación se muestra la codificación utilizada en los demás canales, pero la documentación se suprime ya que es la misma:

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 1

```

Private Sub inicia2_Click()

If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If

u2 = False
det2 = False

If Cont2.Value = True Then

    If inicia2.Caption = "Iniciar" Then
        Cont2.Enabled = False
        inicia2.Caption = "Detener"
        det2 = True
    End If

Else
u2 = True
End If

If inicia2.Caption = "Detener" And det2 = False Then
    inicia2.Caption = "Iniciar"
    Cont2.Enabled = True
    If det1 = False And det3 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If

```

```
Else
End If
End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 2

```
Private Sub inicia3_Click()
```

```
If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If
```

```
u3 = False
det3 = False
```

```
If Cont3.Value = True Then
    If Inicia3.Caption = "Iniciar" Then
        Cont3.Enabled = False
        Inicia3.Caption = "Detener"
        det3 = True
    End If
```

```
Else
u3 = True
End If
```

```
If Inicia3.Caption = "Detener" And det3 = False Then
    Inicia3.Caption = "Iniciar"
    Cont3.Enabled = True
    If det1 = False And det2 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 3

```
Private Sub inicia4_Click()
```

```
If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If
```

```
u4 = False
det4 = False
```

```
If Cont4.Value = True Then
    If Inicia4.Caption = "Iniciar" Then
        Cont4.Enabled = False
        Inicia4.Caption = "Detener"
        det4 = True
    End If
```

```
Else
u4 = True
End If
```

```
If Inicia4.Caption = "Detener" And det4 = False Then
```

```

    Inicia4.Caption = "Iniciar"
    Cont4.Enabled = True
    If det1 = False And det2 = False And det3 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 4

```
Private Sub inicia5_Click()
```

```

If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If

```

```

u5 = False
det5 = False

```

```

If Cont5.Value = True Then
    If Inicia5.Caption = "Iniciar" Then
        Cont5.Enabled = False
        Inicia5.Caption = "Detener"
        det5 = True
    End If

```

```

Else
u5 = True
End If

```

```

If Inicia5.Caption = "Detener" And det5 = False Then
    Inicia5.Caption = "Iniciar"
    Cont5.Enabled = True
    If det1 = False And det2 = False And det3 = False And det4 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 5

```
Private Sub inicia6_Click()
```

```

If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If

```

```

u6 = False
det6 = False

```

```

If Cont6.Value = True Then
    If Inicia6.Caption = "Iniciar" Then
        Cont6.Enabled = False
        Inicia6.Caption = "Detener"
        det6 = True
    End If

```

```

Else
u6 = True
End If

If Inicia6.Caption = "Detener" And det6 = False Then
  Inicia6.Caption = "Iniciar"
  Cont6.Enabled = True
  If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det7 = False And det8 = False Then
    Timer1.Enabled = False
  End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 6

```
Private Sub Inicia7_Click()
```

```

If Timer1.Enabled = False Then
  Timer1.Enabled = True
End If

```

```

u7 = False
det7 = False

```

```

If Cont7.Value = True Then
  If Inicia7.Caption = "Iniciar" Then
    Cont7.Enabled = False
    Inicia7.Caption = "Detener"
    det7 = True
  End If

```

```

Else
u7 = True
End If

```

```

If Inicia7.Caption = "Detener" And det7 = False Then
  Inicia7.Caption = "Iniciar"
  Cont7.Enabled = True
  If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det6 = False And det8 = False Then
    Timer1.Enabled = False
  End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 7

```
Private Sub Inicia8_Click()
```

```

If Timer1.Enabled = False Then
  Timer1.Enabled = True
End If

```

```

u8 = False
det8 = False

```

```

If Cont8.Value = True Then
  If Inicia8.Caption = "Iniciar" Then
    Cont8.Enabled = False
    Inicia8.Caption = "Detener"
    det8 = True
  End If
Else
  u8 = True
End If

If Inicia8.Caption = "Detener" And det8 = False Then
  Inicia8.Caption = "Iniciar"
  Cont8.Enabled = True
  If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det6 = False And det7 = False Then
    Timer1.Enabled = False
  End If
Else
  End If
End Sub

'ACCIONES A EJECUTAR CADA VEZ QUE EL TIMER ESTÉ HABILITADO Y
'TRANSCURRAN 300 MILISEGUNDOS

Private Sub Timer1_Timer()
  enviar = "0"
  Winsock1.SendData enviar
  'ALMACENAR EN ESTA VARIABLE UN VALOR CUALQUIERA,
  'PARA ENVIARLO A TRAVÉS DEL
  'PUERTO DE RED, ACCIÓN QUE INDICA QUE EL CLIENTE ESTÁ
  'ENVIANDO UNA SOLICITUD POR UN DATO ADQUIRIDO
  REMOTAMENTE.

'COMPROBANDO SI SE HA PRESIONADO DETENER EN TODOS LOS 'CANALES QUE ESTABAN
SENSANDO CONTINUAMENTE:

If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And det6
= False And det7 = False And det8 = False Then

'COMPROBANDO SI SE PRESIONARON ÚNICAMENTE CONVERSIONES 'SIMPLES:

If u1 = True Or u2 = True Or u3 = True Or u4 = True Or u5 = True Or u6 = True Or u7 = True Or
u8 = True Then
  Timer1.Enabled = False
  'SI SE CUMPLEN ESTAS CONDICIONES, SE
  'DESHABILITA EL TIMER

  End If
End If
End Sub

'PROCEDIMIENTO A REALIZAR CUANDO SE RECIBA LOS DATOS PETICIONADOS 'A
TRAVÉS DEL PUERTO SERIAL.

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)

  Winsock1.GetData Recep
  procesamiento
  envío
  'SE ALMACENA LO RECIBIDO EN LA 'VARIABLE "Recep"
  'SE INVOCA A LA SUBROUTINA "procesamiento"
  'SE INVOCA A LA SUBROUTINA "envío"

End Sub

'ACCIONES REALIZADAS EN LA SUBROUTINA PROCESAMIENTO

Private Sub procesamiento()

```

'LA INFORMACIÓN RECIBIDA ES UN TREN DE 32 CARACTERES HEXADECIMALES QUE PARA 'MAYOR FACILIDAD SE ALMACENAN UNO POR UNO EN UN VECTOR DE TAMAÑO 32, PARA 'LUEGO UTILIZARLO EN 'LA CONVERSIÓN A DECIMAL:

```
For i = 1 To 32 Step 1
  Aux(i - 1) = Mid(Recep, i, 1)
Next
```

'EL TREN DE CARACTERES TAMBIÉN ES ALMACENADO CADA 4 DE ELLOS EN UN VECTOR DE 'TAMAÑO 8 (4 CARACTERES POR CANAL) PARA UTILIZARLOS SI SE REQUIEREN DATOS EN 'FORMATO HEXADECIMAL

```
For i = 0 To 32 Step 4
  aux1(i / 4) = Mid(Recep, i + 1, 4)
Next
```

'SE COMPRUEBA QUE LOS VALORES DEL VECTOR DE TAMAÑO 32 ESTÉN 'DENTRO DEL RANGO '0 - 9, CASO CONTRARIO SE TRANSFORMAN A SU 'EQUIVALENTE DECIMAL (A→10, B→11, C→12, 'D→13, E→14, F→15)

```
For i = 0 To 31 Step 1
  If Aux(i) = "A" Then
    Dec(i) = 10
  ElseIf Aux(i) = "B" Then
    Dec(i) = 11
  ElseIf Aux(i) = "C" Then
    Dec(i) = 12
  ElseIf Aux(i) = "D" Then
    Dec(i) = 13
  ElseIf Aux(i) = "E" Then
    Dec(i) = 14
  ElseIf Aux(i) = "F" Then
    Dec(i) = 15
  Else
    Dec(i) = Val(Aux(i))
  End If
Next
```

'SE TRANSFORMA A DECIMAL EL VECTOR TAMAÑO 8: MULTIPLICANDO EL VALOR MÁS 'SIGNIFICATIVO POR  $16^3$  Y SUMANDO AL SIGUIENTE MULTIPLICADO POR  $16^2$  Y AL SIGUIENTE 'MULTIPLICADO POR 16 Y 'SUMADO AL MENOS SIGNIFICATIVO, LUEGO ESTE RESULTADO SE 'DIVIDE POR 819.1632, QUE ES EL VALOR EQUIVAMENTE DE LA CONVERSIÓN OFFF 'HEXADECIMAL A 5.000 DECIMAL Y ESTE VALOR SE LO REDONDEA CON TRES CIFRAS 'DECIMALES

```
For i = 0 To 30 Step 4
  deci(i / 4) = Round((Dec(i) * (16 * 16 * 16) + Dec(i + 1) * (16 * 16) + Dec(i + 2) * 16 + Dec(i + 3)) / 819.1638, 3)
Next
```

End Sub

'PROCEDIMIENTO A REALIZAR POR LA SUBROUTINA ENVIO:

Private Sub envio()

```
If u1 = True Or det1 = True Then
  'COMPROBANDO SI EL CANAL 0 REQUIERE DATOS
  'ADQUIRIDOS
  If Hex1.Value = True Then
    'COMPROBANDO SI SE REQUIEREN DATOS EN
    'FORMATO DECIMAL
    datos1.AddItem " " & (aux1(0))
    'MOSTRAR LOS DATOS EN EL CASILLERO
    'CORRESPONDIENTE AL CANAL 0
```

```

datos1.TopIndex = datos1.ListCount - 1 'ARREGLO PARA QUE LOS ÚLTIMOS DATOS
'ADQUIRIDOS 'SE MUESTREN EN LA PANTALLA
End If

```

```

If Dec1.Value = True Then 'COMPROBANDO SI SE REQUIEREN DATOS EN
'FORMATO DECIMAL
datos1.AddItem " " & (deci1(0)) 'MOSTRAR LOS DATOS EN EL CASILLERO
'CORRESPONDIENTE AL CANAL 0

```

```

datos1.TopIndex = datos1.ListCount - 1 'ARREGLO PARA QUE LOS ÚLTIMOS DATOS
ADQUIRIDOS SE MUESTREN EN LA PANTALLA
End If

```

'COMPROBANDO SI LA PETICIÓN FUE SOLO POR UNA CONVERSIÓN 'SIMPLE, SIENDO ESTE EL CASO SE CONSIFURA EN "False" LA VARIABLE ' "u1" INDICANDO QUE SE ATENDIÓ LA PETICIÓN.

```

If u1 = True Then
u1 = False
End If
End If

```

'EL MISMO PROCESO SE SIGUEN EN TODOS LOS CANALES, RAZÓN POR LA 'QUE SOLO SE MUESTRA LA CODIFICACIÓN:

```

If u2 = True Or det2 = True Then
If hex2.Value = True Then
datos2.AddItem " " & (aux1(1))
datos2.TopIndex = datos2.ListCount - 1
End If
If dec2.Value = True Then
datos2.AddItem " " & (deci1(1))
datos2.TopIndex = datos2.ListCount - 1
End If
If u2 = True Then
u2 = False
End If
End If

```

```

If u3 = True Or det3 = True Then
If Hex3.Value = True Then
datos3.AddItem " " & (aux1(2))
datos3.TopIndex = datos3.ListCount - 1
End If
If Dec3.Value = True Then
datos3.AddItem " " & (deci1(2))
datos3.TopIndex = datos3.ListCount - 1
End If
If u3 = True Then
u3 = False
End If
End If

```

```

If u4 = True Or det4 = True Then
If Hex4.Value = True Then
Datos4.AddItem " " & (aux1(3))
Datos4.TopIndex = Datos4.ListCount - 1
End If
If Dec4.Value = True Then
Datos4.AddItem " " & (deci1(3))
Datos4.TopIndex = Datos4.ListCount - 1

```

```
End If
If u4 = True Then
    u4 = False
End If
End If

If u5 = True Or det5 = True Then
    If Hex5.Value = True Then
        Datos5.AddItem " " & (aux1(4))
        Datos5.TopIndex = Datos5.ListCount - 1
    End If
    If Dec5.Value = True Then
        Datos5.AddItem " " & (deci1(4))
        Datos5.TopIndex = Datos5.ListCount - 1
    End If
    If u5 = True Then
        u5 = False
    End If
End If

If u6 = True Or det6 = True Then
    If Hex6.Value = True Then
        Datos6.AddItem " " & (aux1(5))
        Datos6.TopIndex = Datos6.ListCount - 1
    End If
    If Dec6.Value = True Then
        Datos6.AddItem " " & (deci1(5))
        Datos6.TopIndex = Datos6.ListCount - 1
    End If
    If u6 = True Then
        u6 = False
    End If
End If

If u7 = True Or det7 = True Then

    If Hex7.Value = True Then
        Datos7.AddItem " " & (aux1(6))
        Datos7.TopIndex = Datos7.ListCount - 1
    End If
    If Dec7.Value = True Then
        Datos7.AddItem " " & (deci1(6))
        Datos7.TopIndex = Datos7.ListCount - 1
    End If
    If u7 = True Then
        u7 = False
    End If
End If

If u8 = True Or det8 = True Then
    auxtemp = 12 + (deci1(7) - 3) * 10
    Datos8.AddItem " " & auxtemp
    Datos8.TopIndex = Datos8.ListCount - 1

    If u8 = True Then
        u8 = False
    End If
End If

End Sub
```



'SUBROUTINA DE INICIALIZACIÓN DE TODAS LAS VARIABLES NECESARIAS:

```
Private Sub inicializar()
una1.Value = True
una2.Value = True
una3.Value = True
una4.Value = True
una5.Value = True
una6.Value = True
una7.Value = True
una8.Value = True
det1 = False
det2 = False
det3 = False
det4 = False
det5 = False
det6 = False
det7 = False
det8 = False
u1 = False
u2 = False
u3 = False
u4 = False
u5 = False
u6 = False
u7 = False
u8 = False
Recep = "1"
End Sub
```

'FORMULARIO QUE MUESTRA LOS DATOS ADQUIRIDOS EN FORMA DE 'GRÁFICOS (BARRAS).

'DECLARACIÓN DE VARIABLES

```
Option Explicit
Public dirIP, Npuerto As String
Dim det1, u1, det2, u2, det3, u3, det4, u4, det5, u5, det6, u6, det7, u7, det8, u8 As Boolean
Dim Dec(31), i As Integer
Dim enviar, Recep, Aux(31), aux1(8) As String
Dim deci1(8), auxtemp As Single
```

'INICIALIZACIÓN DE VARIABLES

```
Private Sub Form_Load()
Inicializar           'SE INVOCA AL EVENTO "Inicializar" PARA UBICAR LOS VALORES
                       'INICIALES A TODAS LAS VARIABLES NECESARIAS.

Winsock1.RemoteHost = IP.dirIP      'SE UBICA LA DIRECCIÓN DEL SERVIDOR INGRESADA
                                     'EN EL CAMPO "RemoteHost"

Winsock1.LocalPort = 888           'SE ASIGNA EL NUMERO DE PUERTO LOCAL

Winsock1.RemotePort = 888         'SE ASIGNA EL NUMERO DE PUERTO REMOTO

Winsock1.Connect           'PETICION DE CONEXION CON EL SERVIDOR REMOTO
    i = 0
End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 0

Private Sub inicia1\_Click()

```
If Timer1.Enabled = False Then      'SI EL TIMER QUE REALIZA LAS PETICIONES CADA 300
                                     'MILISEGUNDOS ESTA INACTIVO
    Timer1.Enabled = True           'SE ENCIENDE
End If
```

```
u1 = False                          'VARIABLE QUE INDICA SI SE PETICIONÓ SENSAR CONTINUAMENTE EL
det1 = False                         'CANAL (u1 = False → SE PIDIO SENSAR CONTINUAMENTE)
```

```
If Cont1 = True Then                'SI SE ESCOGIO SENSAR CONTINUAMENTE
    If inicia1.Caption = "Iniciar" Then 'Y LA LEYENDA DEL BOTON DICE "Iniciar"
        Cont1.Enabled = False         'VARIABLE QUE INDICA QUE SE PRESIONO "Iniciar"
        inicia1.Caption = "Detener"   'SE UBICA COMO NUEVA LEYENDA "Detener"
        det1 = True                   'VARIABLE PARA QUE EL TIMER REALICE PETICIONES
                                     'CONTINUAS
```

```
    End If
Else
u1 = True                            'SI NO SE PIDIO SENSAR CONTINUAMENTE EL CANAL, ESTA VARIABLE
                                     'SE UBICA EN TRUE
End If
```

```
If inicia1.Caption = "Detener" And det1 = False Then 'SI SE PREISIONÓ EL BOTON "Detener"
    inicia1.Caption = "Iniciar"                     'LA LEYENDA DEL BOTÓN
                                                     'CAMBIA A "Iniciar"

    Cont1.Enabled = True                            'VALOR DE LA VARIABLE QUE INDICA QUE SE DEBEN DETENER
                                                     'LAS PETICIONES
```

'SI SE PRESIONÓ INICIAR PERO UNA SOLA VEZ, SE OBSERVA TODOS LOS CANALES: SI 'NINGUNO PETICIONÓ SENSAR SU CANAL SE DESHABILITA 'EL TIMER PARA LA CONVERSIÓN Y 'SE DETIENE EL PROCESO A LA ESPERA DE UNA TECLA:

```
    If det2 = False And det3 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub
```

'A continuación se muestra la codificación utilizada en los demás canales, pero la 'documentación se suprime ya que es la misma:

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 1

Private Sub inicia2\_Click()

```
If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If
```

```
u2 = False
det2 = False
```

```
If Cont2.Value = True Then

    If inicia2.Caption = "Iniciar" Then
```

```

    Cont2.Enabled = False
    inicia2.Caption = "Detener"
    det2 = True
End If

```

```

Else
u2 = True
End If

```

```

If inicia2.Caption = "Detener" And det2 = False Then
    inicia2.Caption = "Iniciar"
    Cont2.Enabled = True
    If det1 = False And det3 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
        Timer1.Enabled = False
    End If

```

```

Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 2

```

Private Sub inicia3_Click()

```

```

    If Timer1.Enabled = False Then
        Timer1.Enabled = True
    End If

```

```

    u3 = False
    det3 = False

```

```

    If Cont3.Value = True Then
        If Inicia3.Caption = "Iniciar" Then
            Cont3.Enabled = False
            Inicia3.Caption = "Detener"
            det3 = True
        End If

```

```

Else
u3 = True
End If

```

```

    If Inicia3.Caption = "Detener" And det3 = False Then
        Inicia3.Caption = "Iniciar"
        Cont3.Enabled = True
        If det1 = False And det2 = False And det4 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
            Timer1.Enabled = False
        End If

```

```

Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 3

```

Private Sub inicia4_Click()

```

```

    If Timer1.Enabled = False Then
        Timer1.Enabled = True

```

End If

u4 = False  
det4 = False

```
If Cont4.Value = True Then
  If Inicia4.Caption = "Iniciar" Then
    Cont4.Enabled = False
    Inicia4.Caption = "Detener"
    det4 = True
  End If
```

```
Else
  u4 = True
End If
```

```
If Inicia4.Caption = "Detener" And det4 = False Then
  Inicia4.Caption = "Iniciar"
  Cont4.Enabled = True
  If det1 = False And det2 = False And det3 = False And det5 = False And det6 = False And
det7 = False And det8 = False Then
    Timer1.Enabled = False
  End If
Else
  End If
End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 4

Private Sub inicia5\_Click()

```
If Timer1.Enabled = False Then
  Timer1.Enabled = True
End If
```

u5 = False  
det5 = False

```
If Cont5.Value = True Then
  If Inicia5.Caption = "Iniciar" Then
    Cont5.Enabled = False
    Inicia5.Caption = "Detener"
    det5 = True
  End If
```

```
Else
  u5 = True
End If
```

```
If Inicia5.Caption = "Detener" And det5 = False Then
  Inicia5.Caption = "Iniciar"
  Cont5.Enabled = True
  If det1 = False And det2 = False And det3 = False And det4 = False And det6 = False And
det7 = False And det8 = False Then
    Timer1.Enabled = False
  End If
Else
  End If
End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 5

```
Private Sub inicia6_Click()
```

```
    If Timer1.Enabled = False Then
        Timer1.Enabled = True
    End If
```

```
    u6 = False
    det6 = False
```

```
    If Cont6.Value = True Then
        If Inicia6.Caption = "Iniciar" Then
            Cont6.Enabled = False
            Inicia6.Caption = "Detener"
            det6 = True
        End If
```

```
    Else
        u6 = True
    End If
```

```
    If Inicia6.Caption = "Detener" And det6 = False Then
        Inicia6.Caption = "Iniciar"
        Cont6.Enabled = True
        If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det7 = False And det8 = False Then
            Timer1.Enabled = False
        End If
```

```
    Else
        End If
    End Sub
```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 6

```
Private Sub inicia7_Click()
```

```
    If Timer1.Enabled = False Then
        Timer1.Enabled = True
    End If
```

```
    u7 = False
    det7 = False
```

```
    If Cont7.Value = True Then
        If Inicia7.Caption = "Iniciar" Then
            Cont7.Enabled = False
            Inicia7.Caption = "Detener"
            det7 = True
        End If
```

```
    Else
        u7 = True
    End If
```

```
    If Inicia7.Caption = "Detener" And det7 = False Then
        Inicia7.Caption = "Iniciar"
        Cont7.Enabled = True
        If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det6 = False And det8 = False Then
```

```

    Timer1.Enabled = False
End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR SI SE PRESIONA EL BOTON INICIAR EN EL CANAL 7

```
Private Sub inicia8_Click()
```

```

If Timer1.Enabled = False Then
    Timer1.Enabled = True
End If

```

```

u8 = False
det8 = False

```

```

If Cont8.Value = True Then
    If Inicia8.Caption = "Iniciar" Then
        Cont8.Enabled = False
        Inicia8.Caption = "Detener"
        det8 = True
    End If

```

```

Else
u8 = True
End If

```

```

If Inicia8.Caption = "Detener" And det8 = False Then
    Inicia8.Caption = "Iniciar"
    Cont8.Enabled = True
    If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And
det6 = False And det7 = False Then
        Timer1.Enabled = False
    End If
Else
End If
End Sub

```

'ACCIONES A EJECUTAR CADA VEZ QUE EL TIMER ESTÉ HABILITADO Y 'TRANSCURRAN 300 MILISEGUNDOS

```
Private Sub Timer1_Timer()
    enviar = "0"
```

```
Winsock1.SendData enviar
```

```

'ALMACENAR EN ESTA VARIABLE UN VALOR CUALQUIERA,
'PARA ENVIARLO A TRAVÉS DEL
'PUERTO DE RED, ACCIÓN QUE INDICA QUE EL CLIENTE ESTÁ
'ENVIANDO UNA SOLICITUD POR UN DATO ADQUIRIDO
REMOTAMENTE.

```

'COMPROBANDO SI SE HA PRESIONADO DETENER EN TODOS LOS 'CANALES QUE ESTABAN SENSANDO CONTINUAMENTE:

```

If det1 = False And det2 = False And det3 = False And det4 = False And det5 = False And det6
= False And det7 = False And det8 = False Then

```

'COMPROBANDO SI SE PRESIONARON ÚNICAMENTE CONVERSIONES 'SIMPLES:

```

If u1 = True Or u2 = True Or u3 = True Or u4 = True Or u5 = True Or u6 = True Or u7 = True Or
u8 = True Then
    Timer1.Enabled = False
End If

```

'SI SE CUMPLEN ESTAS CONDICIONES, SE 'DESHABILITA EL TIMER

```
End If
End Sub
```

'PROCEDIMIENTO A REALIZAR CUANDO SE RECIBA LOS DATOS PETICIONADOS 'A TRAVÉS DEL PUERTO SERIAL.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
```

```
    Winsock1.GetData Recep          'SE ALMACENA LO RECIBIDO EN LA 'VARIABLE "Recep"
    procesamiento                  'SE INVOCA A LA SUBROUTINA "procesamiento"
    envío                          'SE INVOCA A LA SUBROUTINA "envío"
```

```
End Sub
```

'ACCIONES REALIZADAS EN LA SUBROUTINA PROCESAMIENTO

```
Private Sub procesamiento()
```

'LA INFORMACIÓN RECIBIDA ES UN TREN DE 32 CARACTERES HEXADECIMALES QUE PARA 'MAYOR FACILIDAD SE ALMACENAN UNO POR UNO EN UN VECTOR DE TAMAÑO 32, PARA 'LUEGO UTILIZARLO EN LA CONVERSIÓN A DECIMAL:

```
    For i = 1 To 32 Step 1
        Aux(i - 1) = Mid(Recep, i, 1)
    Next
```

'SE COMPRUEBA QUE LOS VALORES DEL VECTOR DE TAMAÑO 32 ESTÉN 'DENTRO DEL RANGO '0 - 9, CASO CONTRARIO SE TRANSFORMAN A SU 'EQUIVALENTE DECIMAL (A→10, B→11, C→12, 'D→13, E→14, F→15)

```
    For i = 0 To 31 Step 1
        If Aux(i) = "A" Then
            Dec(i) = 10
        ElseIf Aux(i) = "B" Then
            Dec(i) = 11
        ElseIf Aux(i) = "C" Then
            Dec(i) = 12
        ElseIf Aux(i) = "D" Then
            Dec(i) = 13
        ElseIf Aux(i) = "E" Then
            Dec(i) = 14
        ElseIf Aux(i) = "F" Then
            Dec(i) = 15
        Else
            Dec(i) = Val(Aux(i))
        End If
    Next
```

'SE TRANSFORMA A DECIMAL EL VECTOR TAMAÑO 8: MULTIPLICANDO 'EL VALOR MÁS SIGNIFICATIVO POR 16<sup>3</sup> Y SUMANDO AL SIGUIENTE 'MULTIPLICADO POR 16<sup>2</sup> Y AL SIGUIENTE MULTIPLICADO POR 16 Y 'SUMADO AL MENOS SIGNIFICATIVO, LUEGO ESTE RESULTADO SE DIVIDE 'POR 819.1632, QUE ES EL VALOR EQUIVAMENTE DE LA CONVERSIÓN 0FFF 'HEXADECIMAL A 5.000 DECIMAL Y ESTE VALOR SE LO REDONDEA CON 'TRES CIFRAS DECIMALES

```
    For i = 0 To 30 Step 4
        deci1(i / 4) = Round((Dec(i) * (16 * 16 * 16) + Dec(i + 1) * (16 * 16) +
            + Dec(i + 3)) / 819.1638, 3)
    Next
```

```
End Sub
```

'PROCEDIMIENTO A REALIZAR POR LA SUBROUTINA ENVIO:

Private Sub envio()

```
If u1 = True Or det1 = True Then      'COMPROBANDO SI EL CANAL 0 REQUIERE DATOS
    bar1.Value = deci1(0)            'ADQUIRIDOS
                                     'SI ES ASÍ, LO MUESTRA EN LA BARRA DE VALORES
                                     'CORRESPONDIENTE
```

'COMPROBANDO SI LA PETICIÓN FUE SOLO POR UNA CONVERSIÓN SIMPLE, SIENDO ESTE EL 'CASO SE CONSIFURA EN "False" LA VARIABLE "u1" INDICANDO QUE SE ATENDIÓ LA PETICIÓN.

```
    If u1 = True Then
        u1 = False
    End If
End If
```

'EL MISMO PROCESO SE SIGUEN EN TODOS LOS CANALES, RAZÓN POR LA 'QUE SOLO SE MUESTRA LA CODIFICACIÓN:

```
If u2 = True Or det2 = True Then
    bar2.Value = deci1(1)
    If u2 = True Then
        u2 = False
    End If
End If
```

```
If u3 = True Or det3 = True Then
    bar3.Value = deci1(2)
    If u3 = True Then
        u3 = False
    End If
End If
```

```
If u4 = True Or det4 = True Then
    bar4.Value = deci1(3)
    If u4 = True Then
        u4 = False
    End If
End If
```

```
If u5 = True Or det5 = True Then
    bar5.Value = deci1(4)
    If u5 = True Then
        u5 = False
    End If
End If
```

```
If u6 = True Or det6 = True Then
    bar6.Value = deci1(5)
    If u6 = True Then
        u6 = False
    End If
End If
```

```
If u7 = True Or det7 = True Then
    bar7.Value = deci1(6)
    If u7 = True Then
        u7 = False
    End If
End If
```



```

If u8 = True Or det8 = True Then
  auxtemp = 12 + (dec1(7) - 3) * 10
  temp.Value = auxtemp
  If u8 = True Then
    u8 = False
  End If
End If

```

```
End Sub
```

'SUBROUTINA DE INICIALIZACIÓN DE TODAS LAS VARIABLES NECESARIAS:

```

Private Sub inicializar()
una1.Value = True
una2.Value = True
una3.Value = True
una4.Value = True
una5.Value = True
una6.Value = True
una7.Value = True
una8.Value = True
det1 = False
det2 = False
det3 = False
det4 = False
det5 = False
det6 = False
det7 = False
det8 = False
u1 = False
u2 = False
u3 = False
u4 = False
u5 = False
u6 = False
u7 = False
u8 = False
Recep = "1"
End Sub

```

## 2. CODIGO FUENTE DEL PROGRAMA CLIENTE "server.exe"

'FORMULARIO PARA ELEGIR EL PUERTO ACTIVO DEL COMPUTADOR QUE 'VA A HACER DE SERVIDOR (CUANDO NO SE UTILIZA EL EZL - 300).

```
Public Npuerto          'DECLARANDO LA VARIABLE "Npuerto", EN DONDE SE VA A
                        'ALMACENAR EL PUERTO ACTIVO ESCOGIDO PARA UTILIZARLO
                        'EN EL SIGUIENTE FORMULARIO
```

'PROCEDIMIENTO A REALIZAR CUANDO SE PRESIONA ACEPTAR EL 'NÚMERO DE PUERTO

```
Private Sub Command1_Click()
Npuerto = Val(puerto.Text)          'VALIDAR NUMÉRICAMENTE EL NÚMERO DE PUERTO
                                    'ESCOGIDO Y ALMACENARLO EN LA VARIABLE "Npuerto"
```

```
Unload Me          'CERRAR EL FORMULARIO ACTUAL Y
Server.Show        'ABRIR EL FORMULARIO "Server"
```

End Sub

'FORMULARIO PARA ESPERAR POR UNA PETICIÓN DE CONEXIÓN DE UN CLIENTE REMOTO Y  
'LUEGO NUMERAR LAS VECES QUE SE HAN REALIZADO TRANSACCIONES EXITOSAS CON ÉSTE

Dim var, enviar As String, cont As Integer 'DECLARANDO LAS VARIABLES A UTILIZAR

'PROCEDIMIENTO A REALIZAR CUANDO SE PRESIONA "Escuchar una petición de conexión"

Private Sub Command1\_Click()

Winsock1.Listen 'SE INICIA EL EVENTO "Listen" DEL WINSOCK, PARA  
espera.Visible = True 'APARECIMIENTO DE UN TEXTO INDICANDO PROGRESO  
'DE LA CONEXIÓN

Command1.Visible = False 'DESAPARECER EL BOTÓN DE ESPERAR POR UNA  
'CONEXIÓN

progreso.Visible = True 'APARECIMIENTO DE UN INDICADOR DEL PROGRESO  
DE LA CONEXIÓN

Timer.Enabled = True 'HABILITACIÓN DEL TIMER QUE CONTROLA  
'INICIALMENTE EL INDICADOR DE PROGRESO DE LA  
'CONEXIÓN

End Sub

'PROCEDIMIENTO A REALIZAR CUANDO SE ABRE EL FORMULARIO

Private Sub Form\_Load()

cont = 0 'EL CONTADOR DE TRANSACCIONES DE PONE EN CERO  
espera.Visible = False 'SE OCULTA EL TEXTO INDICANDO PROGRESO DE LA  
'CONEXIÓN

var = "x" 'ALMACENANDO UN VALOR EN LA VARIABLE VAR PARA LUEGO  
'ENVIARLO POR EL PUERTO SERIAL PARA QUE EL  
'MICROCONTROLADOR ENVÍE LOS DATOS ADQUIRIDOS.

MSComm1.CommPort = Ingreso.Npuerto 'CONFIGURANDO EL PUERTO SERIAL CON EL  
'NÚMERO DE PUERTO 'ESCOGIDO EN EL  
'FORMULARIO ANTERIOR

MSComm1.PortOpen = True 'HABILITANDO O ABRIENDO EL PUERTO

End Sub

'PROCEDIMIENTO A REALIZAR CUANDO SE ACTIVE EL PUERTO SERIAL POR LA RECEPCIÓN DE  
'CARACTERES EXTERNOS

Private Sub MSComm1\_OnComm()

enviar = MSComm1.Input 'ALMACENANDO LOS CARACTERES RECIBIDOS EN LA  
'VARIABLE "enviar"

Winsock1.SendData enviar 'ENVIANDO LO RECIBIDO POR EL PUERTO SERIAL  
'HACIA EL CLIENTE PETICIONARIO DE DATOS POR EL  
'PUERTO DE RED

End Sub

'PROCEDIMIENTO A REALIZAR CUANDO SE ENCIENDE EL TIMER DE CONTROL INICIAL DEL  
'INDICADOR DE PROGRESO DE LA CONEXIÓN CON EL CLIENTE

Private Sub Timer\_Timer()

progreso.Value = progreso.Value + 5 'EL INDICADOR AVANZA 5% A LA VEZ  
Timer.Interval = Timer.Interval + 100 'EL TIMER SE HACE 100 MILISEGUNDOS MÁS  
'RÁPIDO CADA VEZ QUE EL INDICADOR AVANZA

If progreso.Value = 30 Then 'CUANDO LA BARRA DE PROGRESO SE UBIQUE EN 30%  
Timer.Enabled = False 'EL TIMER SE DESHABILITA

End If

End Sub

'PROCEDIMIENTO A REALIZAR CUANDO EL TIMER DE CONTROL FINAL DEL INDICADOR DE  
'PROGRESO SE ENCIENDE POR LA ACEPTACIÓN DE LA PETICIÓN DE UN CLIENTE REMOTO

Private Sub Timer1\_Timer()

progreso.Value = progreso.Value + 5

'EL INDICADOR AVANZA 5% A LA VEZ

If progreso.Value = 100 Then

'CUANDO LA BARRA DE PROGRESO SE UBIQUE  
'EN 100%

Timer1.Enabled = False

'EL TIMER SE DESHABILITA

End If

escucha.Caption = "Número de transacciones exitosas"

'APARECE EL TEXTO "Número de  
'transacciones exitosas" EN EL  
'FORMULARIO

progreso.Visible = False

'SE OCULTA LA BARRA DE PROGRESO

Text2.Visible = True

'APARECE LA CAJA DE TEXTO EN DONDE APARECERÁ  
'EL NÚMERO DE TRANSACCIONES COMPLETADAS.

End Sub

'ACCIONES A REALIZAR CUANDO SE RECIBA UNA PETICIÓN DE 'CONEXIÓN POR MEDIO DEL  
PUERTO DE RED

Private Sub Winsock1\_ConnectionRequest(ByVal requestID As Long)

Winsock1.Close

'CERRAR EL PUERTO DE RED

Winsock1.Accept requestID

'ACEPTAR LA PETICIÓN

Timer1.Enabled = True

'HABILITANDO EL TIMER ENCARGADO DE GRAFICAR EL  
'PROGRESO DE LA CONEXIÓN

End Sub

'ACCIONES A REALIZAR CUANDO SE RECIBA UN DATO POR EL PUERTO 'DE RED (QUE SIGNIFICA  
UNA PETICIÓN DE DATOS)

Private Sub Winsock1\_DataArrival(ByVal bytesTotal As Long)

Dim datos As String

'DECLARANDO LA VARIABLE EN LA QUE SE VA A ALMACENAR  
'LOS CARACTERES RECIBIDOS

Winsock1.GetData datos

'ALMACENANDO LA CADENA DE CARACTERES RECIBIDA

MSComm1.Output = var

'ENVIANDO UN CARÁCTER ALMACENADO EN LA VARIABLE "var"  
'AL PUERTO SERIAL PARA PETICIONAR LOS DATOS  
'ADQUIRIDOS POR EL MAX180 Y ENVIADOS POR EL  
'MIRCROCONTROLADOR

cont = cont + 1

'AUMENTANDO EN 1 EL CONTADOR DE TRANSACCIONES  
EXITOSAS

Text2.Text = cont

'MOSTRAR ESTE NÚMERO EN EL CAJON DE TEXTO  
'RESPECTIVO

End Sub

## Código Fuente del Programa Desarrollado en formato Assembler (Para el microcontrolador 89c52)

; Etiquetas de los pines del MAX180 conectados con el Microcontrolador en los pines del  
;puerto 1

```
CSEL      EQU   P1.0   ;HABILITACIÓN DEL DISPOSITIVO
HBEN      EQU   P1.1   ;ENTRADA DE HABILITACIÓN DEL BYTE MÁS
                        ;SIGNIFICATIVO (EN LA CONVERSIÓN Y LECTURA DE
                        ;DATOS)
BUSY      EQU   P1.2   ;INDICACIÓN DE QUE EL DISPOSITIVO SE ENCUENTRA
                        ;PROCESANDO EL DATO ADQUIRIDO
```

; Etiquetas de las localidades de memoria en donde se van a almacenar los valores resultantes  
; de la lectura del MAX180 (Dos bytes)

```
LSB_CONV  EQU   60H
MSB_CONV  EQU   61H
```

;Inicio del programa

```
INICIO:    MOV     SP,#2FH
```

;Lamada a la subrutina de inicialización de los parámetros del puerto serial.

```
LCALL INI_SER
```

;Llamada a la subrutina de espera de un caracter recibido por el puerto serial para iniciar el  
;proceso con el MAX180

```
LECTURA:  LCALL RECEP
```

```
MOV R1,#00 ;Este registro servirá como contador de los
            ;ocho canales (0...7)
```

```
RETORNO:   MOV A,R1 ;Se transfiere el número del canal al acumulador para
                ;solicitar el dato adquirido en dicho canal
```

;Llamada a la subrutina de petición y lectura del dato adquirido en el MAX180 en un canal  
;determinado por el valor del acumulador

```
LCALL LECTMAX
```

```
MOV A,MSB_CONV ;Enviando al acumulador el valor más
                ;significativo de la lectura del MAX180
```

```
LCALL SENDASC ;Enviando el byte más significativo
                ;almacenado en el acumulador mediante la
                ;invocación de la subrutina de envío de un
                ;caracter por el puerto serial
```

```

MOV      A,LSB_CONV      ;Enviando al acumulador el valor menos
                        ;significativo de la lectura del MAX180

LCALL    SENDASC         ;Enviando el byte menos significativo
                        ;almacenado en el acumulador mediante
                        ;la invocación de la subrutina de envío
                        ;de un caracter por el puerto serial

INC      R1              ;Se incrementa el registro del conteo de los canales

CJNE     R1,#08,RETORNO  ;Si se han sentido los 8 canales, salir
                        ;del lazo caso contrario solicitar el dato
                        ;adquirido del canal incrementado en 1

SJMP     LECTURA        ;Al terminar el lazo se retorna a esperar
                        ;por un caracter recibido por el puerto

```

serial

;Subrutina de inicialización de la transmisión serial con los valores de velocidad, bits de parada y paridad. Esta transmisión se realiza utilizando el timer 1 en modo autorecarga.

```

INI_SER:  MOV      TH1,#0E8H      ;Valor para una velocidad de
                        ;19200 baudios
          MOV      PCON,#00H      ;Velocidad normal del puerto serial
          MOV      TMOD,#20H      ;Timer 1 configurado en modo de
                        ;autorecarga
          MOV      SCON,#40H      ;Un bit de parada, sin bit de paridad
          SETB     TR1            ;Habilitando el timer
          RET

```

;Subrutina de espera por un caracter recibido por el puerto serial

```

RECEP:   CLR      RI            ;Blanqueando la bandera de recepción
          SETB     REN          ;Habilitando la recepción serial
          JNB      RI,$         ;Esperando por la recepción completa de un
                        ;caracter
          CLR      REN          ;Deshabilitando la recepción
          RET

```

;Subrutina de petición y lectura de los datos adquiridos por el MAX180 en un canal determinado

```

LECTMAX: CLR      CSEL         ;Habilitando el MAX180
          CLR      HBEN        ;Habilitando la lectura del byte menos
                        ;significativo

          MOV      R0,#00H      ;Valor que controlará el tipo de adquisición, en
                        ;este caso, unipolar, no diferencial, en modo 0.

          MOVX     @R0,A        ;Petición del valor con el tipo de
                        ;adquisición y el canal escogido a modo de
                        ;escritura de memoria externa

          JB       BUSY,$       ;Esperando por el pulso del pin BUSY del
                        ;MAX180 que
          JNB      BUSY,$       ;indica que la adquisición se ha completado con
                        ;éxito y que el dato está listo para ser leído.

          MOVX     A,@R0        ;Leyendo el valor byte menos significativo
                        ;entregado por el MAX180 a modo de lectura
                        ;de memoria externa

```

```

MOV      LSB_CONV,A ;Almacenando el byte adquirido para enviarlo
                        ;luego

SETB     BUSY        ;Habilitando la lectura del byte
SETB     HBEN        ;menos significativo

MOVX     A,@R0       ;leyendo el byte más significativo como
                        ;lectura de memoria externa

ANL      A,#0FH      ;Se asegura que el valor recibido sea igual o
                        ;menor a 0FH ya el valor máximo que el
                        ;convertor envía es 0FFFH (5 v) y así se evita
                        ;que alguna señal introducida se envíe

MOV      MSB_CONV,A ;Almacenando el byte adquirido para enviarlo
                        ;luego

SETB     CSEL        ;Se deshabilita el MAX180
RET

```

;Subrutina de transmisión por el pórtilo serial de un caracter almacenado en el acumulador

```

TRANS:   CLR          TI          ;Blanqueando la bandera de transmisión
          MOV         SBUF,A      ;Envío del caracter
          JNB         TI,$        ;Esperando que se haya completado la
                                ;transmisión

          RET

```

;Subrutina de envío al pórtilo serial del contenido del acumulador como dos caracteres  
;ASCII Primero el Nibble más significativo Y luego el menos significativo

```

SENDASC: PUSH         ACC         ;Guardando el valor del acumulador
          SWAP        A           ;Intercambio de nibbles del acumulador
          ANL         A,#0FH      ;Anulando el nibble más significativo
          ADD         A,#12H      ;Sumando 12H al valor del acumulador que
                                ;es el número exacto que se debe aumentar al
                                ;puntero del programa para

          MOVC        A,@A+PC     ;obtener el código ASCII de este número en la
                                ;tabla de abajo (1+12H = 30H,...)

          LCALL       TRANS       ;Invocando la subrutina de transmisión serial
                                ;del valor del acumulador

          POP         ACC         ;Recuperando el valor inicial del acumulador
          PUSH        ACC         ;Guardando el valor del acumulador
          ANL         A,#0FH      ;Anulando el nibble más significativo
          ADD         A,#06H      ;Sumando 06H al valor del acumulador que
                                ;es el número exacto que se debe aumentar al
                                ;puntero del programa para

          MOVC        A,@A+PC     ;obtener el código ASCII de este número en la
                                ;tabla de abajo (1+06H = 30H,...)

          LCALL       TRANS       ;Invocando la subrutina de transmisión serial de
                                ;valor del acumulador

          POP         ACC         ;Recuperando el valor inicial del acumulador
          RET

```

;Tabla de equivalencias de los números hexadecimales con sus códigos ASCII

```
DB    30H,31H,32H,33H,34H
DB    35H,36H,37H,38H,39H
DB    41H,42H,43H,44H,45H
DB    46H
```

```
DB    00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
```

```
END   ;Fin del Módulo Fuente
```

## ANEXO 2

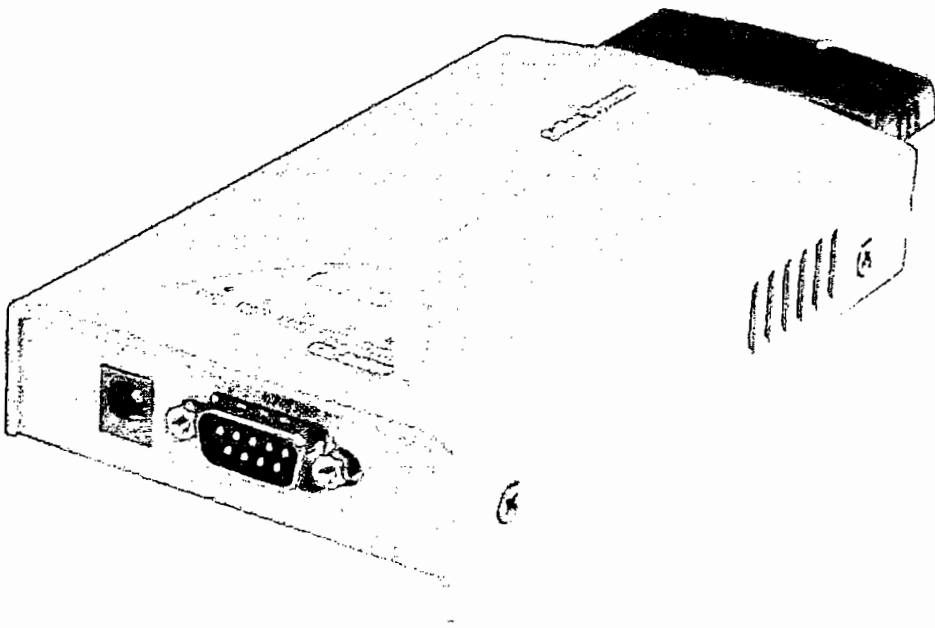
EL EZL – 300W



ezTCP/LAN(EZL-300W)

# User's Manual

Revision B



Sollae Systems

# 1. Overview

## 1.1. Overview

With the development of Internet technologies, demands for data communication through the Internet keep increasing. However, to have data communication through the Internet, users must use TCP/IP protocol which is the basic communication protocol of the Internet. In other words, to connect a device to the Internet, the user must implement TCP/IP protocol. To implement TCP/IP protocol, the user must implement TCP/IP protocol by themselves, port open TCP/IP, or use an OS. However, these methods are quite burdensome in terms of time, cost, and technology.

As a family of serial to TCP/IP protocol converters, ezTCP series manufactured by Sollae Systems provide TCP/IP communication functions (Internet communication functions) "only through serial port connection." The ezTCP series perform TCP/IP protocol processing on data received from the serial port, and send the processed data to the Internet. Similarly, the ezTCP series perform TCP/IP protocol processing on data received from the Internet and sends the processed data to the serial port.

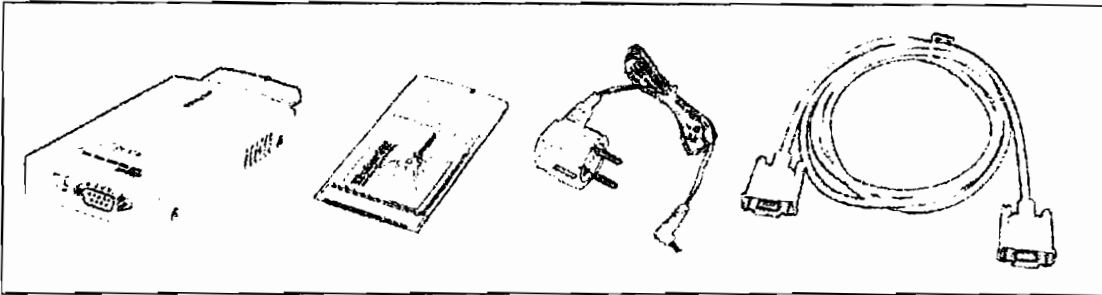
EZL-300W, which is one of ezTCP products, provides TCP/IP communication through IEEE802.11b (wireless LAN). In other words, like other ezTCP products, EZL-300W TCP/IP-processes data received from the serial port and sends the processed data to the wireless LAN network. And EZL-300W TCP/IP-processes data received from the wireless LAN network and sends the processed data to the serial port.

EZL-300W supports the infrastructure network through the Access Point (AP) and also provides an "ad-hoc" network function supporting peer to peer communication without any AP.

As EZL-300W provides DHCP as well as TCP/IP/UDP, EZL-300W can be applied to cable networks.

## 1.2. Components

- EZL-300W Main Body
- 3.3V PCMCIA Wireless LAN Card(16bit PC Card) Supporting IEEE802.11b
- 5V SMPS Adapter
- RS-232 Cable



*※ Sollae Systems recommends the use of the wireless LAN card provided by Sollae Systems. Sollae Systems does not guarantee successful operation of a wireless LAN card provided by other manufacturers.*

## 1.3. Product Specifications

Processor	16bit
Memory	ROM [256K], RAM [512K]
Serial Port	RS232 (up to 115.2Kbps full duplex), RTS/CTS, Parity
Wireless LAN	PCMCIA (IEEE802.11b Wireless LAN)
O/S	MIC Realtime Operating System
Protocols	TCP/UDP/IP/ICMP, ARP, DHCP, TELNET, EAP-MD5
Environment Variable Setting	ezConfig, ezSerialConfig, console, Telnet, ARP, ATcommand
Firmware Download	Serial Port (115200bps)
Operating Voltage	DC 5V
Current Consumption	410mA
Operating Temperature	0°C - 70°C
Storage Temperature	-40°C - 70°C

### 1.4.3. Status LED

In EZL-300W, there are total 5 status LEDs.

Each status LED functions as follows:

Name	Description	Color	LED Status	Description
PWR	Power	Red	Turned on	When the power is on
STS	Status	Yellow	Blinking once per second	When IP address is assigned, but no TCP connection established
			Blinking 4 times every 2 seconds	When no IP address was assigned(DHCP)
			Turned on	During a TCP connection
			Blinking rapidly	In the ISP mode
LINK	Wireless LAN Link	Green	Turned on	the wireless LAN LINK is established
RXD	Wireless LAN Reception	Yellow	Turned on	When a packet is received from the wireless LAN
TXD	Wireless LAN Transmission	Green	Turned on	When a packet is transmitted to the wireless LAN

## 2. Installation and Test Operation

### 2.1. Installation

Install EZL-300W as follows:

Categories	Sub-categories	Context	See
1. Checking Communication Environment	Items to check	Wireless LAN environment	3.1.1.
		IP address environment	
		Serial port setup values	
		Application program to use	
2. Configuring the Wireless LAN	Setup Procedure	Set via ezSerialConfig or console in Console mode. (For entering into Console mode, see 4.1.2.)	3.1.2.
	Setup Items	Wireless LAN Types (Infrastructure and ad-hoc)	3.1.1.
		Service Set Identification (SSID)	
		Channel	
	Whether to use WEP (If used, set key value)		
3. Connecting to the Network	Confirming Procedure	Checking link LED lightning status	
4. Setting Environment Variables	Setup Procedure	Set by ezConfig (a configuration utility through a network)	3.2.1.
		Telnet	3.2.2.
		Console mode	3.2.2.
		Using AT command in ATC mode	5.
		Using ARP (To set only the IP address temporarily)	3.2.4.
	Setup Items	IP address-related items	3.2.
		Serial port-related items	3.2.
Communication mode (To be decided according to the application program)		4.3.	
5. Field Application			

### 2.1.1. Checking Communication Environment

Before installing EZL-300W (hereinafter to be referred as the ezTCP), the user must check conditions of the network to install the ezTCP as follows:

- Wireless LAN network types (infrastructure/ad-hoc)
  - the SSID and the channel of the wireless LAN
  - Whether to use WEP and WEP setup values (the number of bits and key values)
  - Authentication Protocol for wireless LAN access
  - IP address environment (Local IP, subnet mask, gateway, etc.)
  - Serial port items of the device to connect (baudrate, databit, parity, and stop bit)
  - Protocol of the application program to use (TCP/UDP, server/client, etc.)
- ☞ For more information about the wireless LAN, see "3.1 Configuring the Wireless LAN."
- ☞ For more information about protocol of the application program to use, see "4.3 Normal Communication Mode."

### 2.1.2. Configuring the Wireless LAN

Before using the ezTCP, the user must set wireless LAN-related items. Wireless LAN-related items can be set via ezSerialConfig or console in a Console mode. Supply power without the PCMCIA card insertion to enter into Console mode. In the Console mode, the user can set not only wireless LAN-related items but also all setup values of the ezTCP.

- ☞ For more information about Console mode, see "4. Operation Modes."
- ☞ For more information about wireless LAN-related items in Console mode, see "3.1 Configuring the Wireless LAN."

### 2.1.3. Connecting to the Network

Insert the PCMCIA card while power is not supplied. Then, connect the device with the ezTCP using RS232 cable that is compliant with the specification, and supply power. After power is supplied, the link LED should be turned on. When the link LED is on, it means that a link has been established between the AP and wireless LAN device or between the wireless LAN devices to enable communication between

them.

#### 2.1.4. Setting Environment Variables

Once a connection to the network has been established, set environment variables - variables relating to the IP address, the serial port, and communication mode - using ezConfig which is an environment variable configuration program.

- ☛ For more information about environment variables, see “3.2 Setting IP Address and Environment Values.”

## 2.2. Test Operation

Perform test operation according to the following sequence. A method to perform the test operation will now be described, assuming that the network is configured as the infrastructure, the AP and the PC are directly connected via a crossed LAN cable, and the IP address of the PC is 10.1.0.2.

### 2.2.1. Changing the IP Address of the PC

Change the IP address of the PC as follows:

IP Address	10.1.0.2
Subnet Mask	255.0.0.0
Gateway IP Address	0.0.0.0

### 2.2.2. Installing the AP

Connect the AP with the PC through the crossed LAN cable.(When connecting through a hub, use a one-to-one cable for this connection.) After installing the cable, supply power to the AP as provided by the AP vendor. After supplying power, check the link LED indicating a connection between the AP and PC to make sure this connection is successfully made.

### 2.2.3. Installing EZL-300W

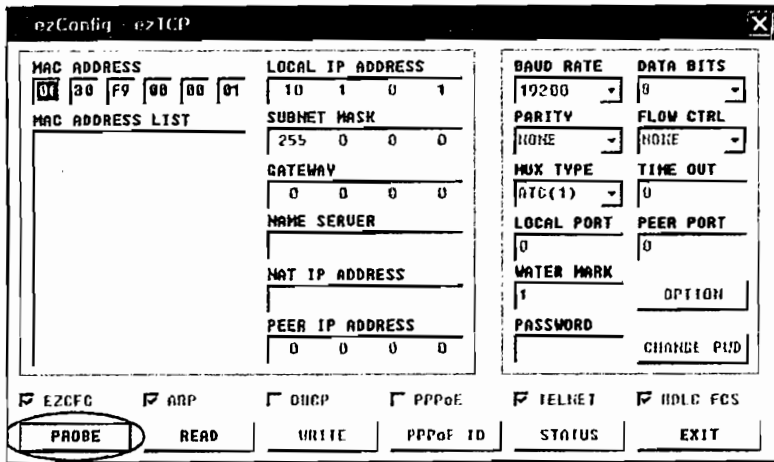
Connect the PC with EZL-300W using RS232 cable provided by Sollae Systems, and insert the PCMCIA card. When power is supplied using the power adapter designed for EZL-300W by Sollae Systems, EZL-300W attempts making a connection with the AP that replies first. Once EZL-300W is connected with the AP, the link LED is turned on and then a communication through the wireless LAN is possible.

### 2.2.4. Configuring EZL-300W

Use ezConfig (the ezTCP configuration program) and change the configuration of ezTCP as follows:

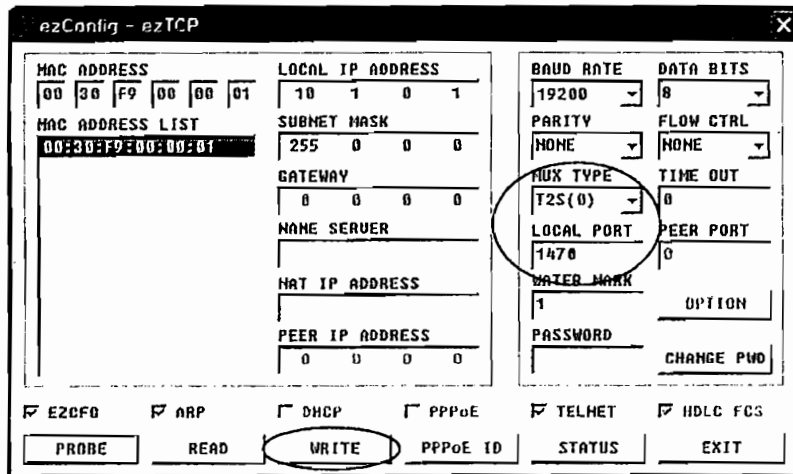
Run ezConfig, and click on [PROBE] button on the ezConfig window. Then, the ezConfig program will search for ezTCPs.





Once the ezTCP(s) is(are) found, the MAC address (hardware address) indicated at the bottom of the ezTCP case will be displayed in the [MAC ADDRESS LIST] window.

Select the corresponding MAC address, and change the [MUX TYPE] into [T2S(0)]. Then, enter 1470 in the [LOCAL PORT] field and click on [WRITE] button to save the setup values.



### 2.2.5. Connecting via the Serial Port of the PC

Connect between the serial port of the PC and the serial port of the ezTCP using the serial communication cable provided by Sollae Systems. Then, run a serial communication program such as Hyper Terminal. Select 19200bps for the serial port, 8 bits for the data bit, 1 bit for the stop bit, and no parity.

### 2.2.6. Actual Communication Test

Once the preparation for serial communication has been completed, enter the following in the command prompt of Windows, and establish a TCP connection using Telnet program.

```
"telnet 10.1.0.1 1470"
```

When TCP connection is successfully established, "STS" LED of the ezTCP is turned on.

If "123" data is transmitted to Telnet window after checking that "STS" LED is on, then "123" is displayed on Hyper Terminal. If "ABC" data is transmitted to Hyper Terminal window, then "ABC" data is displayed on Telnet window. It means the communication test is successful.

## 3. Configuring Wireless LAN, IP Address, and Environment Values

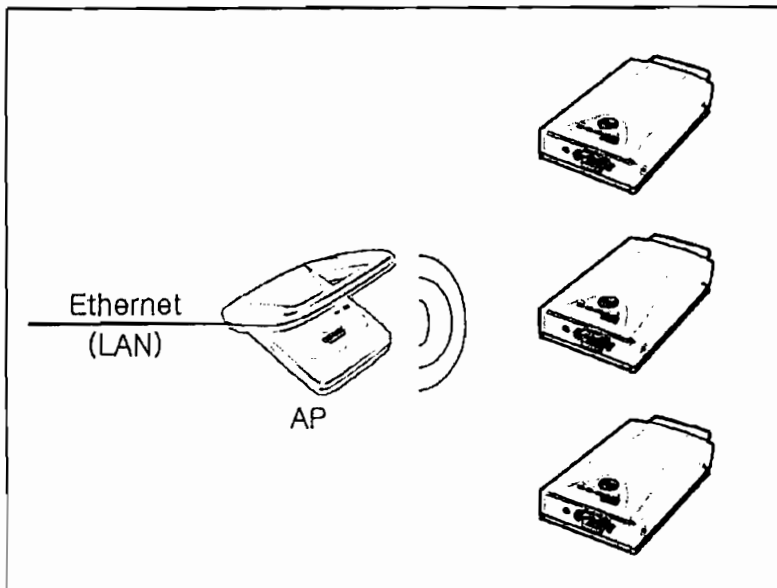
### 3.1. Configuring Wireless LAN

When configuring the wireless LAN, the user must check the network type and security issues of the network currently being used or to be installed in the future.

#### 3.1.1. Items to Check for the Wireless LAN

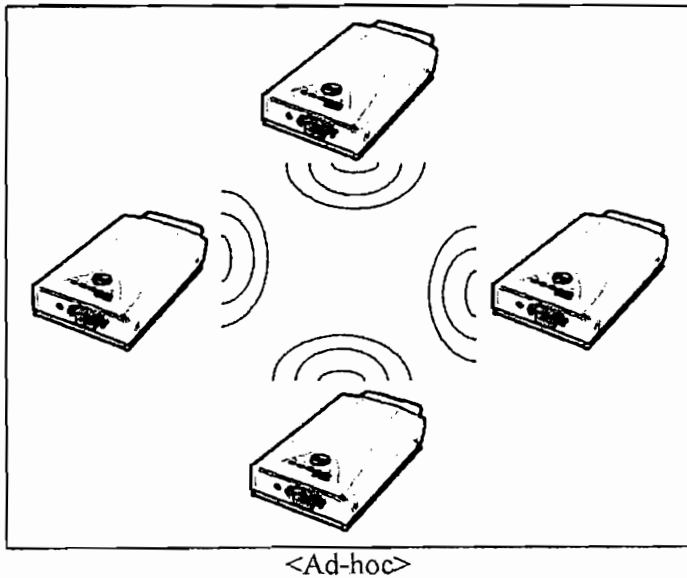
Network Type (infrastructure/ad-hoc)

The infrastructure is a network connection mode that allows communication between wireless LAN devices or between the wireless LAN and the wired LAN (Ethernet) through the Access Point (AP). When a network type is set to infrastructure, communication with wired LAN via AP is possible, which allows both wired and wireless Internet communications.



<Infrastructure>

The ad-hoc network is designed to communicate between wireless LANs without any AP. Since communication is established without any AP, the user cannot access an external network or the Internet. This is also called a peer-to-peer mode.



Service Set Identifier (SSID)

When configuring a network, the user can configure different networks using different APs. In this case, the SSID is used to differentiate one network from another. In other words, when configuring an infrastructure network, the user can make communication with the AP which he/she wants to communicate with by setting the SSID of the desired AP in the ezTCP. (See 3.1.2.) For information about SSID of the AP, AP manual or AP configuration program can be referred to. If the user did not set the SSID, the ezTCP will be connected with the AP that is first found when power is supplied.

The maximum length of the SSID is 32 bytes, and the user can use ASCII code to set the SSID.

Channel

The channel is communication path in the network that it belongs to. Channel values are set as the same value with the AP.

Wired Equivalent Privacy (WEP)

This is about security of the wireless LAN. The wireless LAN provides similar security to that of the wired LAN using the WEP. To use the WEP, the user must set the key value. According to the key value, data is encoded in 64 bits or 128 bits for communication. If the user did not set the WEP, security-related problems may occur.

Authentication Protocols

To avoid illegal access to wireless LAN, there are various authentication protocols.

EZL-300W supports the EAP-MD5 protocol. If a wireless LAN network uses EAP-MD5, set EAP-MD5 on.

### 3.1.2. Setting Wireless LAN Items of the ezTCP

There are two ways to configure wireless LAN-related items - Windows utility "ezSerialConfig" and via console.

The following table describes each configuration field.

Field	Description	Factory-set
TARGET SSID	SSID of the Network to Participate	Not set.
CREATE SSID	SSID of the Network to Newly Create When Configuring an Ad-hoc Network	Not set.
CC TYPE	0) IBSS: Ad-hoc Network	1
	1) BSS: Infrastructure Network	
	2) WDS: Reserved (Reserved)	
	3) Pseudo IBSS: Reserved (Reserved)	
CHANNEL	Channel number	0
WEP TYPE	0) WEP is not used.	0
	1) 64 bit WEP key	
	2) 128 bit WEP key	
KEY ID	Number of the Key to be Used	0
Authentication	For setting EAP-MD5	OFF

Setting via ezSerialConfig utility

① After connecting EZL-300W without PC card to COM port of PC via a serial cable, Turn on the EZL-300W.

② Execute ezSerialConfig.

③ After selecting COM port in ezSerialConfig, press [READ]. If you press [READ] button, The configuration values of EZL-300W will be shown.

④ After inputting configuration value what you want, Press [WRITE] button. The configuration values will be saved into the EEPROM area of EZL-300W.

CC TYPE	LOCAL IP ADDRESS	BAUD RATE	DATA BITS
Infrastructure(1)	10 1 0 1	19200	8
SSID	SUBNET MASK	PARITY	FLOW CTRL
CREATE SSID	255 0 0 0	NONE	NONE
CHANNEL	GATEWAY	MUX TYPE	TIME OUT
0	0 0 0 0	T2S(0)	0
WEP	NAME SERVER	LOCAL PORT	PEER PORT
No WEP(0)	NAT IP ADDRESS	1470	1470
WEP KEY	PEER IP ADDRESS	WATER MARK	OPTION
	255 255 255 255	0	CHANGE PWD
MAC 00:30:F9:00:4a:2b	<input checked="" type="checkbox"/> EZCFG <input checked="" type="checkbox"/> ARP <input type="checkbox"/> DHCP <input checked="" type="checkbox"/> TELNET <input type="checkbox"/> EAP OL		
CONN	READ	WRITE	CLOSE
		EAP ID	EXIT
Write Complete.			

Setting via console

When entering into Console mode, enter "env wlan" and set fields relating to the wireless LAN. (For setting EAPoL, enter "env if".)

If the user presses the Enter key in a field without entering any key, the user will use the existing value in the field. When the user completes entering data, the ezTCP is reset to restart operation.

The user can enter "." to delete existing values when setting the SSID.

☞ For more information about Console mode, see "4. Operation Mode."

### 3.2. Setting the IP Address and Environment Values

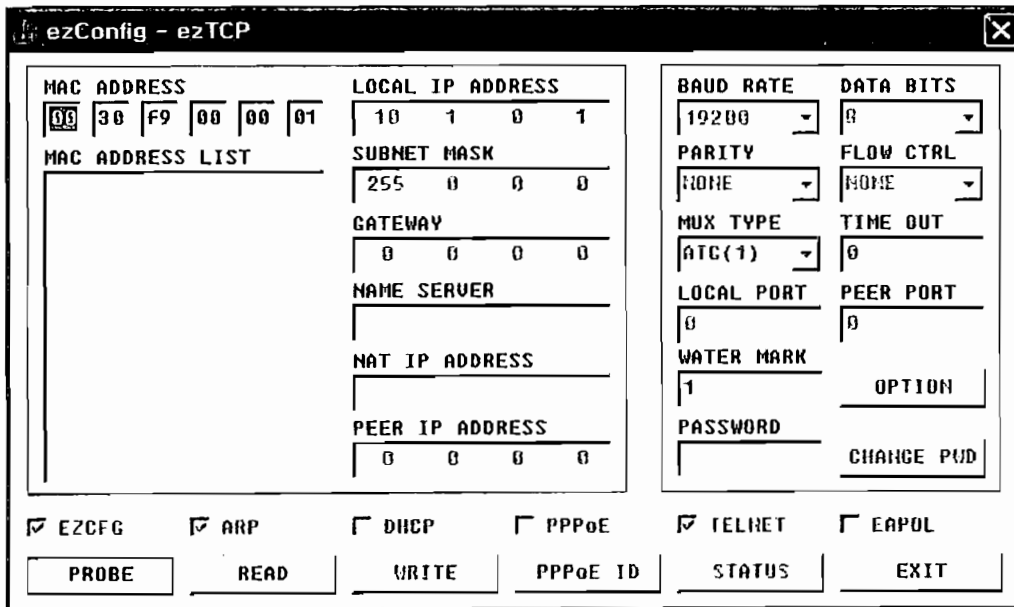
To establish TCP/IP communication, the user must set fields relating to the IP address as well as fields relating to the serial port (baud-rate, data bit length, parity bit, flow control, etc.) in the ezTCP.

The user can set the fields relating to IP address and the serial port through the ezConfig (a configuration utility through the network), Telnet, a serial port in Console mode, or AT command in ATC mode. To set the fields, the user also may temporarily set the IP address and access Telnet by the ARP method using the MAC address (hardware address).

	Items	Description
IP Address Fields	LOCAL IP ADDRESS	ezTCP IP Address
	SUBNET MASK	Subnet Mask
	GATEWAY	Gateway
	LOCAL PORT	Listening Port Number in Server Mode
	PEER IP ADDRESS	peer IP Address in Client Mode
	PEER PORT	Port Number in Client Mode
Serial Port	BAUD RATE	Baud Rate (bps)
	DATA BITS	Data Bit Length
	PARITY	Parity
	FLOW CTRL	Flow Control
Communication Mode	MUX TYPE	Communication Mode
Connection/ Connection Termination Event	WATER MARK	Minimum Number of Bytes to Attempt Connection/Transmission
	TIMEOUT	Connection Time Out
Using Configuration Method	EZCFG	ezConfig Function enable/disable
	TELNET	Remote Login Function enable/disable
	ARP	ARP function enable/disable
Floating IP	DHCP	DHCP(Dynamic Host Configuration Protocol)
Telnet Password	PASSWORD	Password when Performing Remote Login through Telnet
Authentication	EAPOL	Authentication Protocol (EZL-300W supports EAP-MD5 only)


### 3.2.1. ezConfig


The basic environment information of EZL-300W can be configured through console, telnet, DHCP, ARP, and ezConfig. This section takes an example of using ezConfig, which is a configuration program via wired/wireless LAN. For the other configuration modes, see the following sections. ezConfig can run on Microsoft Windows platform but may not operate on some of the existing operating systems. The screen below shows the initial screen of ezConfig:








Each button on ezConfig functions as follows:


 This button is used to search for all of the network-attached ezTCPs. The search results will be displayed on the MAC ADDRESS LIST box and you can select an item using a mouse or cursor as required. The value displayed on the box indicates the MAC ADDRESS of each ezTCP. The selected setup value of ezTCP will be displayed on the right side.

 You can see only the ezTCP configuration values if you press this button after entering the 6-digit hexadecimal number printed on the ezTCP main body in the MAC ADDRESS box. It is useful when there are too many ezTCPs attached to the network to search for one from the LIST box.

 This button is used to save the changed value in ezTCP after modifying the configuration. Make sure not to press this button during operating ezTCP since ezTCP will automatically be reset right after its environment setup value is saved. Otherwise, it may cause malfunction.

 This button is used to close ezConfig. You can also close it by pressing ESC key on the keyboard.

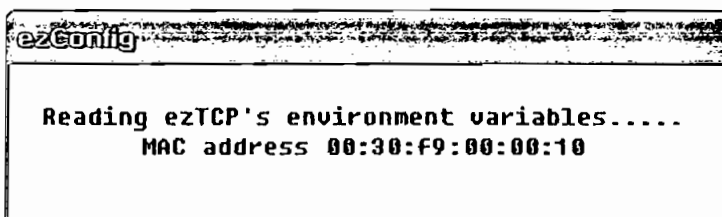
 ezTCP provides User Authentication function to prevent an unwanted person from modifying the configuration. The authentication process is performed through the password string verification. When entering or changing the password strings, you can use this button. Changing the ezTCP configuration details if a password has been entered requires the proper password to be entered in the PASSWORD field.

 This button is used to read a dynamic variable value during operating ezTCP. Pressing this button will display a new window, where the time-elapsd after the power is on, the current IP address, and the data throughput of the serial port are indicated. Double-clicking each item on the MAC ADDRESS LIST will carry out the same function.

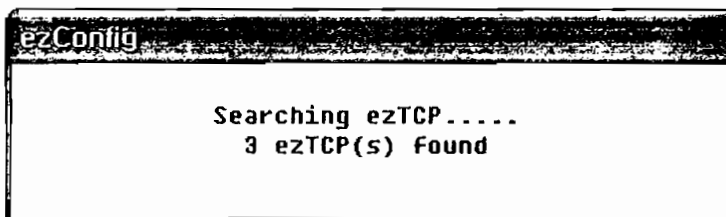
ezConfig can be used to change the IP address related items, the serial port setup value, the serial port operation mode, and how to setup ezTCP. This section describes these functions briefly. For more information, see the following sections.

The following example shows how to read and change ezTCP's basic functions. Try changing ezTCP setup value according to the following sequence:

- When the ezTCP power is turned on and the LAN cable is connected correctly, pressing [PROBE] or [READ] button will display the following window:



- If a network-attached ezTCP is detected, the following message will be displayed. If a message pops up indicating that there is no response from ezTCP, check that the power is turned on and the cable is connected correctly, then try pressing [PROBE] or [READ] button.



- If more than one ezTCP are detected, ezTCP's MAC ADDRESS will be displayed in the [MAC ADDRESS LIST] box on ezConfig. Check if the MAC ADDRESS displayed in the [MAC ADDRESS LIST] window corresponds to that printed on ezTCP main body. The following screen shows this process:

- Select T2S (0) in the [MUX TYPE] field on ezConfig. Setup [LOCAL IP ADDRESS], [LOCAL PORT], and the serial port settings according to a test environment as required.

When the setup is completed, pressing [WRITE] will save the changed values in ezTCP. If an error message pops up during storing the configuration, check that ezTCP is connected correctly, and then try again.

MAC ADDRESS		LOCAL IP ADDRESS				BAUD RATE	DATA BITS
00	30	f9	00	4a	2b	19200	8
MAC ADDRESS LIST		SUBNET MASK				PARITY	FLOW CTRL
00:30:F9:00:4a:2b		255 0 0 0				NONE	NONE
		GATEWAY				MUX TYPE	TIME OUT
		0 0 0 0				T2S(0)	0
		NAME SERVER				LOCAL PORT	PEER PORT
						1470	1470
		NAT IP ADDRESS				WATER MARK	OPTION
						0	
		PEER IP ADDRESS				PASSWORD	CHANGE PWD
		255 255 255 255					
<input checked="" type="checkbox"/> EZCFG	<input checked="" type="checkbox"/> ARP	<input type="checkbox"/> DHCP	<input type="checkbox"/> PPPoE	<input checked="" type="checkbox"/> TELNET	<input type="checkbox"/> EAPOL		
PROBE	READ	WRITE	PPPoE ID	STATUS	EXIT		

- In Windows' MS-DOS prompt window, check the IP address is set correctly by giving the PING command. If the ezTCP IP address is set correctly, the PING results will be displayed as follows. If a message, "Request timed out", is displayed, check that IP address setup value again.

```
C:\>ping a.b.c.d
```

```
Pinging a.b.c.d with 32 bytes of data:
```

```
Reply from a.b.c.d: bytes=32 time=1ms TTL=64
```

```
Reply from a.b.c.d: bytes=32 time=1ms TTL=64
```

```
Reply from a.b.c.d: bytes=32 time=1ms TTL=64
```

```
Reply from a.b.c.d: bytes=32 time=1ms TTL=64
```

### 3.2.2. Console

The ezTCP can be configured environment variables of the ezTCP through the console. The user can configure all environment variables including wireless LAN-related items and other variables relating to the IP address and the serial port through the console.

When the user enters into Console mode, the serial port operates in Console mode at 19200 bps.

The shell is also provided through Telnet connection, which means the same conditions as those of Console mode are provided.

The items, which can be set in Console mode and through Telnet, are network information (if), Telnet connection password (pwd), and ezTCP information (ezl). The command "env" is used to set up environment in the shell program. The factors such as "if", "pwd", and "ezl" can be used for "env". The following results are the output of the factory-set values.

**ANEXO 3**  
**ESTÁNDAR PCMCIA / JEIDA**

## 2. COMMON PIN DESCRIPTION

A number of pins have the same function for all three interfaces specified in this release: VCC, VPP[2::1], and GND. Several other pins are used for the same purpose in all three interfaces and have similar function in terms of providing information about card presence, card type, and VCC requirements. Signals whose function is specific to an interface are described in their respective sections.

### 2.1 Power and Ground Pins

Power and ground for all PC Card interfaces must be provided by the host system. PC Cards may not apply any voltage to VCC, VPP[2::1], or GND. Refer to the *Metaformat Specification* for information on describing the card's VCC and VPP[2::1] requirements in the Card Information Structure.

#### 2.1.1 VCC and GND Pins

The two 16-bit PC Card interfaces support 5 V, 3.3 V and X.X V VCC voltages while the CardBus PC Card interface supports 3.3 V, X.X V and Y.Y V VCC voltages. Deciding whether the socket hardware must support 5 V is a function of the interfaces being supported, the availability of 5 V in the system, and which cards need to be enabled. Sockets are not required to support 5V VCC operation. The connector places the two VCC pins (17 and 51) and four GND pins (1, 34, 35 and 68) at symmetrical positions on the connector.

The voltage level on VCC cannot be changed until the Card Information Structure (CIS) of the card has been read and other permissible values have been determined. If the CIS indicates the card could be operated at a different voltage level, the host can change to the new voltage level.

To change VCC, the host shall direct the socket to discharge the PC Card connector's VCC and VPP[2::1] to ground, then power-up the card at the new voltage. Further, care should be taken when dynamically changing the voltage applied to VCC or VPP[2::1] so that power supply shorts do not occur. The host must recognize that the card will retain no knowledge of the power-up at the previous VCC and all configuration and other initialization must be done following the second power-up. If any Card Detect pin is negated at any time, the host system must recognize that the card may have been replaced and repeat the entire power-up sequence.

#### 2.1.2 VPP1 and VPP2 Pins

The VPP[2::1] (pins 18 and 52) supply signals are used optionally on the card for PC Card operation. VPP[2::1] must be initially powered up at the voltage indicated by the voltage sense pins which means systems are required to be able to supply the VCC level on the VPP[2::1] pins. The voltage level on VPP[2::1] cannot be changed until the Card Information Structure (CIS) of the card has been read and other permissible values have been determined. The voltage applied to the VPP[2::1] pins of a card must never be greater than the VPP[2::1] level appropriate for the card. If the appropriate VPP[2::1] voltage for a card cannot be determined, the voltage applied to the VPP[2::1] pins must not exceed VCC.

Regardless of how PC Cards use VPP[2::1], the respective planes on the card must never be shorted together or shorted to VCC. The host is not required to account for shorted power planes in the design of its power supplies or power delivery schemes.

When the  $V_{PP}[2::1]$  value required by a card is unavailable in a system, the system may reject the card.

## 2.2 Interface Configuration Pins

The Card Detect pins,  $CD[2::1]\#$  or  $CCD[2::1]\#$  and Voltage Sense pins,  $VS[2::1]\#$  or  $CVS[2::1]$  are used by the host system to establish the presence/absence of a PC Card in a socket and the voltage requirements of the card. For CardBus PC Cards, these pins are also used to distinguish between 16-bit PC Card and CardBus PC Cards. Careful attention should be given to the following discussions since subtle, but very important, differences in the usage of these pins exist between the 16-bit PC Card and CardBus PC Card interfaces.

### 2.2.1 Card Detect Pins ( $CD[2::1]\#$ and $CCD[2::1]\#$ )

The Card Detect pins provide a means for sockets to detect PC Card insertion and removal events. These pins are at opposite ends of the connector to ensure a valid insertion (i.e. guarantees both sides of the card are firmly seated).

From the socket's perspective, the Card Detect pins function the same for all three interfaces (16-bit PC Card Memory-only, 16-bit PC Card I/O and Memory, and CardBus PC Card) – they are inputs pulled high through a resistor. The host socket interface circuitry shall provide a 10 K $\Omega$  or larger pull-up resistor to VCC on each of these signal pins. Host sockets shall only report valid insertions when both Card Detect pins are detected low ( $CD[2::1]\#$  or  $CCD[2::1]\#$ ). Failure to do so may cause electrical damage to PC Cards.

Cards implementing the 16-bit PC Card interface must connect  $CD1\#$  and  $CD2\#$  to ground internally on the PC Card causing the socket's inputs to be pulled low whenever a card is inserted. CardBus PC Cards also cause the socket's  $CCD[2::1]\#$  inputs to be pulled low upon insertion. (See 3.4 *Determining Card Type in CardBus PC Card Capable Sockets*.) However, CardBus PC Cards also use the  $CCD[2::1]\#$  pins in conjunction with  $CVS[2::1]$  to encode card type information. (See also *Figure 3-1 CCD[2::1]\# and CVS[2::1] Connections*.)

### 2.2.2 Voltage Sense Pins ( $VS[2::1]\#$ and $CVS[2::1]$ )

The Voltage Sense signals notify the socket of the card's VCC requirements for initial power up and configuration. (See also 3.3 *Graceful Rejection in 16-bit PC Card Only Sockets*.) CardBus PC Cards also use the  $CVS[2::1]$  pins in conjunction with  $CCD[2::1]\#$  to encode card type information. (See also 3.4 *Determining Card Type in CardBus PC Card Capable Sockets*.)

## 3. CARD TYPE DETECTION MECHANISM

The card interface (16-bit PC Card vs. CardBus PC Card) must be detected before the socket notifies Card Services of an insertion event. To initially power up a PC Card and determine its characteristics, VCC and VPP[2::1] must be at a voltage indicated by the Voltage Sense pins. This section describes how sockets determine the card interface and initial voltage requirements.

If the Card Information Structure (CIS) indicates that the card can operate at voltages other than the voltage at which it was initially powered up, the host system may change the card's VCC, and VPP[2::1] accordingly.

### 3.1 PC Card Encodings

This specification provides the ability to support VCC values of 5 V, 3.3 V, X.X V (where X.X V < 3.3 V), Y.Y V (where Y.Y V < X.X V), and various combinations of each. PC Cards must indicate the voltage(s) at which their CIS can be read by connecting the Card Detect and Voltage Sense pins. (See 2. *Common Pin Description*.) The CIS on a card shall be capable of being read at the VCC level indicated by the Voltage Sense pins. Any voltage combinations not listed in Table 3-1 are not supported (i.e., the 16-bit PC Card interface does not support Y.Y V operation and the CardBus PC Card interface does not support 5 V CardBus PC Card operation; a CardBus PC Card socket may support 5 V 16-bit PC Card operation..)

PC Cards must implement one of two physical keys shown, 5 V or Low Voltage (LV) key. (See the *Physical Specification*.) Any card capable of having its CIS read at 5 V shall be keyed with the 5 V key. Any card not capable of having its CIS read at 5 V shall be keyed with the LV key.



Table 3-1 Card Detect and Voltage Sense Connections

CD2#/CCD2# (pin 67)	CD1#/CCD1# (pin 36)	VS2#/CVS2 (pin 57)	VS1#/CVS1 (pin 43)	Card Type		
				Key	Interface	Voltage
ground	ground	open	open	5V	16-bit PC Card	5V
ground	ground	open	ground	5V	16-bit PC Card	5V and 3.3V
ground	ground	ground	ground	5V	16-bit PC Card	5V, 3.3V and X.XV
ground	ground	open	ground	LV	16-bit PC Card	3.3V
ground	connect to CVS1	open	connect to CCD1#	LV	CardBus PC Card	3.3V
ground	ground	ground	ground	LV	16-bit PC Card	3.3V and X.XV
connect to CVS2	ground	connect to CCD2#	ground	LV	CardBus PC Card	3.3V and X.XV
connect to CVS1	ground	ground	connect to CCD2#	LV	CardBus PC Card	3.3V, X.XV and Y.YV
ground	ground	ground	open	LV	16-bit PC Card	X.XV
connect to CVS2	ground	connect to CCD2#	open	LV	CardBus PC Card	X.XV
ground	connect to CVS2	connect to CCD1#	open	LV	CardBus PC Card	X.XV and Y.YV
connect to CVS1	ground	open	connect to CCD2#	LV	CardBus PC Card	Y.YV
ground	connect to CVS1	ground	connect to CCD1#	reserved		
ground	connect to CVS2	connect to CCD1#	ground	reserved		

## 3.2 Socket Key Selection

A 5 V only socket shall be keyed with the 5 V key which allows only cards with the 5 V key to be inserted. Such a socket shall always apply initial VCC at 5 V and need not sense the VS[2::1]# signals. Note that this type of socket is restricted to the 16-bit PC Card interface since 5 V only CardBus PC Cards are not supported.

Sockets which provide 3.3 V or lower voltage VCC levels must implement the Low Voltage (LV) socket. (See the *Physical Specification*.) This key allows the insertion of both 5 V keyed and Low Voltage keyed cards. A socket capable of accepting a card with a Low Voltage key must implement cold insertion (i.e., ensure that VCC and VPP[2::1] are removed from the socket and signals are placed in the High-Z state, without software intervention, before the next PC Card is inserted). Low voltage sockets must only treat the Voltage Sense pins as valid when both Card Detect pins are asserted low. Failure to require both Card Detect pins to be low may result in falsely decoding a card's VCC requirements.

If the Voltage Sense pins indicate a VCC value the socket is capable of providing, the socket shall allow the application of that VCC level to the card. If Voltage Sense pins indicate values of VCC the socket is not capable of providing, the card inputs shall not be driven, the card shall not be powered, and the user may be notified.

## 3.3 Graceful Rejection in 16-bit PC Card Only Sockets

Sockets which do not support the CardBus PC Card interface must tie their VS1# and VS2# inputs high through a pull-up resistor. These sockets may assume that all valid insertions (i.e., both Card Detect pins low) are 16-bit PC Card interface cards and ignore the interrogation protocol required

for the CardBus PC Card interface. This is because a CardBus PC Card always ties one Card Detect pin to a Voltage Sense pin instead of to ground causing it to only pull one of the Card Detect inputs low. If a 16-bit PC Card only socket senses only one Card Detect input low, the user may be notified that one of the following conditions exists:

1. A card has not been inserted correctly or completely, or
2. The card inserted is of a type not supported by this socket (i.e., CardBus PC Card).

### 3.4 Determining Card Type in CardBus PC Card Capable Sockets

Since a valid CardBus PC Card insertion (i.e., CCD[2::1]# pins are sampled low at the same time after having been debounced) can only be detected when both CVS[2::1] pins are low, sockets must always drive their CVS[2::1] outputs low when PC Card removal occurs.

Once a valid insertion is detected and before power is applied, the socket must interrogate the PC Card to determine if it is a CardBus PC Card or 16-bit PC Card. This interrogation consists of determining which CCD[2::1]# and CVS[2::1] pins are shorted to ground or each other and which are not connected by alternately driving each CVS[2::1] output high and monitoring what happens to the CCD[2::1]# and CVS[2::1] inputs. At the completion of this interrogation, the socket must again drive the CVS[2::1] pins low. This means the socket must drive the CVS[2::1] pins low at all times except when determining the card type and VCC requirements.

An example of how the socket's state machine, CCD[2::1]# and CVS[2::1] pins, and the connector might be connected is provided. (See *Figure 3-1 CCD[2::1]# and CVS[2::1] Connections*.) Note that the CVS[2::1] resistors could be integrated into their drivers so that each only consumes a single pin on the socket controller.

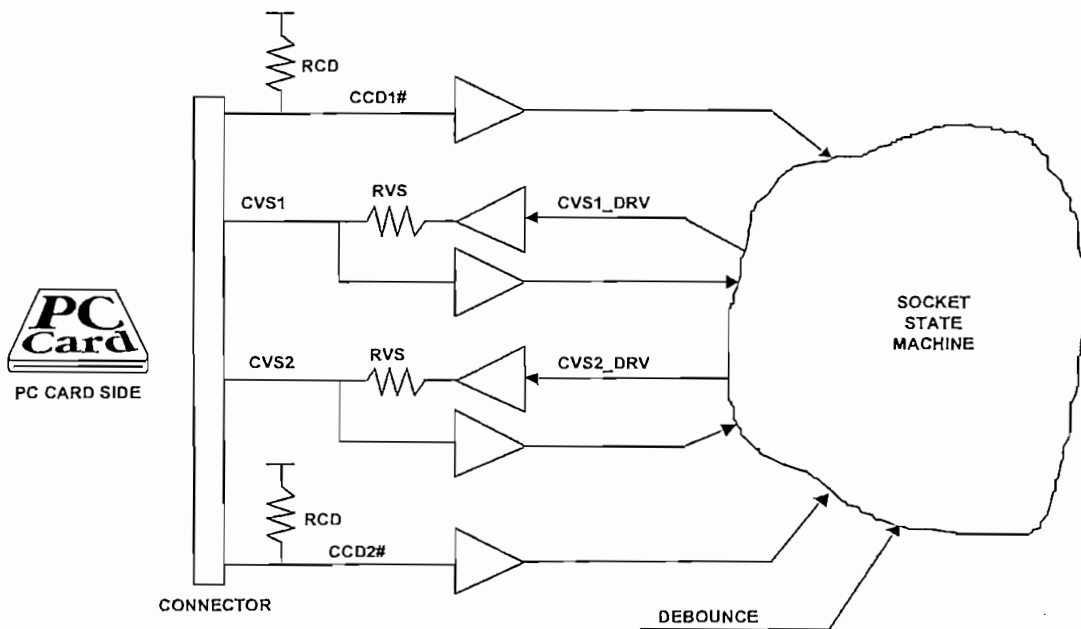


Figure 3-1 CCD[2::1]# and CVS[2::1] Connections

A series of steps is required to identify and configure a card upon insertion into a CardBus PC Card capable socket. (See 5.5.4.6.1 *Card Insertion*.)

## 3. CARD DIMENSIONS

There are six types of PC Cards in this specification. Three types of full-size PC Cards: Type I, Type II, and Type III, and three types of Small PC Cards: also Type I, Type II, and Type III.

Connector location and pin numbers for Type I, Type II, and Type III full-size PC Cards are shown in *Figure 11-2: TYPE I PC Card Package Dimensions*, *Figure 11-3: TYPE II PC Card Package Dimensions*, and *Figure 11-4: Type III PC Card Package Dimensions*. Connector location and pin numbers for Type I, Type II, and Type III Small PC Cards are shown in *Figure 11-5: Small PC Card Type I Package Dimensions*, *Figure 11-6: Small PC Card Type II Package Dimensions*, and *Figure 11-7: Small PC Card Type III Package Dimensions*.

PC Card polarization technique and dimensions are also shown in *Figure 11-2*, *Figure 11-3*, and *Figure 11-4* for full-size PC Cards, and in *Figure 11-5*, *Figure 11-6*, and *Figure 11-7* for Small PC Cards.

PC Cards must be opaque (non see-through).

UNLESS OTHERWISE SPECIFIED, ALL DIMENSIONS ARE IN MILLIMETERS (MM). DIMENSIONS SHOWN DO NOT INCLUDE WARPAGE ALLOWANCES.

### 3.1 Write Protect Switch (WPS)

The WPS, if installed, shall be located at the locations shown in *Figure 11-2*, *Figure 11-3*, *Figure 11-5* and *Figure 11-6*.

The write-protected position of the WPS shall be the far-right position, and shall be indicated by an arrow and the words "Write Protect" or "Protect" or "WP". The arrow and indication may be on the end of the PC Card, as shown in *Figure 11-2*, *Figure 11-3*, *Figure 11-5*, and *Figure 11-6* or on the bottom cover, or on both the end and bottom cover.

If a WPS is used, it is recommended that it pass all requirements, as applicable, in PC Card Environmental. It is also recommended the WPS perform as specified for a minimum of 100 (Write Protect/Write Enable) cycles.

### 3.2 Battery Location

The battery, if installed, shall be located at the locations shown in *Figure 11-2*, *Figure 11-3*, *Figure 11-5* and *Figure 11-6*. The battery holder, if installed, shall be designed so that the positive (+) side of the battery faces Surface A.

### 3.3 Labeling (Marking)

The thickness of labeling, if used, shall not cause the PC Card to exceed the thickness specified in *Figure 11-2*, *Figure 11-3*, *Figure 11-4*, *Figure 11-5*, *Figure 11-6*, or *Figure 11-7*.

The label, if used, must withstand all environmental test specified the PC Card Environmental Section.

The PC Card logo may be displayed by member company manufacturers as authorized.

## 4. CONNECTOR

The specified PC Card interconnect system shall be a 68-position, 2-piece pin-and-socket. The socket contacts shall be within the PC Card connector.

### 4.1 Card Connector

The socket contacts are located on the full-size PC Card as shown in *Figure 11-2: TYPE I PC Card Package Dimensions*, *Figure 11-3: TYPE II PC Card Package Dimensions*, *Figure 11-4: Type III PC Card Package Dimensions*, *Figure 11-8: Type I PC Card (3D)*, *Figure 11-9: Type II PC Card (3D)*, and *Figure 11-10: Type III PC Card (3D)*. The socket contacts for Small PC Card are located on the Small PC Card as shown in *Figure 11-5: Small PC Card Type I Package Dimensions*, *Figure 11-6: Small PC Card Type II Package Dimensions*, and *Figure 11-7: Small PC Card Type III Package Dimensions*.

The PC Card connector socket shall be configured as shown in *Figure 11-11: Full-size PC Card Connector Socket* or *Figure 11-12: Small PC Card Connector Socket*.

The PC Card connector socket layout shall match the host pin-connector layout as shown in *Figure 11-15: Full-size PC Card Pin Connector* or *Figure 11-16: Small PC Card Pin Connector*.

The CardBus PC Card connector (or PCMCIA/JEIDA approved equivalent) shall contain a top side planar, electrically conductive, ground plate (*Figure 11-40: CardBus PC Card Recommended Connector Grounding Interface Dimensions*) with eight (8) raised dimples 0.5 mm in height. This ground plate is connected to the PC Card electrical system ground (*Figure 11-41: CardBus PC Card Recommended PCB Footprint*), must be isolated from chassis ground, and shall meet Electrostatic Discharge requirements as specified in the Electrostatic Discharge Section.

### 4.2 Host Connector

The full-size PC Card host pin connector shall be a 68-pin connector with opening, polarization, keying, and pin location as shown in *Figure 11-15: Full-size PC Card Pin Connector*. The host connector-pin configuration is shown in *Figure 11-17: Full-size PC Card Host Connector, Pin Contacts*, and the host-pin lengths are shown in *Figure 11-13: Full-size PC Card Pin/Socket Contact Length with Wipe* and *Figure 11-17* and pin type, length, and number in *Table 4-1: Host Connector Pin Configuration*.

The Small PC Card host pin connector shall be a 68-pin connector with opening, polarization, keying, and pin location as shown in *Figure 11-16: Small PC Card Pin Connector*. The host connector-pin configuration is shown in *Figure 11-18: Small PC Card Host Connector, Pin Contacts*, and the host-pin lengths are shown in *Figure 11-14: Small PC Card Pin/Socket Contact Length with Wipe* and *Figure 11-18*, and pin type, length, and number in *Table 4-1: Host Connector Pin Configuration*.

Table 4-1: Host Connector Pin Configuration

Pin Type	Pin Length (L) $\pm 0.10$ mm	Pin Number
Detect	3.50	36, 67
General	4.25	All Other Pins
Power/Ground	5.00	1, 17, 34, 35, 51, 68

The outermost plating of socket and pin contact area shall be gold or other plated materials compatible with gold and shall meet the requirements specified in the Connector Reliability and Connector Durability Sections.

The host pin connector for CardBus PC Card applications (or PCMCIA/JEIDA approved equivalent) shall contain a top side planar electrically conductive, ground shroud (*Figure 11-42: CardBus PC Card Recommended Host Connector Grounding Interface Dimensions*) with eight (8) fingers having an effective minimum contact wipe length of 3.6 mm when mating with the CardBus PC Card Connector (*Figure 11-45: CardBus PC Card Reference Shrouded Connector* and *Figure 11-46: CardBus PC Card Reference Shrouded Connector (Stacked Connector)*). These eight (8) fingers shall be recessed within the host pin connector protective dielectric shroud providing a 0.254 mm minimum air gap when mating with 16-bit PC Cards.

The recommended host connector PCB footprints for: the right angle connector (*Figure 11-19: Recommended Right Angle Connector PCB Footprint*), the straight connector (*Figure 11-20: Recommended Straight Connector PCB Footprint*), two row surface mount connector (*Figure 11-21: Recommended Two Row Surface Mount Connector PCB Footprint*), and one row surface mount connector (*Figure 11-22: Recommended One Row Surface Mount Connector PCB Footprint*) are shown without mounting or hardware hole locations.

The recommended CardBus PC Card host connector PCB footprints for the right angle connector (*Figure 11-43: CardBus PC Card Recommended Right Angle PCB Footprint* and *Figure 11-44: CardBus PC Card Recommended Right Angle PCB Footprint (Stacked)*) is shown with mounting hole locations. The host connector ground shroud is connected to host system electrical ground signals and must be isolated from chassis ground.

The interconnect system shall pass all requirements of the Connector Reliability Section and the Connector Durability Section.

If a connector ejector mechanism is used, it is recommended the ejector mechanism pass all requirements for reliability and durability, as applicable, in Section 7 *Connector Reliability* and Section 8 *Connector Durability*.

## 7. CONNECTOR RELIABILITY

The interconnect system as specified in the Connector Section shall meet or exceed all reliability test requirements of this subsection. Unless otherwise specified, all test and measurements shall be made at:

Temperature	15°C to 35°C
Air pressure	86 to 106 kPa
Relative humidity	25% to 85%

If conditions must be closely controlled in order to obtain reproducible results, the parameters shall be:

Temperature	23°C $\pm$ 1°C
Air pressure	86 to 106 kPa
Relative humidity	50% $\pm$ 2%

See Section 7.5 *Approved Test Procedures* for approved test procedures.

### 7.1 Mechanical Performance

The interconnect system mechanical performance is specified in the following sections.

#### 7.1.1 Office Environment

STANDARD	TESTING
Guaranteed number of insertions/ejections = 10,000 minimum	See <i>Office Environment</i> Section, EIA 364-B Class 1.1

#### 7.1.2 Harsh Environment

STANDARD	TESTING
Guaranteed number of insertions/ejections = 5,000 minimum	See <i>Harsh Environment</i> Section, EIA 364-B Class 1.3

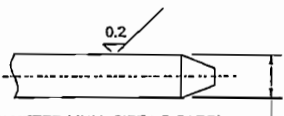
#### 7.1.3 Total Insertion Force

STANDARD	TESTING
39.2 N maximum	Insert at speed of 25 mm/minute

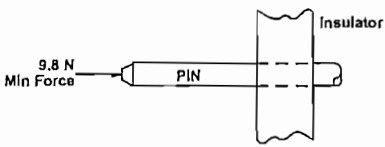
#### 7.1.4 Total Pulling Force

STANDARD	TESTING
6.67 N minimum and 39.2 N maximum	Extract at speed of 25 mm/minute

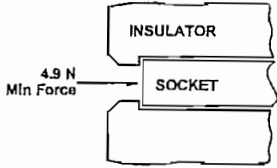
### 7.1.5 Single Pin Pulling Force

STANDARD	TESTING
<p>0.098 N minimum initial value</p>  <p>0.420 ±0.005 mm DIAMETER (FULL-SIZE PC CARD) 0.350 ±0.005 mm DIAMETER (SMALL PC CARD)</p> <p>Gauge: Material - Tool making steel Hardness - HRC = 50 to 55</p>	<p>Pull the gauge pin shown to left at speed of 25 mm/minute</p> <p>Gauge pin's surface must be wiped clean of dirt and lubrication oil</p>

### 7.1.6 Single Pin Holding Force

STANDARD	TESTING
<p>Pin shall not push out of the insulator when 9.8 N minimum force is applied to the pin</p>	<p>Push pin on the axis at speed of 25 mm/minute with 9.8 N minimum force while holding insulator rigid.</p> 

### 7.1.7 Single Socket Holding Force

STANDARD	TESTING
<p>Socket shall not be dislodged or damaged when 4.9 N minimum force is applied to the socket</p>	<p>Push socket on the axis with 4.9 N minimum force at a speed of 25 mm/minute while holding insulator rigid</p> 

### 7.1.8 Vibration and High Frequency

STANDARD	TESTING
<p>a. No mechanical damage shall occur on the parts b. Must not cause current interruption greater than 100 ns</p>	<p>147 m/s<sup>-1</sup> (15G) peak amplitude, 10 Hz to 2000 Hz, 20 minute sweep, 12 cycles per axis, 3 axis. See Figure 11-25: Connector Shock &amp; Vibration Test Fixture</p>

### 7.1.9 Shock

STANDARD	TESTING
a. No mechanical damage shall occur on the parts b. Must not cause current interruption greater than 100 ns	Acceleration 490 m/s <sup>2</sup> (50G) Standard holding time 11 ms, Semi-sine wave Velocity change 3.44 m/s (11.3 ft/s) See <i>Figure 11-25: Connector Shock &amp; Vibration Test Fixture</i>

### 7.1.10 Pin Connector Inverse Insertion

STANDARD	TESTING
Measurements shall be made by putting aside a gauge card to a guide portion of two different pin connectors and right side measurement at another.	Maximum insertion force: 58.8 N or more Travel after contact to key of pin connector: 5 mm or less Since this requirement is for a single pin connector, inverse insertion test method in Section 9.3.11 <i>Card Inverse Insertion</i> is applied to test the performance at actual use.

## 7.2 Electrical Performance

The interconnect system electrical performance is specified as follows.

### 7.2.1 Contact Resistance (low level)

STANDARD	TESTING
a. Initially 40 mΩ maximum b. After test 20 mΩ maximum change	Open voltage 20 mV Test current 1 mA a. Measure and record the initial resistance (R <sub>i</sub> ) of the separate connector contact interface. See <i>Figure 11-47: Contact Resistance Measurement</i> $R_i \leq 40 \text{ m}\Omega$ b. Measure and record resistance after test (R <sub>f</sub> ) of the connector system. Resistance value after test: $R_f = R_i \pm 20 \text{ m}\Omega$

### 7.2.2 Withstanding Voltage

STANDARD	
a. No shorting or other damages when 500 V <sub>rms</sub> AC is applied for 1 minute b. Current leakage 1 mA maximum	

### 7.2.3 Insulation Resistance

STANDARD	TESTING
a. Initially 1000 MΩ minimum b. After test 100 MΩ minimum	Measure within 1 minute after applying 500V DC



## 7.2.4 Current Capacity

STANDARD	TESTING
0,5 A per contact	Based upon 30°C rise above ambient temperature

## 7.2.5 Insulation Material

STANDARD	
Flame retardant material will not burn nor support combustion	

## 7.2.6 Ground Return Inductance

Note: This requirement applies to CardBus PC Card applications.

STANDARD	TESTING
18,0 nH maximum Inductance @ 1 MHz applies to both single and stacked configurations	Low level inductance

## 7.3 Environmental Performance

### 7.3.1 Operating Environment

STANDARD
Operating Temperature: -20°C to +60°C Relative humidity: 95% maximum (non-condensing)

### 7.3.2 Storage Environment

STANDARD
Storage Temperature: -40°C to +70°C Relative humidity: 95% maximum (non-condensing)

## 7.4 Environmental Resistance

### 7.4.1 Moisture Resistance

STANDARD	TESTING
Per <i>Contact Resistance (low level)</i> Section, Part b Per <i>Insulation Resistance</i> Section, Part b	Temperature Cycling (excluding vibration); 10 cycles (1 cycle = 24 hours) with connectors engaged

### 7.4.2 Thermal Shock

STANDARD	TESTING
No physical damage shall occur during testing Per <i>Contact Resistance (low level)</i> Section, Part b Per <i>Insulation Resistance</i> Section, Part b	-55°C to +85°C 5 minute transition time (max) 5 cycles (1 cycle = 1 hour) with connectors engaged

### 7.4.3 Durability (High Temperature)

STANDARD	TESTING
Per <i>Contact Resistance (low level)</i> Section, Part b	85°C, 250 hours with connectors engaged Exclude load and insulation resistance measurements

### 7.4.4 Cold Resistance

STANDARD	TESTING
Per <i>Contact Resistance (low level)</i> Section, Part b	-55°C, 96 hours with connectors engaged

### 7.4.5 Humidity (Normal Condition)

STANDARD	TESTING
Per <i>Contact Resistance (low level)</i> Section, Part b Per <i>Insulation Resistance</i> Section, Part b	Steady State 40°C, 90 to 95% RH 96 hours with connectors engaged

### 7.4.6 Hydrogen Sulfide

STANDARD	TESTING
Per <i>Contact Resistance (low level)</i> Section, Part b	3 PPM hydrogen sulfide 40°C, approx. 80% RH 96 hours, with connectors engaged

# 11. FIGURES

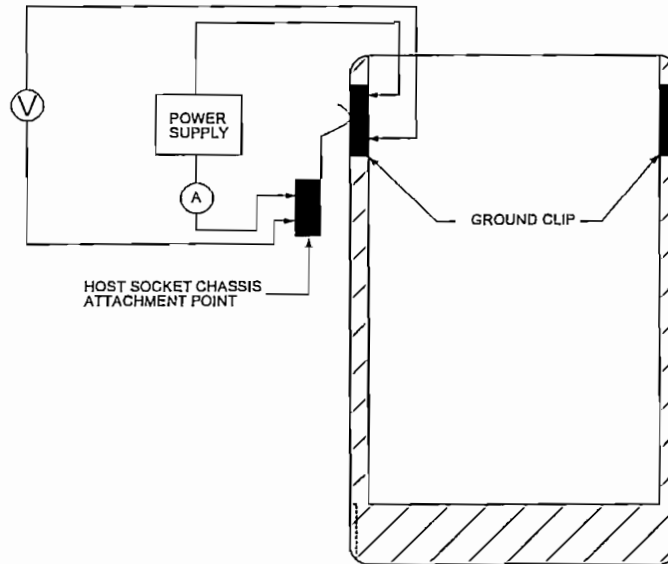
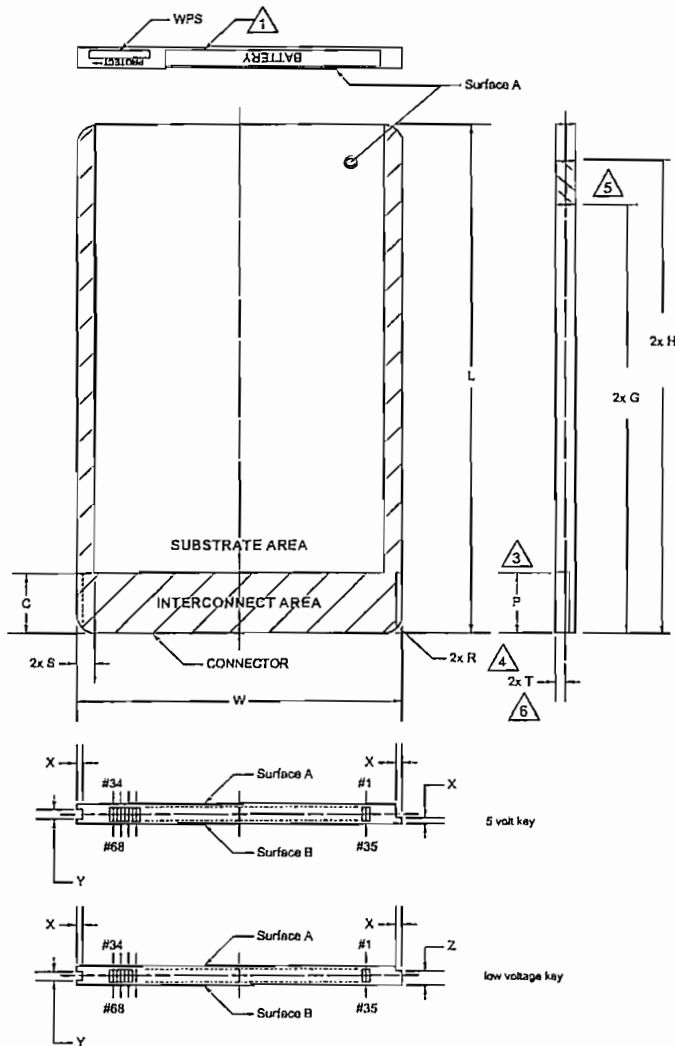


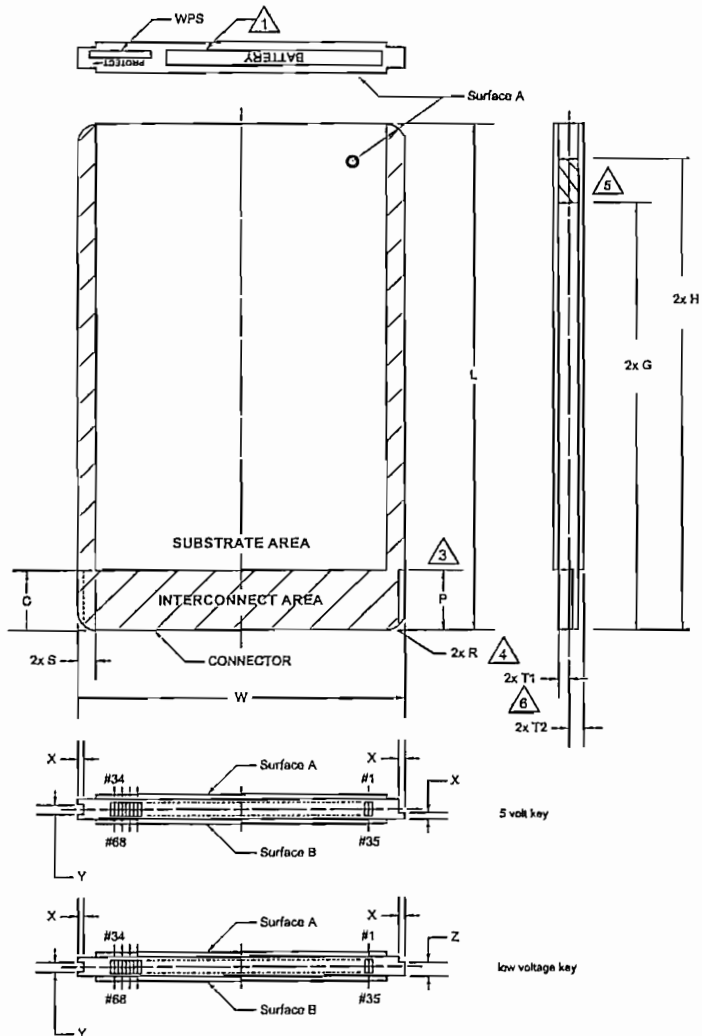
Figure 11-1: Grounding/EMI Clips Contact Resistance Measurement



C MIN	L ± 0.20	P MIN $\triangle$ 3	R ± 0.10 $\triangle$ 4	S MIN	T $\triangle$ 5	W ± 0.10	X ± 0.05	Y ± 0.05	Z ± 0.05	G ± 0.60	H ± 0.60
10.0	85.60	10.0	0.60	3.0	1.65	54.00	1.00	1.60	2.10	65.60	79.60

- $\triangle$  1 RECOMMENDED BATTERY LOCATION. THE BATTERY HOLDER SHOULD BE DESIGNED SO THAT THE POSITIVE SIDE OF THE BATTERY IS UP (TOWARD SURFACE A)
- 2 THE PC CARD SHALL BE OPAQUE (NON SEE THRU)
- $\triangle$  3 POLARIZATION KEY LENGTH
- $\triangle$  4 DIMENSION R CORNER RADIUS
- $\triangle$  5 GROUND CLIP LOCATION
- $\triangle$  6 FOR CARDBUS PC CARDS DIMENSION T IS INCREASED BY  $0.50 \pm 0.05$  mm OVER DIMPLES  
(REFER TO Figure 11-40: CardBus PC Card Recommended Connector Grounding Interface Dimensions)  
INTERCONNECT AREA TOLERANCE =  $\pm 0.05$  mm  
SUBSTRATE AREA TOLERANCE =  $\pm 0.10$  mm

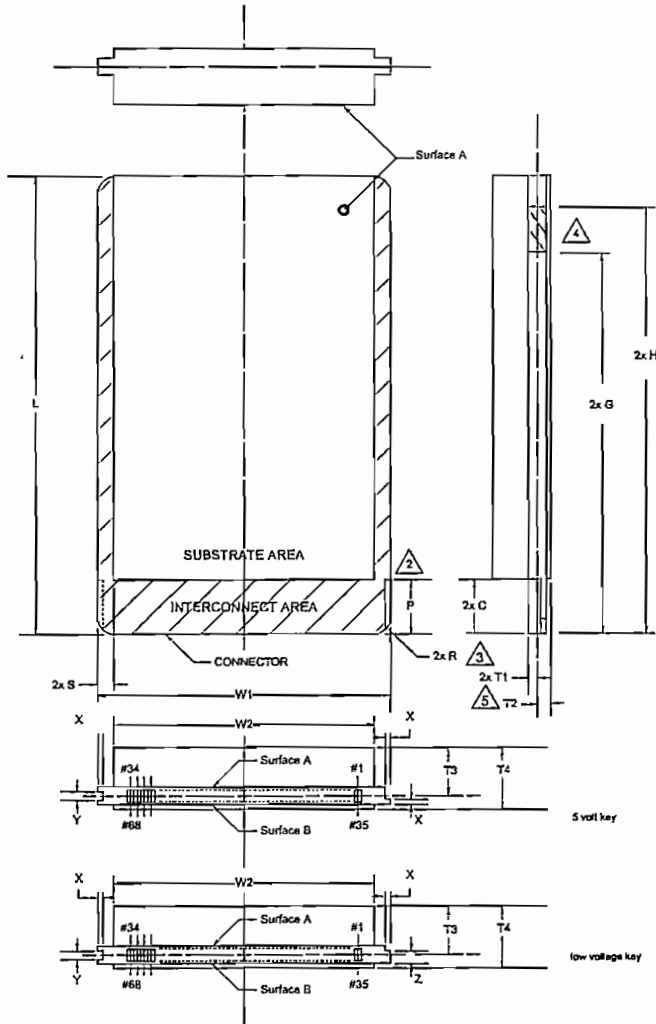
Figure 11-2: TYPE I PC Card Package Dimensions



C MIN	L ± 0.20	P MIN	R ± 0.10	S MIN	T1 ± 0.05	T2 MAX	W ± 0.10	X ± 0.05	Y ± 0.05	Z ± 0.05	G ± 0.60	H ± 0.60
10.0	85.60	10.0	0.60	3.0	1.65	2.50	54.00	1.00	1.60	2.10	65.60	79.60

- 1 RECOMMENDED BATTERY LOCATION. THE BATTERY HOLDER SHOULD BE DESIGNED SO THAT THE POSITIVE SIDE OF THE BATTERY IS UP (TOWARD SURFACE A)
- 2 THE PC CARD SHALL BE OPAQUE (NON SEE THRU)
- 3 POLARIZATION KEY LENGTH
- 4 DIMENSION R CORNER RADIUS
- 5 GROUND CLIP LOCATION
- 6 FOR CARDBUS PC CARDS DIMENSION T1 IS INCREASED BY 0.50 ± 0.05 mm OVER DIMPLES (REFER TO Figure 11-40: CardBus PC Card Recommended Connector Grounding Interface Dimensions)

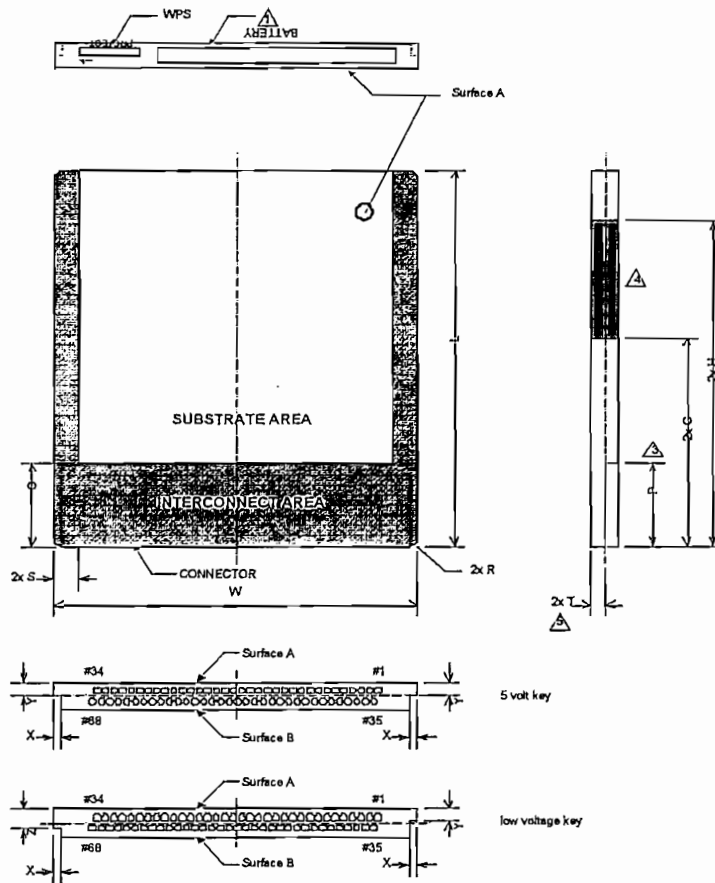
Figure 11-3: TYPE II PC Card Package Dimensions



C MIN	L ± 0.20	P MIN <sup>2</sup>	T1 ± 0.05 <sup>2</sup>	T2 MAX	T3 MAX	T4 REF	G ± 0.60
10.0	85.60	10.0	1.65	2.50	8.00	10.50	65.60
W1 ± 0.10	W2 MAX	X ± 0.05	Y ± 0.05	Z ± 0.05	R ± 0.10 <sup>3</sup>	S MIN	H ± 0.60
54.00	51.0	1.00	1.60	2.10	0.60	1.50	79.60

- 1 THE PC CARD SHALL BE OPAQUE (NON SEE THRU)
- <sup>2</sup> POLARIZATION KEY LENGTH
- <sup>3</sup> DIMENSION R CORNER RADIUS
- <sup>4</sup> GROUND CLIP LOCATION
- <sup>5</sup> FOR CARDBUS PC CARDS DIMENSION T1 IS INCREASED BY 0.50 ± 0.05 mm OVER DIMPLES  
(REFER TO Figure 11-40; CardBus PC Card Recommended Connector Grounding Interface Dimensions)

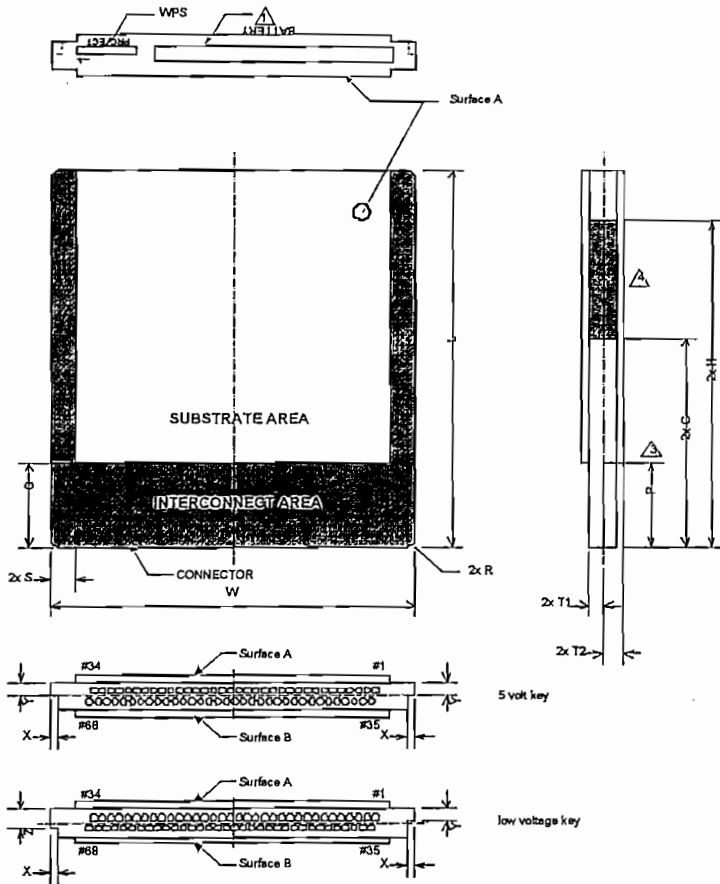
Figure 11-4: Type III PC Card Package Dimensions



C MIN	L ± 0.2	P MIN	R ± 0.10	S MIN	T $\Delta$	W $\Delta$	X ± 0.05	Y ± 0.05	Z ± 0.05	G ± 0.60	H ± 0.60
10.0	45.00	10.0	0.60	3.0	1.65	42.80	1.00	1.55	2.40	25.00	39.00

- $\Delta$  1 RECOMMENDED BATTERY LOCATION. THE BATTERY HOLDER SHOULD BE DESIGNED SO THAT THE POSITIVE SIDE OF THE BATTERY IS UP (TOWARD SURFACE A)
- 2 THE PC CARD SHALL BE OPAQUE (NON SEE THRU)
- $\Delta$  3 POLARIZATION KEY LENGTH
- $\Delta$  4 GROUND CLIP LOCATION
- $\Delta$  5 INTERCONNECT AREA TOLERANCE = ± 0.05 mm  
SUBSTRATE AREA TOLERANCE = ± 0.10 mm
- $\Delta$  6 TOLERANCE OF ENGAGEMENT AREA C = +0.10/-0.05 mm  
TOLERANCE OF OTHER AREA = ± 0.10 mm

Figure 11-5: Small PC Card Type I Package Dimensions

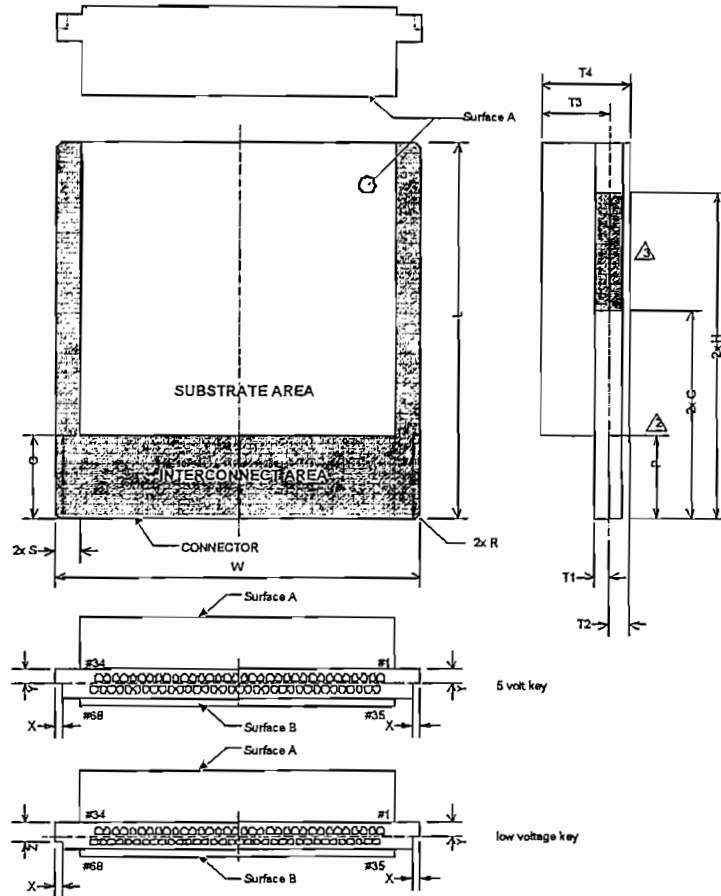


C MIN	L ± 0.20	P MIN	R ± 0.10	S MIN	T1 ± 0.05	T2 MAX	W $\triangle$	X ± 0.05	Y ± 0.05	Z ± 0.05	G ± 0.80	H ± 0.60
10.0	45.00	10.0	0.60	3.0	1.65	2.50	42.80	1.00	1.55	2.40	25.00	39.00

- $\triangle$  RECOMMENDED BATTERY LOCATION. THE BATTERY HOLDER SHOULD BE DESIGNED SO THAT THE POSITIVE SIDE OF THE BATTERY IS UP(TOWARD SURFACE A)
- 2 THE PC CARD SHALL BE OPAQUE(NON SEE THRU)
- $\triangle$  POLARIZATION KEY LENGTH
- $\triangle$  GROUND CLIP LOCATION
- $\triangle$  TOLERANCE OF ENGAGEMENT AREA C = +0.10/-0.05 mm  
TOLERANCE OF OTHER AREA = ± 0.10 mm

Figure 11-6: Small PC Card Type II Package Dimensions





C MIN	$L \pm 0.20$	P MIN	$R \pm 0.10$	S MIN	$T1 \pm 0.05$	T2 MAX	T3 MAX	T4 REF
10.0	45.00	10.0	0.60	3.0	1.65	2.50	8.00	10.50
$W \triangle 4$	$X \pm 0.05$	$Y \pm 0.05$	$Z \pm 0.05$	$G \pm 0.60$	$H \pm 0.60$			
42.80	1.00	1.55	2.40	25.00	39.00			

- 1 THE PC CARD SHALL BE OPAQUE(NON SEE THRU)
- $\triangle 2$  POLARIZATION KEY LENGTH
- $\triangle 3$  GROUND CLIP LOCATION
- $\triangle 4$  TOLERANCE OF ENGAGEMENT AREA  $C = +0.10/-0.05$  mm  
TOLERANCE OF OTHER AREA =  $\pm 0.10$  mm

Figure 11-7: Small PC Card Type III Package Dimensions

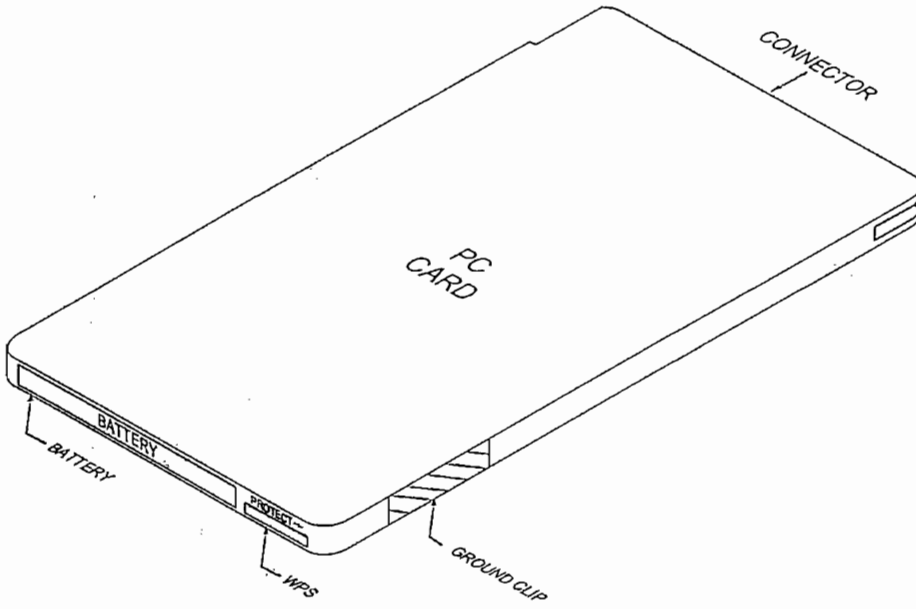


Figure 11-8: Type I PC Card (3D)

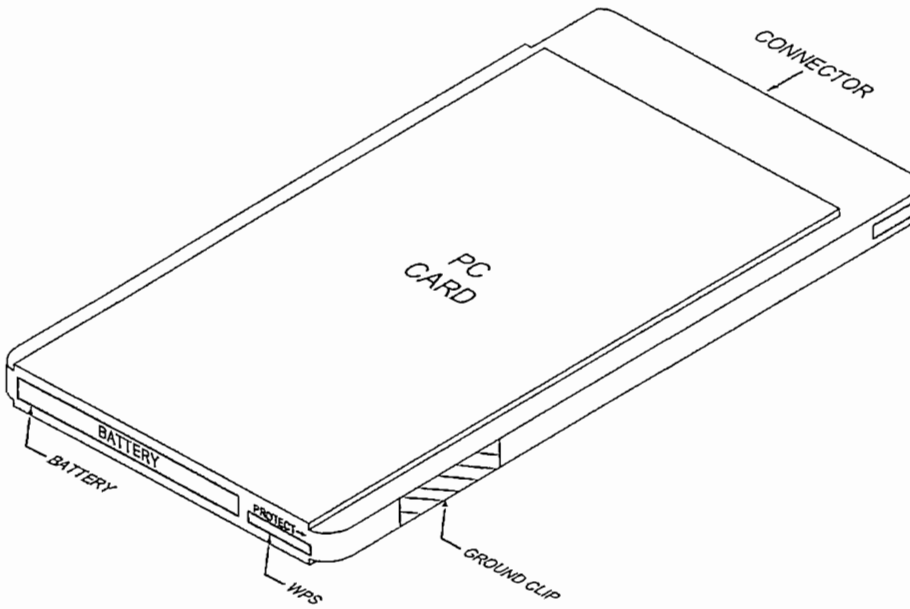


Figure 11-9: Type II PC Card (3D)

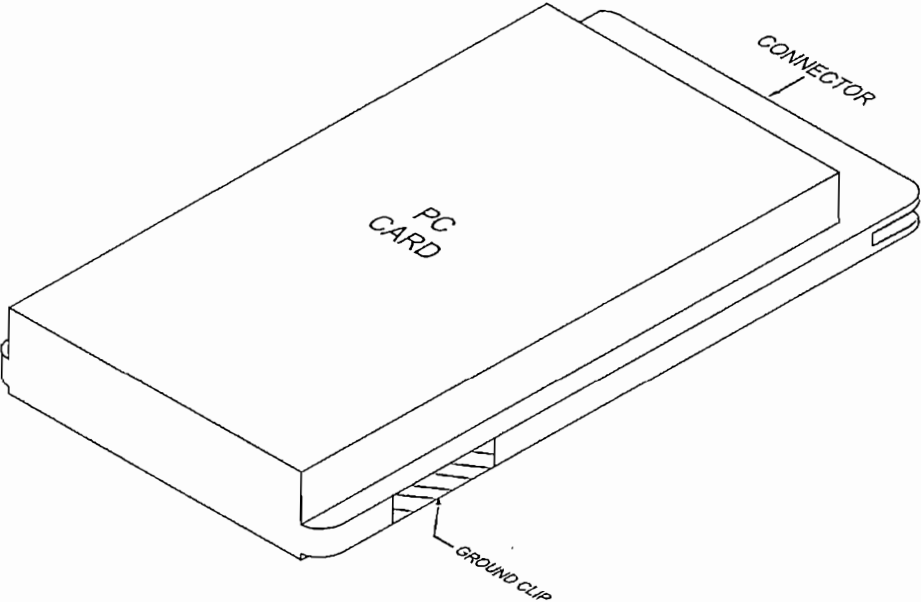


Figure 11-10: Type III PC Card (3D)

## 2.3 Metaformat Architecture

### 2.3.1 Basic Tuple Format and Tuple Chain Structure

The Card Information Structure is one or more chains (or linked lists) of data blocks or tuples. Longlink and linktarget tuples are used to connect chains (See 3.1 *Control Tuples*.) All tuples have the format shown below.

Table 2-1 Basic Tuple Format

Byte	7	6	5	4	3	2	1	0
0	TPL_CODE Tuple code: CISTPL_XXX.							
1	TPL_LINK Offset to next tuple in chain. This is the number of bytes in the tuple body. (n)							
2..(n + 2)	The tuple body. (n bytes)							

Byte 0 of each tuple contains a tuple code. A tuple code of FFH is a special mark indicating that there are no more tuples in the chain. Byte 1 of each tuple contains a link to the next tuple in the chain. If the link field is zero, then the tuple body is empty. If the link field of a 16-bit PC Card contains FFH, then this tuple is the last tuple in its chain.

There are two ways of marking the end of a tuple chain for 16-bit PC Cards: a tuple code of FFH, or a tuple link of FFH. There is only one way of marking the end of the tuple chain for CardBus PC Cards: A tuple code of FFH. (See also 2.3.8 *16-bit PC Card Tuple Chain Processing* and 2.3.9 *CardBus PC Card Tuple Chain Processing*.)

The use of an FFH link value is allowed in 16-bit PC Cards for backward compatibility, but it is recommended to use the End of Chain tuple. System software must use the link field to validate tuples. No 16-bit PC Card tuple can be longer than 257 bytes: 1 byte *TPL\_CODE* + 1 byte *TPL\_LINK* + FFH byte tuple body (and this 257 byte tuple ends the chain). No CardBus PC Card tuple can be longer than 256 bytes: 1 byte *TPL\_CODE* + 1 byte *TPL\_LINK* + FEH byte tuple body. Some tuples provide a termination or stop byte that marks the end of the tuple. In this case, the tuple can effectively be shorter than the value implied by its link field. However, software must not scan beyond the implied length of the tuple, even if a termination byte has not been seen.

### 2.3.2 Byte Order Within Tuples

Within tuples, all multi-byte numeric data shall be recorded in little-endian order. That is, the least-significant byte of a data item shall be stored in the first byte of a given field.

Within tuples, all character data shall be stored in the natural order. That is, the first character of the field shall be stored in the first byte of the field. Fixed-length character fields shall be padded with null characters, if necessary.

### 2.3.3 Byte Order on Wide Cards

If a card has a data path wider than 8-bits, one must assign a byte order to the data path. This applies to all CardBus PC Card CIS's and to those fields within a 16-bit PC Card CIS that are recorded in Common Memory space. At present, Attribute Memory Space is byte-wide only; only the even bytes are present. This standard requires that the low-order byte of word 0 be used to record byte 0 of the CIS. Ascending bytes of each word shall be used to sequentially record bytes from the CIS. When the first word is filled, the same process shall be repeated on subsequent words

until the entire CIS is recorded. On x86 architecture machines, this byte order is equivalent to the native order; other machines may need to reorder the bytes when reading or writing the CIS.

The basic compatibility layer does not impose any particular byte order on non-header portions of the card. However, some data-format layers impose further requirements.

### 2.3.4 16-bit PC Card Metaformat in Attribute Memory Space

16-bit PC Cards have two address spaces: Attribute Memory space and Common Memory space. The electrical specification for 16-bit PC Cards requires that information be placed only in even-byte addresses of Attribute Memory space. The contents of odd-byte addresses of Attribute Memory space are not defined.

For simplicity, this specification describes the tuples of the Metaformat as if the bytes of each tuple were recorded consecutively. When a tuple is recorded in Common Memory space of a 16-bit PC Card, the bytes will indeed be recorded consecutively. However, when a tuple is recorded in Attribute Memory space, the data will be recorded in even bytes only.

Link fields of tuples stored in Attribute Memory space are handled as follows. If only the even bytes are read, and the tuples are packed into consecutive bytes in system memory, the link fields shall be set appropriately for byte addressing. This means that the link-field values are conceptually the same whether a tuple resides in Common or in Attribute memory. However, this does mean that if Attribute Memory is directly addressed, the offset to the next tuple in Attribute Memory is two times the link field.

### 2.3.5 16-bit PC Card Metaformat in Common Memory Space

For cost reasons, many 16-bit PC Cards, such as ROM cards, will not implement a separate Attribute Memory space. On these cards, regardless of the state of the REG# line, memory cycles always access Common Memory. These cards provide an Attribute Memory-style CIS starting at byte zero of the card, and recorded in even bytes only. If, for space reasons, the manufacturer wants to switch to a Common Memory-style CIS (packed into ascending bytes), a long link to Common Memory shall be embedded in the CIS. The Common Memory CIS may be stored immediately following the Attribute Memory CIS.

It is important to distinguish between Attribute Memory *space* and Attribute Memory. All 16-bit PC Cards will have Attribute Memory space, accessed by asserting the REG# pin. In addition, some 16-bit PC Cards will have a distinct physical Attribute Memory. In this case, the contents of location 0 in Attribute Memory space will be different and distinct from the contents of location 0 in Common Memory space. However, some 16-bit PC Cards will not have Attribute Memory distinct from Common Memory. Here, memory-read operations from a given location in Attribute Memory space will return the same data as read operations from the same location in Common Memory space. Data accessed from Attribute Memory space must be stored in the even bytes only, even if Attribute Memory is not distinct from Common Memory. Regardless of the presence or absence of Attribute Memory, the CIS for 16-bit PC Cards always begins at location 0 of Attribute Memory space.

This standard allows attribute information to be stored both in Attribute Memory and Common-Memory space. Tuples stored in Common Memory space are recorded in sequential bytes. Both the card's even and the odd bytes are used to record data.

Note: The use of odd bytes to represent tuple data is controlled by the logical-address space in which the tuple resides, not by the type of memory actually used to record the tuple. If the tuple is intended to be accessed in Attribute Memory space, it must be stored only in the even bytes. If it is intended to be accessed in Common Memory space, it must be stored in both even and odd bytes following a longlink target.

### 2.3.6 16-bit PC Card Metaformat for Multiple Function Cards

Multiple function 16-bit PC Cards shall contain multiple Card Information Structures (CIS). The first or global CIS on a 16-bit PC Card shall identify the card as containing multiple functions by the presence of a *CISTPL\_LONGLINK\_MFC* tuple. The 16-bit PC Card shall also contain a separate function-specific CIS for each set of Configuration Registers on the card.

The starting location of each function-specific CIS is given in a single *CISTPL\_LONGLINK\_MFC* tuple in the global CIS. Each function-specific CIS begins with a *CISTPL\_LINKTARGET* tuple.

The global CIS on a multiple function 16-bit PC Card shall contain the following tuples.

Note: A *CISTPL\_FUNCID* with a *TPLFID\_FUNCTION* field reset to zero (0) shall not be placed in the CIS of a Multiple Function PC Card. This tuple is reserved for vendor-specific multiple function PC Cards that do not follow the multiple function PC Card definitions in this specification.

Table 2-2 Global CIS for Multiple Function PC Cards

Tuple	Code	Presence
<i>CISTPL_DEVICE</i>	01H	Mandatory (if PC Card has 5 volt key)
<i>CISTPL_EXTDEVICE</i>	09H	Mandatory (if PC Card has > 64 MBytes of common memory)
<i>CISTPL_DEVICE_OC</i>	1CH	Recommended
<i>CISTPL_LONGLINK_MFC</i>	06H	Mandatory
<i>CISTPL_VERS_1</i>	15H	Mandatory
<i>CISTPL_MANFID</i>	20H	Mandatory
<i>CISTPL_END</i>	FFH	Mandatory

There is a function-specific Card Information Structure for each function on a Multiple Function PC Card. The following tuples are contained in each function-specific CIS.

Table 2-3 Function-specific CIS for Multiple Function PC Cards

Tuple	Code	Presence
<i>CISTPL_LINKTARGET</i>	13H	Mandatory
<i>CISTPL_CONFIG</i>	1AH	Mandatory
<i>CISTPL_CFTABLE_ENTRY</i>	1BH	Mandatory
<i>CISTPL_FUNCID</i>	21H	Recommended
<i>CISTPL_FUNCE</i>	22H	Recommended
<i>CISTPL_END</i>	FFH	Mandatory

### 2.3.7 CardBus PC Card Metaformat

There is one CIS per card function, which may consist of multiple tuple chains in different spaces. Tuple chains may be located in any of the card space with the exception of I/O space. All tuple chains must start with a CISTPL\_LINKTARGET tuple aligned on a four word boundary, but subsequent tuples within a chain need not be so aligned. The beginning of the function's CIS is pointed to by the CIS Pointer in the function's configuration space header. If the function's CIS does not complete in the current chain, the location of the next tuple chain is indicated by a CISTPL\_LONGLINK\_CB. Tuple chains can be located in the following card spaces:

- **Configuration Space.** Tuple chains in configuration space may only be placed in the device dependent region.
- **Memory Space.** Tuple chains may be located anywhere within memory space.
- **Expansion ROM.** Tuple chains may appear in any of the images in an expansion ROM, but no single chain shall span multiple images. The formats of the CIS Pointer and the *Address-Space-Offset* field of the CISTPL\_LONGLINK\_CB allow specifying a twenty-eight bit offset from the base of any of the first sixteen images in the expansion ROM. The chain need not be located in the image on which the offset is based, allowing chains to be placed in images beyond the first sixteen.

Multi-function CardBus PC Cards have an independent configuration space and CIS for each function. Requiring a CIS for each function allows each function to be generically described and helps in delivering such things as function specific executables in the expansion ROM associated with the function. This also allows functions to be individually manufactured, independent of their eventual placement on a CardBus PC Card.

### 2.3.8 16-bit PC Card Tuple Chain Processing

The information block must be located such that it can be easily found by low-level software. This Standard requires that the primary CIS of a 16-bit PC Card be recorded in Attribute Memory starting at address zero (00H).

The first tuple in the primary CIS chain of a 16-bit PC Card with a 5 volt key must be either a CISTPL\_DEVICE (tuple code 01H), a CISTPL\_NULL (tuple code 00H), or an CISTPL\_END (tuple code FFH, see 3.1.2 *CISTPL\_END: The End Of Chain Tuple* for processing). The CISTPL\_DEVICE (tuple code 01H) must be the first non-control tuple found when traversing the chain(s).

It is recommended that 16-bit PC Cards using a low voltage key begin the primary CIS chain with a CISTPL\_LINKTARGET (tuple code 13H). Low voltage keyed cards shall omit the CISTPL\_DEVICE (tuple code 01H) or shall include a CISTPL\_DEVICE with NULL *Device Info* fields.

Cards with > 64 MBytes of Common Memory must contain the tuple CISTPL\_EXTDEVICE (tuple code 09H).

For flexibility, the CIS of a 16-bit PC Card can be extended into Common Memory. To facilitate automatic identification of "blank" cards, Attribute Memory can be read-only memory.

It is expected that the CIS will be written once when the card is manufactured (or formatted) and then infrequently updated. On any PC Card that expects or requires the CIS to be erased (for example a Flash or EEPROM technology card that erases the CIS area when it is reorganized), it is suggested that problems of process interruption and disaster recovery be addressed. These issues are beyond the scope of the Standard.

Note that most implementations will be limited to reading cards of a specific format, or at most, of a few different formats. Thus, many combinations of values available in the tuples will be non-

portable. It is recommended that implementers restrict themselves to the suggested low-level formats defined in the Media Storage Formats Specification.

### 2.3.9 CardBus PC Card Tuple Chain Processing

The recommended method of traversing a CardBus PC Card CIS is to use the Card Services *GetFirstTuple/GetNextTuple* interface. When a client does this, Card Services will decode any long link tuples for the client. The beginning of each function's CIS is indicated by the CIS Pointer in that function's configuration space header. The *Address Space Indicator* field indicates in which space the CIS begins and the *Address Space Offset* field gives the offset into that space for the memory spaces and the expansion ROM space. For the configuration space, the *Address Space Offset* field gives the absolute address in device-dependent configuration space. (See also the *Electrical Specification* and 3.1.6 *CISTPL\_LONGLINK\_CB: The CardBus PC Card LongLink Tuple*.)

Each tuple chain on a CardBus PC Card must begin with a *CISTPL\_LINKTARGET* (tuple code 13H). It may contain one *CISTPL\_LONGLINK\_CB* (tuple code 05H) to another tuple chain in the current space or another space. The *CISTPL\_LONGLINK\_CB* tuple allows the placement of tuple chains in configuration space, memory space, or the expansion ROM.

Traversing a CardBus PC Card's CIS(s) is the same as traversing that of a 16-bit PC Card, with the following exceptions:

1. There is a separate CIS for each card function.
2. For a given function, the beginning of the CIS is pointed to by the CIS Pointer.
3. All tuple chains, including the first one, begin with a *CISTPL\_LINKTARGET* tuple.
4. The encoding of the *CISTPL\_LONGLINK\_CB* address matches that of the CIS Pointer.

The client requests tuples from each function's CIS by specifying the logical function number (0 through 7).

### 2.3.10 Tuple Processing Recommendations

This standard requires that system software be carefully coded in order to prevent incompatibilities from one system to another. The following are some specific recommendations.

The routine that reads a given tuple should be coded to start by examining the tuple code. If the tuple code is not recognized by the routine (e.g. if the code is vendor specific or represents an extension under a future standard), then the tuple should be ignored. If the code is not recognized, it is safe to read the code byte and the link byte. However, other bytes within the tuple may represent active registers.

#### 2.3.10.1 Tuple Code Known

If the tuple code is known, and if the tuple does not contain active registers (which is the case for all standard tuples), then the routine should copy bytes into a buffer in main storage. Bytes should be copied from the code byte up to the last byte before the next tuple. If the link field is FFH (which also means end-of-chain and is only allowed for a 16-bit PC Card) then a maximum of 257 bytes — the code byte, the link byte and as many as 255 bytes of tuple data — should be copied from the card to the main store.



### 2.3.10.2 Processing Longlink Tuple

When processing a longlink tuple (CISTPL\_LONGLINK\_A, \_C, \_CB), software should merely record the target address and address space. The software should not validate the target address, nor should it immediately begin processing of tuples from the target address. Similarly, when a no-link tuple (CISTPL\_NO\_LINK) is found, that fact should be recorded for later use.

Longlink and no-link tuples should be processed *after* reaching the end of the tuple chain. At that time, if a longlink is to be processed, software should validate the target address (by checking for a CISTPL\_LINKTARGET tuple) and begin processing the target chain if it appears to be valid.

### 2.3.10.3 Longlink Pointing to Invalid Tuple Chain

A longlink that points to an invalid tuple chain should not usually cause any diagnostic messages to be displayed to the user. This situation may result from an uninitialized card, from a card which was initialized for some unanticipated use, or from corrupted data. Since only the corrupted data case merits a diagnostic message, it is better to assume either that the card is uninitialized, or that it is initialized in some non-conforming way.

### 2.3.11 16-bit PC Card CIS with Indirect Access PC Card Memory

16-bit PC Cards with very limited Attribute and Common Memory spaces can utilize an optional, Common Memory register-based indirect access mechanism. (See the *Electrical Specification*.) Using this indirect access method these cards provide full disclosure of their capabilities and attributes by extending their CIS into the indirect space(s). In all cases, a minimal legal CIS must be placed in the standard Attribute and Common Memory spaces.

The CISTPL\_INDIRECT tuple is used to indicate the presence of indirect access registers. After processing tuple chains in the Attribute and Common memory spaces, tuple processing follows the implied link indicated by the CISTPL\_INDIRECT tuple to the indirect Attribute and indirect Common spaces. Once tuple processing begins in the indirect spaces there is no return link to the direct access spaces. This means that all longlink tuples placed in the indirect spaces refer to the indirect spaces, i.e. a CISTPL\_LONGLINK\_A refers to indirect Attribute memory.

Tuple processing in the indirect spaces follows the previously stated rules for processing tuples on all 16-bit PC Cards. For example, the implied link goes first to the indirect Attribute space where a CISTPL\_LINKTARGET should be located at indirect Attribute address zero. There is an implied link to indirect Common address zero if there is no tuple chain at indirect Attribute address zero.

An example of a minimal, but legally sufficient, Card Information Structure is shown in the table below.

Table 2-4: Minimal CIS for Indirect Memory Access PC Cards

Address Space / Offset	Tuple	Values
Attribute / 00H	CISTPL_DEVICE	01H 02H 00H FFH
Attribute / 08H	CISTPL_INDIRECT	03H 00H
Attribute / 0CH	CISTPL_END	FFH
Common / 00H	CISTPL_END	FFH

## 3. FUNCTIONAL DESCRIPTION

### 3.1 Architecture

Safely using PC Cards and sockets in a non-conflicting manner involves the interaction of several hardware and software architectural layers.

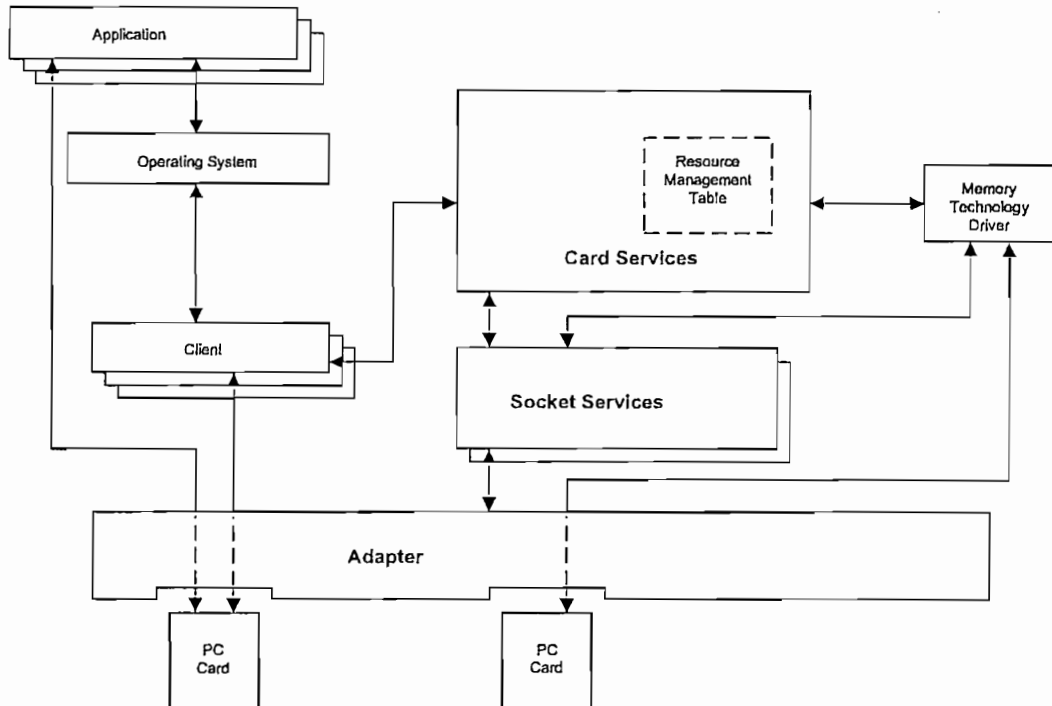


Figure 3-1: Software Architecture Diagram

#### 3.1.1 Hardware Layer (PC Cards, Sockets and Adapters)

Cards compliant with the PC Card Standard are referred to as PC Cards. Originally the standard specified data storage or memory cards. Later releases of the standard expanded the definition of PC Cards to include peripheral expansion or I/O cards, provided for additional tuples, and further refined the software interface. The latest release specifies 3.3 volt operation, third party DMA operation, standard multiple function PC Cards and a 32-bit interface for all types of PC Cards.

All PC Cards have the same physical characteristics and compatible electrical characteristics.

PC Cards are plugged into sockets on a host system. Host systems may have one or more sockets and these sockets may be grouped together on one or more adapters. An example of a host system with more than one adapter would be one where an adapter was built into the motherboard and another plugged into the system's expansion bus.

Adapters usually generate maskable hardware interrupts when status changes occur in sockets or on PC Cards. Status changes include:

- card inserted or removed,
- battery low or dead,
- ejection or insertion request,
- card locked or unlocked in socket, and
- a busy to ready transition.

### 3.1.2 Socket Services

Immediately above the hardware layer the Socket Services software provides a standardized interface to manipulate PC Cards, sockets, and adapters. (See the *Socket Services Specification*.)

As noted above, host systems may have more than one PC Card adapter present. Each adapter may have its own Socket Services handler. All instances of Socket Services are intended to support a single instance of Card Services. Card Services registers to receive notification of status changes on PC Cards or in sockets.

By making all accesses to adapters, sockets, and PC Cards through the Socket Services interface, higher-level software (including Card Services) is unaffected by different implementations of the hardware. Only a hardware-specific Socket Services implementation must be modified to accommodate any different hardware implementations.

### 3.1.3 Card Services

Above the Socket Services software layer is the Card Services layer. Card Services coordinates access to PC Cards, sockets and system resources among multiple clients. These clients may be resident or transient device drivers, system utilities, or application programs. There is only one Card Services implementation in a host system. (Unlike Socket Services where there may be multiple implementations to accommodate multiple adapters).

Card Services makes all access to the hardware layer through the Socket Services software interface. The single Card Services implementation is intended to be the sole client of all Socket Services implementations present. All Socket Services status change reporting is routed to this single Card Services implementation. Card Services then notifies interested clients when status changes occur.

To prevent conflicts with clients who are unaware of Card Services, direct access to the Socket Services interface is blocked by Card Services. A method of bypassing the Card Services blockage is provided for software developers of specialized applications which must access Socket Services. Programs which bypass Card Services and make direct access to Socket Services must ensure such access is benign and does not interfere with Card Services usage of Socket Services, PC Cards, sockets, or adapters.

Card Services preserves for its clients an abstract, socket-hardware-implementation independent view of a card and its resources. Card Services presents the same tuple organizational and resource allocation view to all of its clients whether the card is a 16-bit PC Card or a CardBus PC Card.

### 3.1.4 Memory Technology Drivers

The PC Card Standard supports a wide range of memory devices on PC Cards. While all PC Cards containing any such memory device may be read as if they contained static-RAM devices, special programming algorithms may be required to write or erase the memory devices. Card Services

hides the details of what is required to write or erase memory devices from client device drivers through byte-oriented write and copy services and a block-oriented erase service.

Within Card Services, Memory Technology Drivers (MTD) implement the specific programming algorithms required to access memory devices. These drivers may be embedded within Card Services or may register with a Card Services implementation at run-time. When PC Cards are installed, MTDs monitoring insertion events register with Card Services to support access to a memory device region through the Card Services read, write, copy, and erase services.

Card Services provides default MTDs for recognized regions. If Card Services recognizes a region as being composed of Static RAM devices, it installs a default MTD that supports read and write requests. Reads and writes are performed as simple memory accesses without any algorithmic operation. If Card Services recognizes a memory region but not the type of devices in the region, it installs a default MTD that supports read and write requests and fails erase requests. The reads and writes are performed as simple memory accesses without any algorithmic operation. Card Services may include MTD support for other device types that require specific programming algorithms. (See 3.6 *Memory Technology Drivers* and see also *Appendix-E, 10. MTD Helper Service Reference*.)

### 3.1.5 Client Device Drivers

Client device drivers refers to all users of Card Services. These may be device drivers, utility programs, or application programs.

## 3.2 Programming Interface

### 3.2.1 Calling Conventions

The Card Services interface uses a common set of conventions for all services.

#### 3.2.1.1 Basic Operation

Card Services is invoked in a processor and Operating System dependent manner called a binding. (See *Appendix-D, 9. Bindings*.)

All arguments for Card Services requests are passed in binding specific fashions. Card Services defines six generic arguments:

Service	Status
Handle	Argument Length
Pointer	Argument Pointer

Many Card Services requests pass all data in the *Service*, *Handle* and *Pointer* arguments. For such services, no argument packet (as referenced by *ArgPointer* below) is required. If a request requires more than these generic arguments, an argument packet must be used. Status of the Card Services request is returned in the Status argument. Using functional notation, a generic Card Services call is as follows:

```
status = CardServices(Service, Handle, Pointer, ArgLength, ArgPointer)
```

All requests pass the service code of the request in the *Service* argument. Individual services and their service field values are described in later sections. Many requests require a Card Services handle to identify some resource. These requests pass the handle in the *Handle* argument. Some requests require an additional pointer value which is passed in the *Pointer* argument.

Many Card Services requests have an additional argument packet which is pointed to by *ArgPointer*. The length of the argument packet is passed in the *ArgLength* argument. If the *ArgLength* argument is zero, there is no argument packet and the value of the *ArgPointer* argument is undefined.

The *ArgLength* argument may be used by a Card Services implementation to validate that the argument packet is appropriate for the indicated service. In different releases of this specification, the appropriate length of the argument packet may vary. Card Services uses this field to determine which packet length, and by extension, which version of the packet is being used by the client requesting the service.

See also *Appendix-D, 9. Bindings* for specific processor bindings for the generic Card Services arguments.

### 3.2.1.2 Argument Packet

Most argument packets are a fixed size determined by the particular service as implemented for a particular publication of this specification. Some argument packets can be variable in length. The size of these variable packets is determined by the caller. Variable length packets are used to contain data set by Card Services, for example, the Vendor Name ASCII string used to identify a particular version of Card Services.

The *ArgLength* argument indicates the length of the total packet. For variable length argument packets, there are additional fields in the packet that indicate the maximum length of the variable portion (set by the caller) and the actual length of the returned data (set by Card Services). Also some requests have more than one variable length argument. In this case, there is also an offset field that indicates where each additional variable length field begins.

The specific content of an argument packet is defined for each request that requires an argument packet.

### 3.2.1.3 Logical Sockets

The Card Services interface, except for *MapPhySocket*, uses logical sockets in identifying the socket a service is intended to address. The first physical socket on the first physical adapter is logical socket zero (0). The maximum logical socket is the total number of sockets present minus one.

### 3.2.1.4 Reserved Fields

Reserved fields and undefined bits shall be reset to zero before invoking a service because future releases of Card Services may define them. Future releases will use the reset value for behavior compliant with this release of Card Services.

Any reserved fields or undefined bits in fields returned by Card Services are reset to zero by Card Services so future releases of Card Services will be able to notify clients in a manner compliant with this release.

### 3.2.1.5 Multi-Byte Fields

All multi-byte fields are stored in binding specific format. Multi-byte data returned in bulk from a PC Card is kept in little-endian format with the least significant byte appearing first in memory. For example, the *GetTupleData* and *Read/Write/CopyMemory* requests transfer the data without any byte swapping processing.

(See also *Appendix-D, 9. Bindings*.)

### 3.2.1.6 Multiple Function PC Cards

Some PC Cards may contain multiple functions. To address a particular function on a PC Card, the client passes a card function number in the logical socket field of the appropriate request. Card Functions are numbered from zero to one less than the number of functions on the PC Card.

### 3.2.2 Presence Detection

The Card Services GetCardServicesInfo service is used to determine the presence of Card Services. If this request fails, Card Services is not present. If it succeeds, Card Services is present.

### 3.2.3 Initialization of Card Services

Card Services is designed to be implemented as an Operating System Dependent Device Driver or OS extension. If a processor supports different modes of operation, Card Services can assume that it is used in only a single mode. For example, processors in x86 architecture systems can run in Real Mode or Protect Mode. Card Services can assume that it is only used in one of these modes at any time.

During initialization, Card Services determines the state of the host environment. This includes determining available system memory, available I/O ports, DMA Channels, IRQ assignments, installed PC Cards, and socket states.

Initialization is implementation specific.

After Card Services initializes, all Socket Services requests (080H through 0AEH) are blocked. Card Services returns an UNSUPPORTED\_SERVICE error if any attempt is made to use these services. This prevents Socket Services clients who are unaware of the Card Services interface from crashing the system by making direct access to hardware through Socket Services. Such crashes could be caused by changing hardware state without Card Services being aware of the change. Should a Card Services aware client still require access to Socket Services, it may do so by using the entry point returned by the ReturnSSEntry service.

During initialization, Card Services determines all Socket Services implementations present so that it can manage the status change interrupt handling required for adapters. Socket Services status events are enabled based on client event masks. If no clients request an event, Card Services does not need to enable the event. Card Services records the event when it occurs and notifies any clients who have registered for its status change event and who have unmasked the event specified.

Card Services notifies registered clients and Memory Technology Drivers when events requiring callback notification have occurred. Notification is delayed until Card Services is in an enterable state which allows callback handlers registered with Card Services to make requests during event notification so they may reconfigure immediately to react to the event.

### 3.2.4 Return Codes

Card Services indicates success or failure of a request with the generic Status argument. If the Status argument is set to a non-zero value on return from a Card Services request, the request failed and the value in the Status argument describes why the request failed. If the Status argument is reset to zero on return from a Card Services request, the request succeeded. (See *Appendix-C, 8. Return Codes* and see also *Appendix-D, 9. Bindings*.)

## 3. FUNCTIONAL DESCRIPTION

### 3.1 System Architecture

Socket Services is a software interface to the hardware used to manage PC Card sockets in a host system. Above Socket Services, an operating system-specific layer known as Card Services virtualizes Socket Services to allow it to be shared by multiple processes. These processes may include such things as eXecute-In-Place (XIP), Flash File System (FFS), and other types of device drivers.

Socket Services provides only the lowest level access to PC Cards. For example, Socket Services allows the 16-bit PC Card attribute memory space to be read, but it does not interpret the Card Information Structure (CIS).

Socket Services is invoked in a platform dependent manner. All service arguments are passed to Socket Services in a binding specific fashion. Status of a Socket Services request is returned in the status argument. (See *Appendix-C, 9. Socket Services Bindings*.) Using functional notation, a Socket Services request generically can be considered as:

```
status = service(arg1, arg2 ...)
```

While this notation resembles a C language function call, Socket Services is implemented in an appropriate manner for its environment. For example, on an x86 architecture platform a ROM BIOS Socket Services interface is handled through Interrupt 1AH with services based at 80H. A client simply sets the host processor's registers for the service desired and executes the Socket Services software interrupt. Status is returned using the Carry flag ([CF]) and registers specific to the service invoked.

Special handling is required to be able to write many types of memory cards. It is not feasible to attempt to include all the necessary handlers within Socket Services for all the possible types of write/erase routines. Handling of technology-specific write requirements is intended to be performed by a software layer above Socket Services. Socket Services provides access to the hardware for these card technology routines.

### 3.2 Initialization

Socket Services is internally initialized during installation and no specific installation is required by the client before making service requests. It is expected the client of Socket Services will check the Socket Services *Compliance* to determine the level of service available. (See 5.3.12 *GetSSInfo [BOTH]*.)

### 3.3 Configuration

The next step is to enumerate the capabilities of the implementation. This entails determining the number of adapters installed, how many sockets, bridge and 16-bit PC Card windows are supported by each adapter, and exploring the power management and indicators available for each adapter.

As noted above, it is expected that Socket Services is virtualized by Card Services. Above Card Services are device drivers for different types of PC Cards. These drivers map PC Cards into system I/O and/or memory space to implement their functions. Multiple drivers may share PC Cards and

sockets and may even share windows. Card Services arbitrates requests for Socket Services resources and is responsible for preserving any state information required to share these resources.

### 3.4 Status Change Notification

A Socket Services client may desire notification when a status change occurs. Status changes include, but are not limited to, the following: card removal or insertion, battery low or dead, and READY changes. Socket Services supports steering and enabling status change interrupts from an adapter. A client installs a status change interrupt handler on the host interrupt level selected to receive such interrupts. A client may choose to poll for changes in socket and card status.

When an adapter configured for status change interrupts detects a status change, it generates an interrupt which invokes the client's status callback handler. This handler uses the Socket Services `AcknowledgeInterrupt` service to determine which socket or sockets experienced the status change. It records this information and completes the hardware interrupt processing. Later, during background processing, the client notes which sockets require attention and uses the `GetStatus` service to determine current PC Card and socket state. This state is used to determine what action should be taken by the client. Status change interrupt handling is provided by Card Services. (See the *Card Services Specification*.)

### 3.5 Power Management

The Socket Services interface provides controls for conserving adapter power. Two power conservation modes are provided: reduced with all state information maintained and reduced without state information being maintained. These levels are established with the `SetAdapter` service.

Socket Services may also be used to manage power to PC Card sockets. Independent controls and levels are provided for VCC, VPP1 and VPP2. Since available power levels are generally limited, Socket Services provides a list of supported levels and then allows power adjustment based on an index into that list. Power management is performed at the socket level. How Socket Services resolves power management requests in hardware implementations that only allow control of power at the adapter level is vendor specific. Socket Services reports the level of power management control available through the `InquireAdapter` service.

### 3.6 Docking

Whether or not Socket Services is dynamically loaded (or unloaded) there is a general sequence of things that Socket Services needs to perform in order to handle dock events. Considering all possible dock scenarios Socket Services really is performing one of three actions: add support (dock where new controllers are present requiring new Socket Services handlers), remove support (undock where controllers are gone requiring removal of Socket Services handlers) or change/replace support (either dock or undock using same socket services instance). This leads to the following sequences for communications between Socket Services and Card Services:

- I. Replace Support
  - A. Socket Services issues `ReplaceSocketServices` to Card Services w/ Base log, Socket # (obtained via `MapPhyLogSocket`) and number of sockets to replace. Until Socket Services receives `GetSetPriorHandler` or Card Services returns from `ReplaceSocketServices` this Socket Services rejects any request (except `GetSSInfo`, see below for more info) w/ BUSY return code.
  - B. Upon receipt of `GetSetPriorHandler`



1. If previous is NULL then return w/ adapter 0; else,
  2. Add itself as supporting next adapter (if any such adapter exists that needs support else may take steps to remove itself from memory if environment supports this).
- C. Receives return from ReplaceSocketServices request.
- D. Receives and processes normal "initialization" requests from Card Services.
- II. Add Support
- A. Socket Services issues AddSocketServices to Card Services
  - B. Socket Services receives GetSetPriorHandler and before returning numbers his adapter (via GetSSInfo) and return
  - C. Returns proper data to GetSSInfo
  - D. Receives return from AddSocketServices
  - E. Receives and processes normal "initialization" requests from Card Services.
- III. Remove Support
- A. Use same logic flow as Replace Support except return zero (0) supported adapters for GetSSInfo request.

NOTE: The GetSetPriorHandler request is used by Card Services implementations that expect Socket Services handlers to track the chain of handlers. Some Card Services implementations will track the handlers themselves and in this situation Socket Services may not receive any GetSetPriorHandler requests during processing of dock events.

## 3.7 Overview of Services

### 3.7.1 Non-specific Service

There is one Socket Services service which applies to the interface in general and not to any objects manipulated by the interface. It is:

GetAdapterCount

### 3.7.2 Adapter Services

Socket Services addresses adapters with the following services:

AcknowledgeInterrupt	GetSSInfo
GetSetPriorHandler	GetVendorInfo
GetSetSSAddr	InquireAdapter
GetAccessOffsets	SetAdapter
GetAdapter	VendorSpecific

### 3.7.3 Socket Services

Socket Services addresses sockets with the following services:

GetSocket	ReSetSocket
GetStatus	SetSocket
InquireSocket	AccessConfigurationSpace

### 3.7.4 Window Services

Socket Services addresses windows with the following services:

GetBridgeWindow	SetBridgeWindow
GetPage	SetPage
GetWindow	SetWindow
InquireWindow	InquireBridgeWindow

**WARNING:**

*Windows which map 16-bit PC Cards into host system memory address space may have one or more pages. If a Window contains multiple pages, each page must be 16 KBytes and windows must be sized as a multiple of the 16 KByte page size.*

### 3.7.5 Error Detection and Correction Services

Adapters and/or Sockets may optionally provide error detection and correction support. The following services handle EDC capabilities:

GetEDC	ResumeEDC
InquireEDC	SetEDC
PauseEDC	StartEDC
ReadEDC	StopEDC

### 3.7.6 Status Change Handling

Socket Services provides for asynchronous notification when a socket's status changes. Each adapter may provide a hardware interrupt when there is a status change. This interrupt is processed by a handler installed by the Socket Services client.

While only one interrupt per adapter is anticipated, the Socket Services interface allows status changes to be masked on a per socket basis. Masking must be performed in hardware since the hardware interrupt is handled directly by the Socket Services client.

If status change interrupts are supported, each Socket Services client determines which interrupt it uses for status changes based on the set of supported interrupts reported by **InquireAdapter**. A Socket Services client may enable or disable this capability and may steer the interrupt to a supported host interrupt level.

## 3. PARTITIONS

PCMCIA/JEIDA recognize two methods for partitioning PC Cards. First, linear memory PC Cards such as flash and S-RAM cards use tuples in the Card Information Structure (CIS) to describe how a PC Card is partitioned. Second, PC Card ATA drives are partitioned using a Master Boot Record with a partition table.

Each recognized partitioning method is described separately in the following sections. The following information is provided about each recognized method:

Overview	An overview of the partitioning method and where it is used.
Partition Operations	How partition operations are performed.
Initial Program Load	How a host system boots using the partitioning method.
Data Structures	The data structures used by the partitioning method.

All PC Cards used for data storage must provide partition information as described in this section. PC Cards used for data storage that do not contain partition information described in this section may be assumed to be unformatted.

## 3.1 Card Information Structure (CIS) Partitioning

### 3.1.1 Overview

The PC Card's CIS describes partitions using the following tuples:

Tuple Name	Tuple Constant	Tuple Value
Format	CISTPL_FORMAT	41H
Organization	CISTPL_ORG	46H

Both tuples must be present. The Format Tuple describes where that partition is located on the media and the Organization Tuple identifies the data storage format used within the partition. Data storage within a partition is also affected by the following tuples, if they are present

Tuple Name	Tuple Constant	Tuple Value
Geometry	CISTPL_GEOMETRY	42H
Byte-Order	CISTPL_BYTEORDER	43H
Software Interleave	CISTPL_SWIL	23H

### 3.1.2 Partition Operations

#### 3.1.2.1 Creation

A partition is created by adding the tuples described above to a PC Card's Card Information Structure (CIS). The ability to write to a PC Card's CIS is dependent on the card and potentially installed device drivers. A PC Card may require that the entire CIS be erased and then re-written to modify the CIS.

Some PC Cards use multiple tuple chains to describe physical characteristics of the card separately from how the card is used. For example, a primary tuple chain in a PC Card's attribute memory space might describe the physical characteristics of the card, such as the type and size of the memory device used on the card. A secondary tuple chain, in the PC Card's common memory space, might be used to describe the logical characteristics of the card, such as the partitioning. In this manner, the PC Card might be manufactured with the physical information hard-coded into read-only memory in attribute memory space and logical partitioning information would be added by using writable memory in common memory space. (See the *Metaformat Specification* for more information about how tuple chains are linked together within the CIS.)

#### 3.1.2.2 Deletion

To delete a partition, all of the tuples describing the partition must be deleted from the Card Information Structure (CIS). Depending on the PC Card, the CIS may have to be erased and then re-written without the tuples that describe the partition.

### 3.1.2.3 Extension

Some PC Cards allow the Card Information Structure to be extended without erasing existing tuples. These cards permit additional partitions to be defined by adding tuples to the end of the last tuple chain on the PC Card.

### 3.1.2.4 Evaluation Order

Host software evaluates partition information as it is encountered in the Card Information Structure (CIS). If host software recognizes a partition type, the next available drive specifier is assigned to the partition. If there are multiple partitions of different types on a PC Card, each partition type may be recognized by a separate host device driver. For this reason, the order host drive specifiers are assigned to partitions is host system specific.

## 3.1.3 Initial Program Load

The PC Card Standard does not currently define a method for booting from a PC Card using partition definitions in the CIS.

## 3.1.4 Data Structures

See the *Metaformat Specification* for a complete definition of the tuples used to describe partitions.

## 3.2 Master Boot Record (MBR)

### 3.2.1 Overview

PC Card ATA drives are partitioned using a Master Boot Record (MBR) with a Partition Table in the first physical sector of the media. Partition Table Entries describe the size, location and type of data within a partition. A Partition Table Entry may also describe an Extended Partition which is further divided into one or more partitions.

The MBR contains a word of 55AAH at offset 1FEH. The sector contains operating system independent code to perform Initial Program Load on x86 architecture host systems. For system and PC Card interoperability, all systems, including those that do not use the IPL code for booting, must include such information when formatting the MBR. The MBR also contains a Partition Table with four (4) Partition Table Entries at offset 1BEH. When booting from a device with an MBR on an x86 architecture system, code within the MBR evaluates the Partition Table for a partition marked as the default boot partition. Only one partition may be marked as the default boot partition.

If the x86 bootstrap code locates a default boot partition in the MBR's Partition Table, the Partition Boot Record (PBR), the first sector of the partition, is loaded into memory. If the word at offset 1FEH of the PBR is 55AAH, control is passed to the next stage bootstrap loader in the PBR image in memory.

During operating system initialization, MBRs on all fixed disk devices are evaluated by the file system for partition definitions in reverse order starting with the entry at offset 1EEH. The host system assigns unit designations (drive letters under MS-DOS), to each partition matching a type supported by the file system.

### 3.2.2 Partition Operations

#### 3.2.2.1 Creation

A partition is created by setting the fields of a partition entry in the partition table of the Master Boot Record (MBR) to describe the desired partition. A partitioning utility first reads the MBR into host system memory. If the word at offset 1FEH of the MBR is not 55AAH, the device is not formatted and the utility must create an initial MBR with an empty partition table before proceeding.

If the word at offset 1FEH of the MBR is 55AAH, the partitioning utility searches the partition table for an empty entry. An entry is considered empty if the NumSectors field is zero (0). If there are no empty entries in the partition table, a partition cannot be created.

If there is an empty partition entry, the partitioning utility creates a new partition using contiguous unallocated space on the media. The utility determines if there is any available space on the media by subtracting the space allocated to other partitions from the total size of the media. The total size of the media is determined in a host system dependent manner. For example, on x86 systems with a PC-compatible ROM BIOS the Get Drive Parameters function (Int 13H Function 8) is typically used.

If there is unallocated space on the media, the partitioning utility must also determine where the space is located by comparing the Start and End of each allocated partition. How a partitioning utility decides which space to use when multiple unallocated spaces are available is implementation specific.

After a partition entry is created the partitioning utility needs to notify the host system of the change to the MBR.

A partition table entry uses Cylinder, Head and Sector (CHS) addressing to describe the starting and ending boundaries of a partition. Some PC Card ATA drives translate their physical CHS information to logical values that are compliant with limits imposed by some host systems that are unable to address cylinder, head or sector values that exceed system-specific limits. Once a partition table entry has been created, all subsequent accesses to the media must use the same logical CHS addressing.

The PC Card Standard requires that all partitions described in the partition table within the MBR end on a logical cylinder boundary based on the logical CHS addressing in use when the first partition was created. This allows a host system to validate the logical CHS addressing in use is correct by confirming the maximum head and sector values used for media access are the same as those used to indicate the ending head or sector of all partitions on the media.

### 3.2.2.2 Deletion

A partition is deleted by resetting all of the fields of a partition entry in the partition table of the Master Boot Record to zero (0). After a partition entry is deleted the partitioning utility needs to notify the host system of the change to the MBR.

### 3.2.2.3 Extension

Some partition types extend the partition table in a system-specific manner. For example, MS-DOS defines a special partition type called the Extended MS-DOS partition. The space allocated to an Extended MS-DOS partition is sub-divided into logical drives. The first sector of an Extended MS-DOS Partition contains a partition table formatted in the same manner as the partition table in a Master Boot Record (MBR). The extended partition table typically contains two entries, an MS-DOS partition and another Extended MS-DOS partition entry.

The MS-DOS partition entry in an extended MS-DOS partition describes a logical drive. If an Extended MS-DOS partition entry is also present in the partition table, another potential logical drive may exist within the area described by the Extended MS-DOS partition entry. Extended MS-DOS partition entries create a forward-linked list of logical drives within the Extended MS-DOS partition in the MBR.

The one difference between partition entries in an MBR and partition entries in the partition table in an Extended MS-DOS partition is the StartSector field of the partition entries. In the MBR this field is relative to the beginning of the media. In an Extended MS-DOS partition this field is relative to the beginning of the Extended MS-DOS partition described in the MBR.

### 3.2.2.4 Evaluation Order

The order partition entries are evaluated in the partition table of the Master Boot Record (MBR) is dependent on the operating system. For example, MS-DOS evaluates primary partition types on the first two physical fixed drives on x86 systems addressed by the ROM BIOS Int 13H Disk I/O handler as drives 80H and 81H. Primary partition types are 01H, 04H and 06H.

If the first physical fixed drive has a primary partition, MS-DOS assigns the next available logical drive letter to the partition. If there is a second physical fixed drive and it has a primary partition, MS-DOS assign the next available logical drive letter to this partition. After MS-DOS assigns the primary partition types on the first two physical fixed drives as logical drives, Extended MS-DOS partitions are evaluated.

If the first physical fixed drive has an Extended MS-DOS partition, each logical drive described in the chain of Extended MS-DOS partitions is added as a logical drive letter. If there is a second

physical fixed drive and it has an Extended MS-DOS partition, each logical drive described in the chain of Extended MS-DOS partitions is added as a logical drive letter.

### 3.2.3 Initial Program Load

The PC Card Standard does not describe a system independent method for booting from a device with a Master Boot Record (MBR). However, x86 systems use the MBR as the first stage of a multi-stage program loader. The host system reads the MBR of the first physical drive into host system memory at 0000H:7C00H. If the word at offset 1FEH of the MBR is not 55AAH, the media is not bootable and the system continues the boot process with another device or displays an error message.

If the word at offset 1FEH of the MBR is 55AAH, the host system transfers control to the code at 0000H:7C00H. No arguments are provided to the code by the host system. No stack is established and no indication of which device the MBR was read from is provided.

The boot code in the MBR evaluates the partition table from the last entry at offset 1EEH to the first entry at 1BEH. If an entry is found with the default x86 boot partition field set to 80H, the first sector of the partition described by the partition entry, known as the Partition Boot Record (PBR), is loaded into host system memory. If the word at offset 1FEH of the PBR is 55AAH, control is transferred to offset zero of the PBR and the boot process continues. If the word at offset 1FEH of the PBR is not 55AAH, the code in the MBR displays an error message and halts.



### 3.2.4 Data Structures

#### 3.2.4.1 Master Boot Record

The Master Boot Record contains the following fields:

Offset	Size (Bytes)	Description
00H	446	Boot code
1BEH	16	Partition Entry (See below)
1CEH	16	Partition Entry (See below)
1DEH	16	Partition Entry (See below)
1EEH	16	Partition Entry (See below)
1FEH	2	Signature Word (0x55AA)

#### 3.2.4.2 Partition Entry

Each of the four Partition Entries in the Master Boot Record have the following format:

Offset	Size (Bytes)	Description
00H	1	x86 default boot partition 00H = Not default boot partition 80H = Default boot partition
01H	1	StartHead - Zero-based (0) head number where partition starts on media.
02H	1	StartSector - Bits 0 .. 5 are one-based (1) sector number where partition starts on media. Bits 6 and 7 are high bits of zero-based (0) cylinder number where partition starts on media.
03H	1	StartCylinder - Least significant eight bits of zero-based (0) cylinder number where partition starts on media. Upper two bits of starting cylinder number are in StartSector field.
04H	1	Partition Type 00H: Unknown or deleted if NumSectors is zero 01H: MS-DOS 12-bit BPB/FAT < 16 MB 04H: MS-DOS 16-bit BPB/FAT < 32 MB 05H: Extended MS-DOS Partition 06H: MS-DOS 16-bit BPB/FAT >= 32 MB
05H	1	EndHead - Zero-based (0) head number where partition ends on media.
06H	1	EndSector - Bits 0 .. 5 are one-based (1) sector number where partition ends on media. Bits 6 and 7 are high bits of zero-based (0) cylinder number where partition ends on media.
07H	1	EndCylinder - Least significant eight bits of zero-based (0) cylinder number where partition ends on media. Upper two bits of ending cylinder number are in StartSector field.
08H	4	StartSector (relative to beginning of media or Extended MS-DOS Partition)
0CH	4	NumSectors

**ANEXO 4****EL MAX 180**

# MAXIM

## Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

### General Description

The MAX180/MAX181 are complete 12-bit Data Acquisition System (DAS) which combine 8/6-channel input multiplexer, high bandwidth Track-and-Hold (T/H), low-drift zener reference, and flexible microprocessor ( $\mu$ P) interface with high conversion speed and low power consumption. The MAX180/MAX181 can be configured by a  $\mu$ P for unipolar or bipolar conversions and single-ended or differential inputs. Both devices sample and digitize at 100kHz throughput rate and feature a fast 8- or 16-bit  $\mu$ P interface.

The MAX180 has 8 analog input channels, while the MAX181 has 6. The multiplexer output of the MAX180 is fed directly into the Analog-to-Digital Converter (ADC) input. The MAX181 brings out both the multiplexer output and ADC input to separate pins, allowing a programmable gain amplifier to be inserted between the MUX and the ADC.

The systems allow the user to choose between an internal or an external reference. Furthermore, the internal reference value and the offset can be adjusted, allowing the overall system gain and offset errors to be nulled. The multiplexer has high impedance inputs, simplifying analog drive requirements.

### Applications

- High-Speed Servo Loops
- Digital-Signal Processing
- High-Accuracy Process Control
- Automatic Testing Systems

### Features

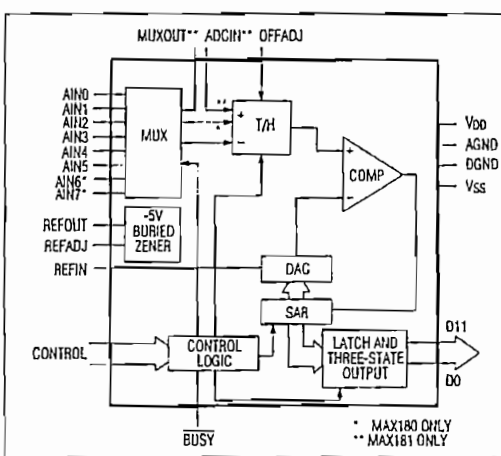
- ◆ 12-Bit Resolution,  $\pm 1/2$ LSB Linearity
- ◆ 8-Channel Multiplexed Inputs (MAX180)
- ◆ Single-Ended 1-of-6 Multiplexer (MAX181)
- ◆ Built-In Track-and-Hold
- ◆ 100kHz Sampling Rate
- ◆ DC and Dynamically Tested
- ◆ Internal 25ppm/ $^{\circ}$ C Voltage Reference
- ◆ Each Channel Configurable for Unipolar (0V to +5V) or Bipolar (-2.5V to +2.5V) Input Range
- ◆ Each Channel Configurable for Single-Ended or Differential Inputs
- ◆ Fast 8-/16-Bit  $\mu$ P Interface
- ◆ +5V and -12V to -15V Supply Operation
- ◆ 110mW Power Consumption

### Ordering Information

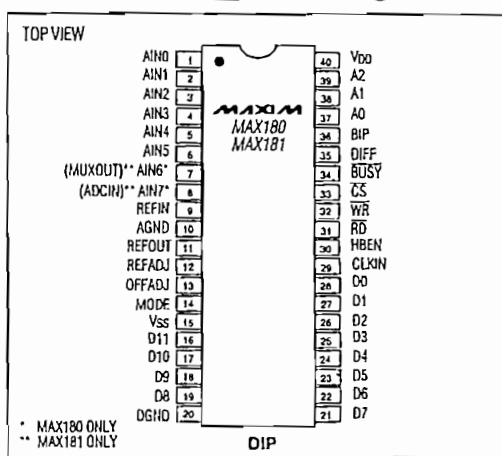
PART	TEMP. RANGE	PIN-PACKAGE	ERROR (LSBs)
MAX180ACPL	0 $^{\circ}$ C to +70 $^{\circ}$ C	40 Plastic DIP	$\pm 1/2$
MAX180BCPL	0 $^{\circ}$ C to +70 $^{\circ}$ C	40 Plastic DIP	$\pm 1$
MAX180CCPL	0 $^{\circ}$ C to +70 $^{\circ}$ C	40 Plastic DIP	$\pm 1$
MAX180ACQH	0 $^{\circ}$ C to +70 $^{\circ}$ C	44 PLCC	$\pm 1/2$
MAX180BCQH	0 $^{\circ}$ C to +70 $^{\circ}$ C	44 PLCC	$\pm 1$

Ordering Information continued on last page.

### Block Diagram



### Pin Configurations



MAXIM

MaxIm Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800

### Electrical Characteristics

(VDD = +5V  $\pm 5\%$ , VSS = -12V  $\pm 5\%$  or -15V  $\pm 5\%$ , REFIN = -5V, Internal Reference Mode, Bipolar Mode, Slow-Memory Mode (see text), CLK = 1.6MHz external, MAX180/MAX181 all grades, TA = TMIN to TMAX, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN TYP MAX UNITS			
			MIN	TYP	MAX	UNITS
Resolution	N		12			Bits
Integral Nonlinearity Error	INL	MAX18_A				Bits
Differential Nonlinearity Error	DNL	MAX18_B/C			$\pm 1/2$	LSB
Unipolar Offset Error (Note 3)		Guaranteed monotonic over temperature			$\pm 1$	LSB
Bipolar Offset Error (Note 3)					$\pm 1$	LSB
Unipolar Gain Error				$\pm 1$	$\pm 4$	LSB
Bipolar Gain Error				$\pm 1$	$\pm 6$	LSB
Gain-Error Tempco (Note 4)				$\pm 2$	$\pm 10$	LSB
Channel-to-Channel Matching				$\pm 2$	$\pm 15$	LSB
				$\pm 5$		ppm/ $^{\circ}$ C
				$\pm 1/4$		LSB
<b>DYNAMIC PERFORMANCE (Note 2)</b>						
Signal-to-Noise + Distortion Ratio	SINAD	10kHz input signal, 100kHz sampling rate, bipolar mode, TA = +25 $^{\circ}$ C	70			dB
Total Harmonic Distortion (up to the 5th harmonic)	THD	10kHz input signal, 100kHz sampling rate, bipolar mode, TA = +25 $^{\circ}$ C				dB
Spurious-Free Dynamic Range	SFDR	10kHz input signal, 100kHz sampling rate			-80	dB

MA

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

## ABSOLUTE MAXIMUM RATINGS

V <sub>DD</sub> to DGND	-0.3V, +7V
V <sub>SS</sub> to DGND	-0.3V, -17V
AGND to DGND	-0.3V, V <sub>DD</sub> + 0.3V
AIN <sub>-</sub> , MUXOUT, ADCIN, REFADJ,	
OFFADJ to REFIN	-0.3V, V <sub>DD</sub> + 0.3V
REFIN to DGND	+0.3V, V <sub>SS</sub> - 0.3V
CS, WR, RD, CLK, A2-A0,	
BIP_DIFF, HBEN to DGND	-0.3V, V <sub>DD</sub> + 0.3V
BUSY, D0-D11 to DGND	-0.3V, V <sub>DD</sub> + 0.3V

Continuous Power Dissipation (any package)	1000mW
to +70°C	
derates above +70°C by	10mW/°C
Operating Temperature Ranges:	
MAX18_C	0°C to +70°C
MAX18_E	-40°C to +85°C
MAX18_MJL	-55°C to +125°C
Storage Temperature Range	-65°C to +160°C
Lead Temperature (soldering, 10 sec)	+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## ELECTRICAL CHARACTERISTICS

(V<sub>DD</sub> = +5V ±5%, V<sub>SS</sub> = -12V ±5% or -15V ±5%, REFIN = -5V, Internal Reference Mode, Bipolar Mode, Slow-Memory Mode (see text), I<sub>CLK</sub> = 1.6MHz external, MAX180/MAX181 all grades, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>ACCURACY (Note 2)</b>						
Resolution	N		12			Bits
Integral Nonlinearity Error	INL	MAX18_A			±1/2	LSB
		MAX18_B/C			±1	
Differential Nonlinearity Error	DNL	Guaranteed monotonic over temperature			±1	LSB
Unipolar Offset Error (Note 3)				±1	±4	LSB
Bipolar Offset Error (Note 3)				±1	±6	LSB
Unipolar Gain Error				±2	±10	LSB
Bipolar Gain Error				±2	±15	LSB
Gain-Error Tempco (Note 4)				±5		ppm/°C
Channel-to-Channel Matching				±1/4		LSB
<b>DYNAMIC PERFORMANCE (Note 2)</b>						
Signal-to-Noise + Distortion Ratio	SINAD	10kHz input signal, 100kHz sampling rate, bipolar mode, T <sub>A</sub> = +25°C	70			dB
Total Harmonic Distortion (up to the 5th harmonic)	THD	10kHz input signal, 100kHz sampling rate, bipolar mode, T <sub>A</sub> = +25°C			-80	dB
Spurious-Free Dynamic Range	SFDR	10kHz input signal, 100kHz sampling rate, bipolar mode, T <sub>A</sub> = +25°C	80			dB
Full-Power Sampling Bandwidth		In track mode, under-sampled waveform		6		MHz
Track-and-Hold Acquisition Time (Note 5)	t <sub>ACO</sub>		1.875			μs
Conversion Time	t <sub>CONV</sub>	Asynchronous hold mode	Note 5	7,500	8,125	μs
		ROM, Slow-Memory, and I/O Port Modes; 15-16 clock cycles		9,375	10,000	
<b>ANALOG INPUT</b>						
Voltage Range		AIN <sub>-</sub> , MUXOUT, and ADCIN	REFIN		V <sub>DD</sub>	V
Unipolar, Single-Ended Range		AIN <sub>-</sub> to AGND	0		5.0	
Unipolar, Differential Range		AIN <sub>+</sub> to AIN <sub>-</sub>	0		5.0	
Bipolar, Single-Ended Range		AIN <sub>-</sub> to AGND	-2.5		2.5	
Bipolar, Differential Range		AIN <sub>+</sub> to AIN <sub>-</sub>	-2.5		2.5	

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

**MAX180/MAX181**

## ELECTRICAL CHARACTERISTICS (continued)

(V<sub>DD</sub> = +5V ±5%, V<sub>SS</sub> = -12V ±5% or -15V ±5%, REF<sub>IN</sub> = -5V, Internal Reference Mode, Bipolar Mode, Slow-Memory Mode (see text), f<sub>CLK</sub> = 1.6MHz external, MAX180/MAX181 all grades, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>ANALOG INPUT (continued)</b>						
Input Current		AIN <sub>n</sub> , MAX180			±1.0	μA
		ADCIN, MAX181			±0.1	
Mux-On Resistance	R <sub>ON</sub>	AIN <sub>n</sub> = 2.5V, I <sub>MUXOUT</sub> = 1.25mA, MAX181			2	kΩ
Mux-On Leakage Current	I <sub>ON</sub>	AIN <sub>n</sub> = MUXOUT = ±5V, MAX181			±100	nA
Mux-Off Leakage Current	I <sub>IN (OFF)</sub>	AIN <sub>n</sub> = ±5V, V <sub>OUT</sub> = ±5V, MAX181			±100	nA
	I <sub>OUT (OFF)</sub>	AIN <sub>n</sub> = ±5V, V <sub>OUT</sub> = ±5V, MAX181			±100	
Input Capacitance (Note 5)	C <sub>IN</sub>	AIN <sub>n</sub> , ADCIN		25	35	pF
		MUXOUT		35	45	
<b>REFERENCE INPUT</b>						
Input Range (Note 5)			-4.92	-5.00	-5.08	V
Input Current					-2	mA
Input Resistance			2.5			kΩ
<b>REFERENCE OUTPUT</b>						
VREF Output Voltage		T <sub>A</sub> = +25°C	-4.98	-5.00	-5.02	V
VREF Output Tempco (Note 6)		MAX18_A/B			25	ppm/°C
		MAX18_C			45	
VREF Load Regulation (Note 7)		I <sub>OUT</sub> = 0mA to 5mA, T <sub>A</sub> = +25°C		0.2	1.0	mV/mA
<b>REFADJ, OFFADJ</b>						
Input Current		V <sub>REFADJ</sub> , V <sub>OFFADJ</sub> = V <sub>DD</sub> to REF <sub>IN</sub>			±1	μA
Disable Threshold			4.5			V
REFADJ Adjustment Range		REF <sub>IN</sub> < REFADJ < AGND	±60	±80		mV
OFFADJ Adjustment Range		REF <sub>IN</sub> < OFFADJ < AGND	±15	±25		LSB
<b>LOGIC INPUTS</b>						
Input Low Voltage	V <sub>IL</sub>	MODE			0.5	V
		CS, RD, WR, CLK, A2-A0, DIFF, BIP, HBEN			0.8	
Input High Voltage	V <sub>IH</sub>	MODE		4.5		V
		CS, RD, WR, CLK, A2-A0, DIFF, BIP, HBEN		2.4		
Input Mid-Level Voltage	V <sub>MID</sub>	MODE		1.5	3.5	V
Input Floating Voltage	V <sub>FLT</sub>	MODE			2.5	V
Input Current	I <sub>IN</sub>	MODE	T <sub>A</sub> = +25°C	±50	±100	μA
			T <sub>A</sub> = T <sub>MIN</sub> to T <sub>MAX</sub>	±50	±100	
		CS, RD, WR, CLK, A2-A0, DIFF, BIP, HBEN	T <sub>A</sub> = +25°C		±1	
			T <sub>A</sub> = T <sub>MIN</sub> to T <sub>MAX</sub>		±10	
Input Capacitance (Note 5)	C <sub>IN</sub>				15	pF
<b>LOGIC OUTPUTS</b>						
Output Low Voltage	V <sub>OL</sub>	D11-D0, BUSY, RDY, I <sub>SINK</sub> = 1.6mA			0.4	V
Output High Voltage	V <sub>OH</sub>	D11-D0, BUSY, RDY, I <sub>SOURCE</sub> = 360μA	4.0			V
Floating State Leakage Current	I <sub>LKG</sub>	D11-D0, V <sub>OUT</sub> = 0V to V <sub>DD</sub>			±10	μA
Floating State Output Capacitance (Note 5)	C <sub>OUT</sub>				15	pF

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

## ELECTRICAL CHARACTERISTICS (continued)

(V<sub>DD</sub> = +5V ±5%, V<sub>SS</sub> = -12V ±5% or -15V ±5%, REFIN = -5V, Internal Reference Mode, Bipolar Mode, Slow-Memory Mode (see text), I<sub>CLK</sub> = 1.6MHz external, MAX180/MAX181 all grades, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>POWER REQUIREMENTS</b>						
Supply Voltage (Note 1)	V <sub>DD</sub>		4.75	5.00	5.25	V
	V <sub>SS</sub>		-11.40		-15.75	
Supply Current	I <sub>DD</sub>	V <sub>DD</sub> = 5V		4.5	7.0	mA
	I <sub>SS</sub>	V <sub>SS</sub> = -12V		7.0	10.0	
Power Dissipation	PD	V <sub>DD</sub> = 5V, V <sub>SS</sub> = -15V		110	155	mW
Power-Supply Rejection, with Internal Reference	PSR	Input near FS, V <sub>SS</sub> = -12V, V <sub>DD</sub> = 4.75V to 5.25V		±1/2	±1	LSB
		Input near FS, V <sub>DD</sub> = 5V, V <sub>SS</sub> = -14.25V to -15.75V		±1/8	±1/2	
		Input near FS, V <sub>DD</sub> = 5V, V <sub>SS</sub> = -11.4V to -12.6V		±1/8	±1/2	

## TIMING CHARACTERISTICS

(V<sub>DD</sub> = +5V, V<sub>SS</sub> = -12V, I<sub>CLK</sub> = 1.6MHz, Internal Reference Mode, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 8)

PARAMETER	SYMBOL	CONDITIONS	T <sub>A</sub> = +25°C			MAX18_C/E			MAX18_M			UNITS
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
CS to $\overline{RD}$ Setup time	t <sub>1</sub>	Note 5	0			0			0			ns
CS to $\overline{RD}$ Hold time	t <sub>2</sub>		0			0			0			ns
CS to $\overline{WR}$ Setup time	t <sub>3</sub>		0			0			0			ns
CS to $\overline{WR}$ Hold time	t <sub>4</sub>	Note 5	0			0			0			ns
$\overline{WR}$ Low Pulse Width	t <sub>5</sub>		120			120			120			ns
$\overline{WR}$ High Pulse Width	t <sub>6</sub>	MODE = 0 or 1 Note 5	200			200			200			ns
DATA IN to $\overline{WR}$ Setup Time	t <sub>7</sub>		80			100			120			ns
DATA IN to $\overline{WR}$ Hold Time	t <sub>8</sub>		0			0			0			ns
$\overline{WR}$ Rising to $\overline{BUSY}$ Delay	t <sub>9</sub>	C <sub>L</sub> = 50pF, MODE = 1			160			180			200	ns
$\overline{WR}$ Falling to $\overline{BUSY}$ Delay	t <sub>10</sub>	C <sub>L</sub> = 50pF, MODE = open			220			260			280	ns
$\overline{RD}$ Low Pulse Width	t <sub>11</sub>		100			130			150			ns
$\overline{RD}$ High Pulse Width	t <sub>12</sub>	Note 5	200			200			200			ns
DATA IN to $\overline{RD}$ Setup Time	t <sub>13</sub>		80			100			120			ns
DATA IN to $\overline{RD}$ Hold Time	t <sub>14</sub>		0			0			0			ns
$\overline{RD}$ to $\overline{BUSY}$ Fall Delay	t <sub>15</sub>	C <sub>L</sub> = 50pF			150			170			200	ns
$\overline{RD}$ to Data out Valid	t <sub>16</sub>	C <sub>L</sub> = 100pF Note 9		50	100			130			150	ns
$\overline{RD}$ to Data out Three-State	t <sub>17</sub>	Notes 9, 10		30	50			65			75	ns
HBEN to $\overline{RD}$ or $\overline{WR}$ Setup Time	t <sub>18</sub>		80			100			120			ns
HBEN to $\overline{RD}$ or $\overline{WR}$ Hold Time	t <sub>19</sub>		0			0			0			ns
CS to $\overline{READY}$ Fall Delay	t <sub>20</sub>	C <sub>L</sub> = 50pF			110			130			150	ns

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

**MAX180/MAX181**

## TIMING CHARACTERISTICS (continued)

(V<sub>DD</sub> = +5V, V<sub>SS</sub> = -12V, f<sub>CLK</sub> = 1.6MHz, Internal Reference Mode, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 8)

PARAMETER	SYMBOL	CONDITIONS	T <sub>A</sub> = +25°C			MAX18_C/E			MAX18_M			UNITS
			MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
BUSY to Data Out Valid	t <sub>21</sub>	C <sub>L</sub> = 100pF, Note 9			125			150			170	ns
CS, RD, or WR to CLK Setup time for 15 clock conversion	t <sub>22</sub>	Note 5	220			220			220			ns
CS, RD, or WR to CLK Setup time for 16 clock conversion	t <sub>23</sub>	Note 5	0			0			0			ns

**Note 1:** Performance at power-supply tolerance limits guaranteed by power-supply rejection test.

**Note 2:** V<sub>DD</sub> = +5V, V<sub>SS</sub> = -15V, FS = +5V, REF<sub>IN</sub> = -5V.

**Note 3:** Typical change over temperature is ±1LSB.

**Note 4:** FS Tempco = ΔFS/ΔT, where ΔFS is full-scale change from T<sub>A</sub> = +25°C to T<sub>MIN</sub> or to T<sub>MAX</sub>.

**Note 5:** Guaranteed by design.

**Note 6:** REF<sub>IN</sub> TC = ΔREF<sub>IN</sub>/ΔT, where ΔREF<sub>IN</sub> is reference voltage change from T<sub>A</sub> = +25°C to T<sub>MIN</sub> or to T<sub>MAX</sub>.

**Note 7:** Load current should remain constant during conversion. This current is in addition to the DAC input current.

**Note 8:** All inputs are 0V to +5V swing with t<sub>r</sub> = t<sub>f</sub> = 5ns (10% to 90% of 5V) and timed from a voltage level of +1.6V.

**Note 9:** t<sub>16</sub> and t<sub>21</sub> are measured with the load circuits of Figure 1 and defined as the time required for an output to cross 0.8V or 2.4V.

**Note 10:** t<sub>17</sub> is defined as the time required for the data lines to change 0.5V when the circuit load is as shown in Figure 2.

## Pin Description

NAME	MAX180		MAX181		FUNCTION
	DIP	PLCC	DIP	PLCC	
AIN0-AIN5	1-6	2-7	1-6	2-7	Analog Inputs to the mux: 0V to +5V unipolar, -2.5V to +2.5V bipolar
AIN6-AIN7	7,8	8,9			Analog Inputs to the mux: 0V to +5V unipolar, -2.5V to +2.5V bipolar
MUXOUT			7	8	Multiplexer Output
ADCIN			8	9	Analog Input to track-and-hold
REF <sub>IN</sub>	9	10	9	10	Reference Input
AGND	10	11	10	11	Analog Ground
REFOUT	11	13	11	13	-5V Reference Output
REFADJ	12	14	12	14	-5V Reference Adjust. Connect to V <sub>DD</sub> if not required.
OFFADJ	13	15	13	15	Offset Adjust. Connect to V <sub>DD</sub> if not required.
MODE	14	16	14	16	Interface Mode Select pin.
V <sub>SS</sub>	15	17	15	17	Negative Supply: -15V or -12V
D11-DB	16-19	18-21	16-19	18-21	Three-State Data Outputs, MSB = D11
DGND	20	22	20	22	Digital Ground
D7-D0	21-28	24-31	21-28	24-31	Three-State Data Outputs, LSB = D0
CLKIN	29	32	29	32	Clock Input, TTL/CMOS compatible
HBEN	30	33	30	33	High-Byte Enable Input
RD	31	35	31	35	READ Input
WR	32	36	32	36	WRITE Input (MODE = 1 or Open) READY Output (MODE = 0)
CS	33	37	33	37	CHIP-SELECT Input
BUSY	34	38	34	38	BUSY Output
DIFF	35	39	35	39	Single-Ended Mode: DIFF = 0. Differential Mode: DIFF = 1
BIP	36	40	36	40	Unipolar Mode: BIP = 0, Bipolar Mode: BIP = 1
A0-A2	37-39	41-43	37-39	41-43	Multiplexer Channel Address Input: A2 = MSB, A0 = LSB
V <sub>DD</sub>	40	44	40	44	Positive Supply: +5V Input (substrate connected to V <sub>DD</sub> )
N.C.		1,12, 23,34		1,12, 23,34	No Connect. No internal connection. Leave pin open or connect to AGND.

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

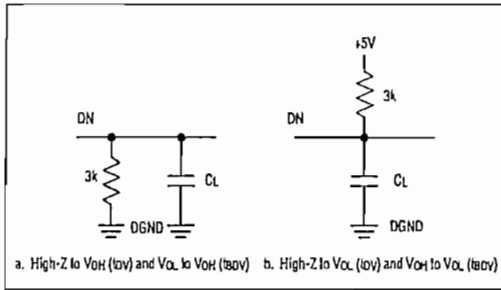


Figure 1. Load Circuits for Access Time

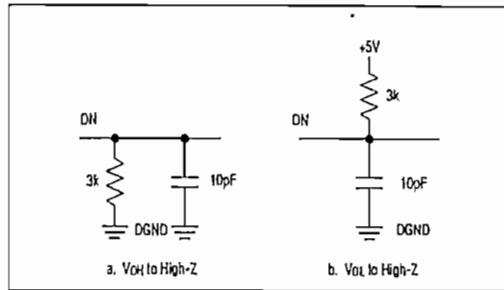


Figure 2. Load Circuits for Bus-Relinquish Time

## A/D Converter Operation

The MAX180/MAX181 use successive approximation and input track-and-hold (T/H) circuitry to convert an analog signal to a series of 12-bit digital output codes. The control logic interfaces easily to  $\mu$ Ps, requiring only a few passive components for most applications. The T/H does not require an external capacitor. Figure 3 shows the MAX180 typical operating circuit.

### Starting a Conversion

Regardless of the mode or interface selected, the following sequence occurs once conversion is started:

1. The data inputs that configure the data-acquisition system (DAS) latch, and the interface signals the  $\mu$ P that a conversion has started.
2. The mux directs the selected input signal to the T/H input.
3. A fixed time delay allows the T/H to acquire the signal. In all modes except asynchronous hold, this delay is 3 clock cycles. In asynchronous hold, the  $\mu$ P controls this delay.
4. The T/H switches to hold mode. The T/H output delivers a stable, single-ended sample of the input signal to the A/D input.
5. The successive approximation cycle begins. The ADC tests and sets each of the 12 bits in turn, from most to least significant. Bit decisions occur on the CLKIN falling edges, for a total of 12 clock cycles.
6. Output data is latched by the output registers, and the interface signals the  $\mu$ P that conversion is complete and data is available.

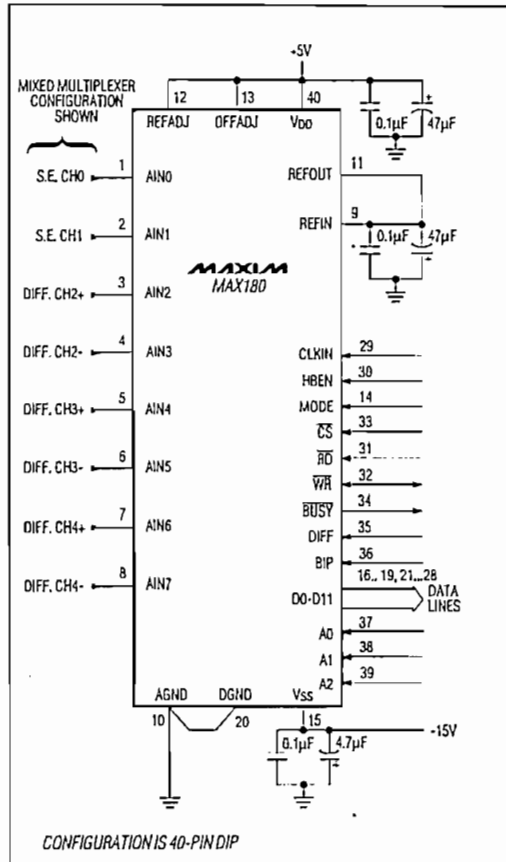


Figure 3. MAX180 Typical Operating Circuit



# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

## Analog Input - Track-and-Hold

Figure 4 shows the equivalent input circuit, illustrating the sampling architecture of the ADC's analog comparator. The input capacitance acts as the hold capacitor and is charged by the input signal with every A/D conversion. The capacitance is charged through an internal 1kΩ resistor in series with the input. Note: Figure 4's switches represents both the mux and hold switches.

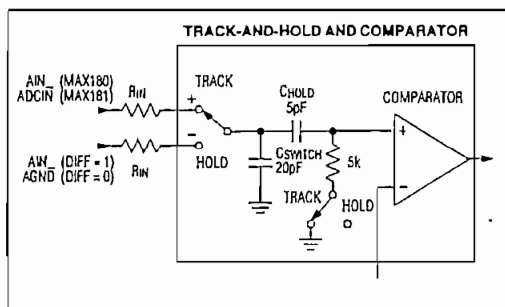


Figure 4. Equivalent Input Circuit

When in single-ended input mode and between conversions ( $BUSY = \text{High}$ ), the selected analog input is connected to the hold capacitor (track mode). When a conversion starts,  $CHOLD$  disconnects from the + T/H input, thus sampling the input (see "Digital Interface" section for per-chip T/H timing). When the switch closes at conversion end,  $CHOLD$  reconnects to the input and charges to the input signal. The loading effect of the analog inputs on the signal is such that a high-speed input buffer is usually NOT needed because the ADC disconnects from the input during the actual conversion.

The previous explanation applies for the differential input mode if "input" is replaced by  $AIN+$  and "analog ground" is replaced by  $AIN-$ . In the differential input mode,  $A0-A2$  select the input channel pairs (Table 1). Only the signal side of the input channel is held by the T/H; the return side must remain stable within  $\pm 0.5LSB$  ( $\pm 0.1LSB$  for best results) during the conversion. For example, a common-mode signal of 0.33Vp-p at 60Hz results in a maximum error of 0.5LSB.

The T/H starts tracking when the ADC is deselected ( $BUSY = \text{High}$ ). Hold mode begins 3 clock cycles after a conversion is initiated in all but the Asynchronous Hold Mode. Variation in hold-mode delay from one conversion to the next (aperture jitter) is less than 100ps. Figures 7-11 detail the T/H and interface timing for the various interface modes.

The time required for the T/H to acquire an input signal is a function of how quickly the input capacitance is charged. If the input source impedance is high, the acquisition time lengthens and more time must be allowed between conversions. Acquisition time is calculated by:

$$t_{ACQ} = 10(R_S + R_{IN})20pF \text{ (but never less than } 1.875\mu s)$$

where  $R_{IN} = 1k\Omega$ , and  $R_S =$  source impedance of the input signal.

## Input Bandwidth

The A/D's input tracking circuitry is excellent for tracking large signals and wide bandwidths and does not exhibit the slew-rate limitations of many other ADC T/Hs. The MAX180/MAX181 T/H's full-power bandwidth is typically 6MHz; this allows the measurement of periodic signals with bandwidths exceeding the ADC's sample rate (100kHz) using under-sampling techniques. Important note: If under-sampling is used to measure high-frequency signals, take special care to avoid aliasing errors. Without adequate input filtering, high-frequency noise could be aliased into the measurement band.

## Reference

The MAX180/MAX181 operate with either the internal reference or an external -5V reference. In both cases,  $REFIN$  must be bypassed to  $AGND$  with a 47μF electrolytic capacitor in parallel with a 0.1μF ceramic capacitor to minimize noise and maintain a low impedance at high frequencies.  $REFIN$  is connected directly to the internal DAC, and the current load varies between 0mA and 1mA during conversion.

## Internal Reference

The internal reference is buffered through an amplifier whose output connects to  $REFOUT$ . To operate the MAX180/MAX181 with the internal reference, connect  $REFIN$  to  $REFOUT$ . Do not connect a resistor between the bypass capacitors and  $REFIN$ . The reference buffer amplifier can sink 5mA for external loads. Adjust the reference output at  $REFADJ$  (Figure 14).

## External Reference

With a -5V external reference, bypass  $REFIN$  to  $AGND$  with a 47μF electrolytic capacitor in parallel with a 0.1μF ceramic capacitor. The reference source impedance must be less than 0.2Ω and must be able to sink the internal DAC load of 1mA. Connect  $REFOUT$  to  $V_{SS}$  and  $REFADJ$  to  $V_{DD}$  to prevent noise. If  $REFIN$  is driven above  $AGND$  during power sequencing, latchup can occur. Connect a Schottky clamp diode (1N5817) to prevent  $REFIN$  from substantially exceeding  $AGND$ .

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

Table 1. Address vs. Channel Selection (see Figure 4)

	A2	A1	A0	SE/DIFF	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	COM	
MAX180/MAX181	0	0	0	0	+								-	
MAX180/MAX181	0	0	1	0		+							-	
MAX180/MAX181	0	1	0	0			+						-	
MAX180/MAX181	0	1	1	0				+					-	
MAX180/MAX181	1	0	0	0					+				-	
MAX180/MAX181	1	0	1	0						+			-	
MAX180	1	1	0	0							+		-	
MAX181	1	1	0	0	MUXOUT CONNECTED TO AGND									+,-
MAX180	1	1	1	0								+	-	
MAX181	1	1	1	0	CH 0-5, AND MUXOUT ARE OPEN									-
MAX180/MAX181	0	0	0	1	+	-								
MAX180/MAX181	0	0	1	1	-	+								
MAX180/MAX181	0	1	0	1			+	-						
MAX180/MAX181	0	1	1	1			-	+						
MAX180/MAX181	1	0	0	1					+	-				
MAX180/MAX181	1	0	1	1					-	+				
MAX180	1	1	0	1							+	-		
MAX180	1	1	1	1							-	+		
MAX181	1	1	0	1	MUXOUT CONNECTED TO AGND									+,-
MAX181	1	1	1	1	CH 0-5, AND MUXOUT ARE OPEN									-

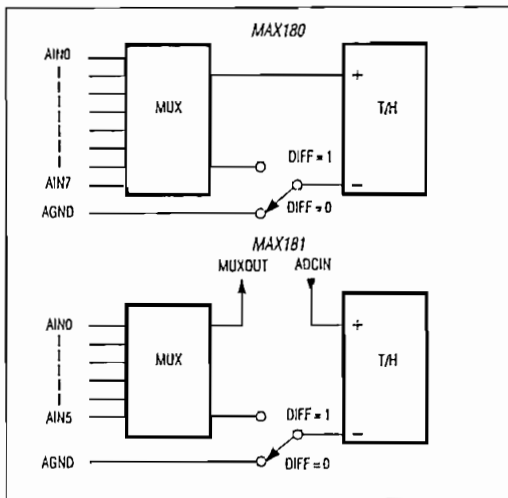


Figure 5. Multiplexer channel configuration

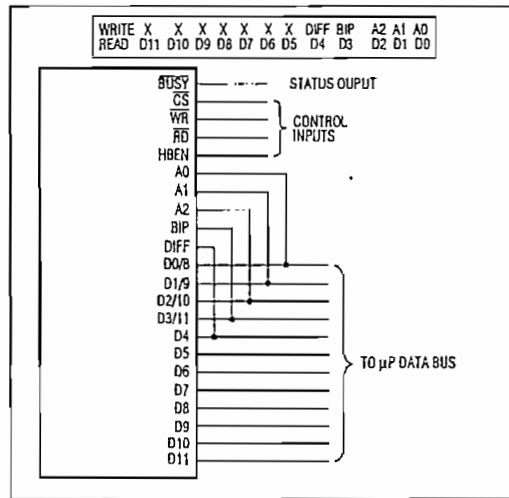


Figure 6. Input/Output Port Mode (12-Bit-Wide Data Bus Shown)

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

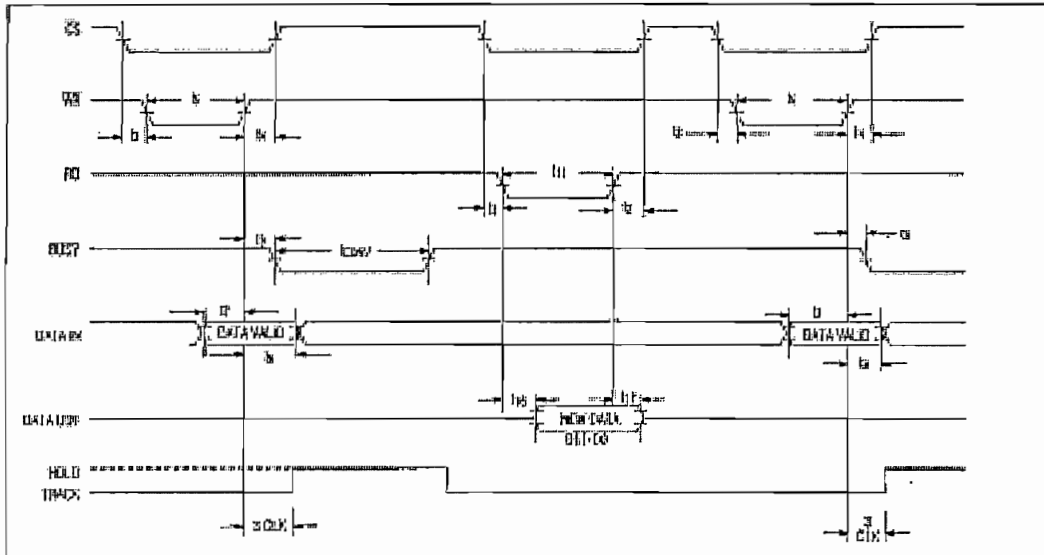


Figure 7a. Input/Output Port-Mode timing, parallel read (MODE = 1, HSEN = 0).

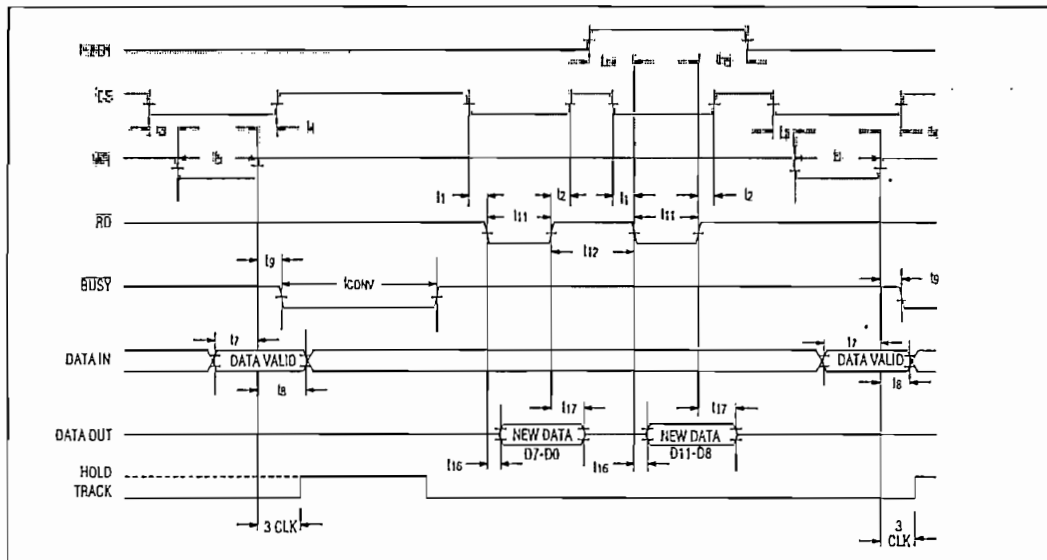


Figure 7b. Input/Output Port-Mode timing, two-byte read (MODE = 1).

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

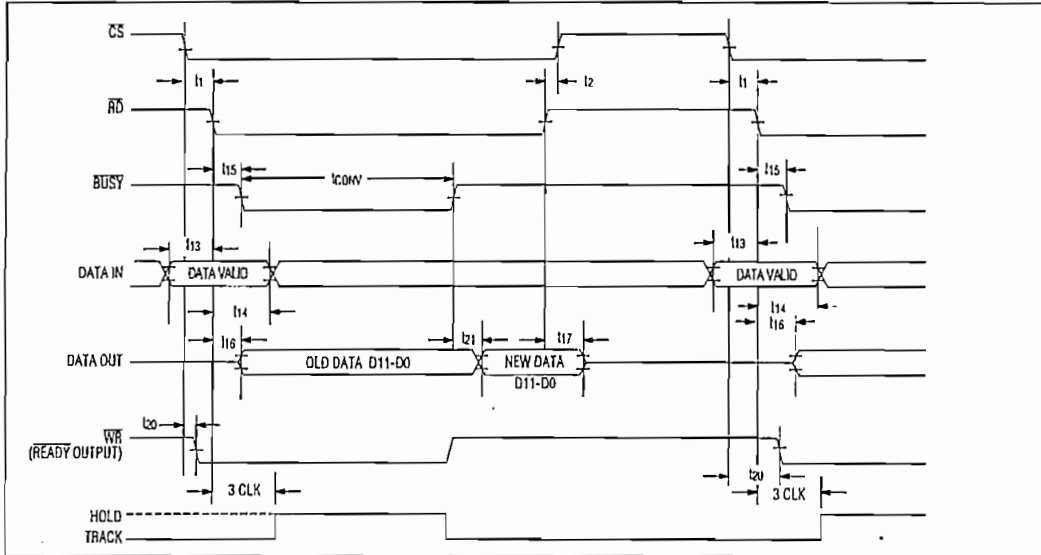


Figure 8a. Slow Memory Mode timing, parallel read (MODE = 0, HBEN = 0).

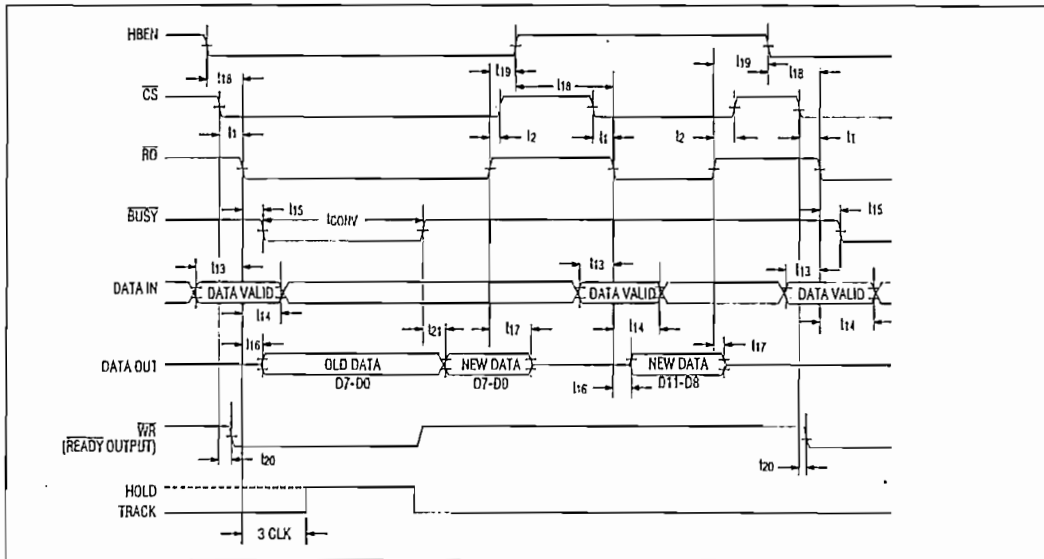


Figure 8b. Slow Memory Mode timing, two-byte read (MODE = 0).

## Complete, 8-Channel, 12-Bit Data-Acquisition Systems

**MAX180/MAX181**

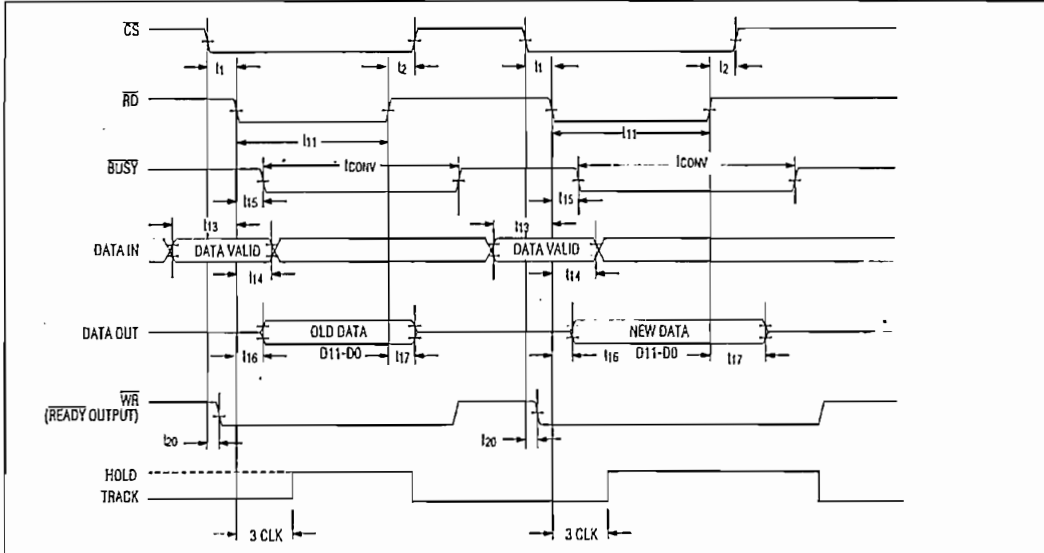


Figure 9a. ROM Mode timing, parallel read ( $MODE = 0$ ,  $HBEN = 0$ ).

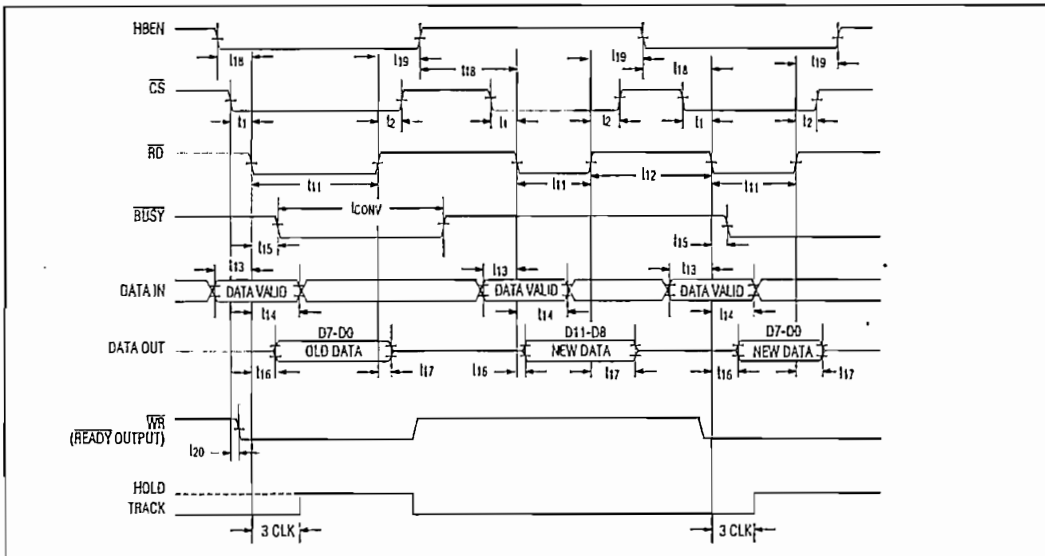


Figure 9b. ROM Mode timing, two-byte read ( $MODE = 0$ ).

**Complete, 8-Channel, 12-Bit Data-Acquisition Systems**

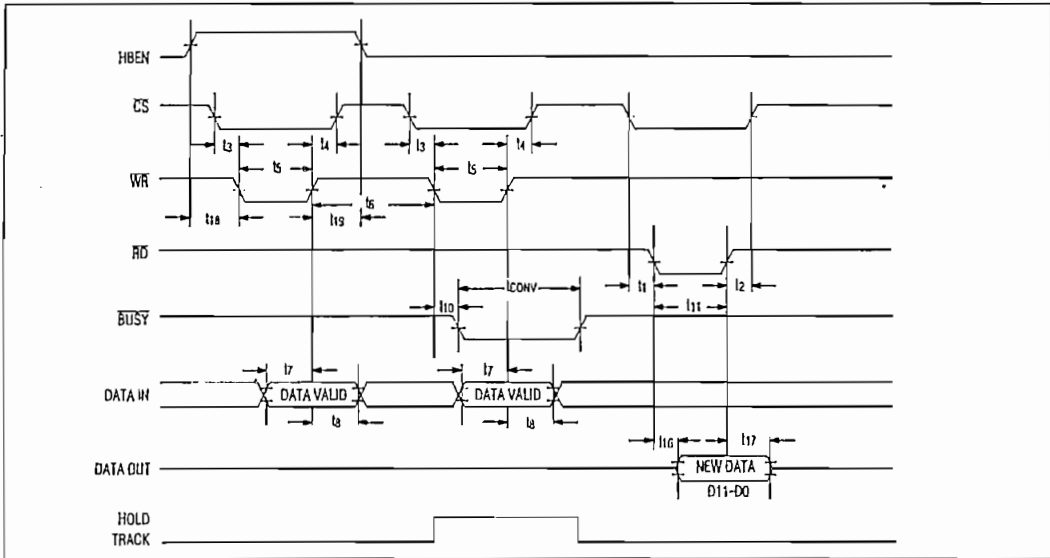


Figure 10a. Asynchronous Hold Mode timing, parallel read (MODE = open circuit)

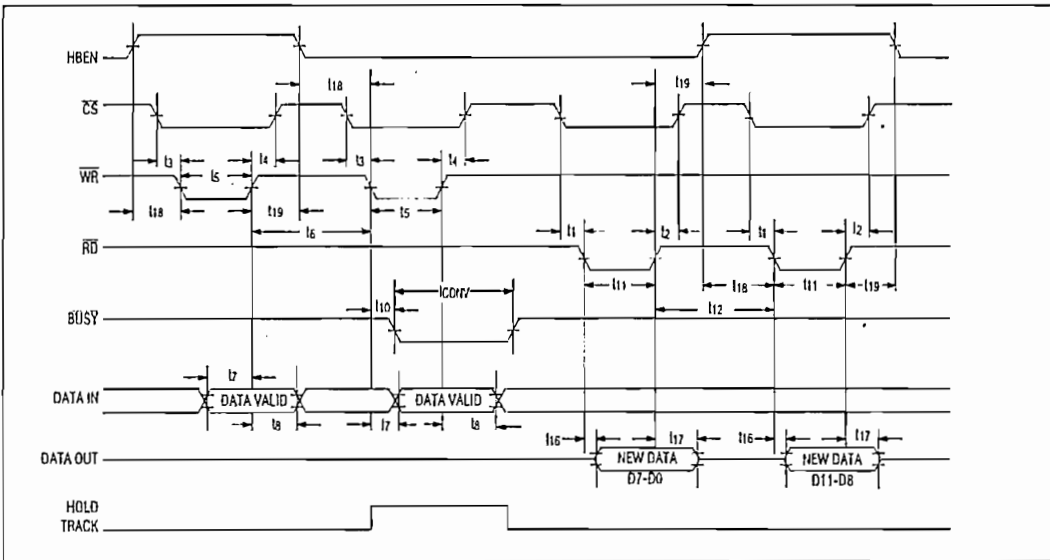


Figure 10b. Asynchronous Hold Mode timing, two-byte read (MODE = open circuit)

## Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

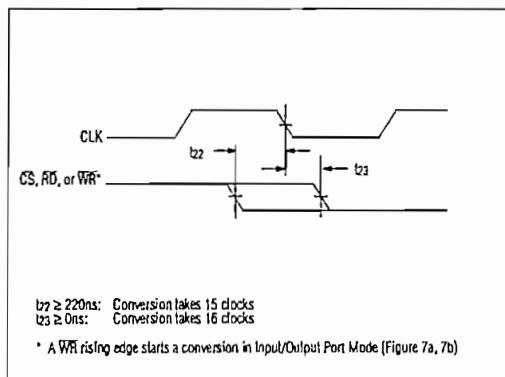


Figure 11.  $\overline{\text{CS}}$ ,  $\overline{\text{RD}}$ , or  $\overline{\text{WR}}$  to CLK Setup and Hold Time for Synchronous Operation

### Digital Interface

#### Input/Output Port Mode (MODE = 1)

In this mode, data inputs and outputs are usually connected together (Figure 6), and the  $\mu\text{P}$  writes the configuration data to the DAS internal register with a write instruction (Figure 7). This starts a conversion, as indicated by the high-to-low transition of  $\overline{\text{BUSY}}$ . The mux connects the selected input channel to the T/H, which acquires the signal during the first 3 clock cycles. On the falling edge of the 3rd clock, the T/H switches to hold mode, and the A/D conversion starts. 15 clock cycles after  $\overline{\text{WR}}$  goes high,  $\overline{\text{BUSY}}$  goes high, and the conversion result latches into three-state output buffers. The  $\mu\text{P}$  can then access the conversion result with a read instruction. For 16-bit bus operation,  $\text{HBEN} = 0$ , and the 12-bit result is read directly. For 8-bit bus operation,  $\text{HBEN} = 0$  during the conversion, and the read instruction returns the 8 LSBs. A second read with  $\text{HBEN} = 1$  returns the 4 MSBs in the low nibble. Note: In any mode,  $\text{HBEN} = 1$  disables conversion start.

The DAS internal register is 5 bits wide: 3 bits for the analog-channel address, 1 bit for single-ended/differential mux operation, and 1 bit for unipolar/bipolar A/D operation.

#### Slow Memory Mode (MODE = 0)

The DAS appears to the  $\mu\text{P}$  as memory or as a slow peripheral in memory mode. The 5 configuration bits can be preset by an external data latch, a decoded device address, or any external selection logic. A

read instruction initiates a conversion as shown in Figure 8. In this mode, the  $\overline{\text{WR}}$  input functions as the  $\overline{\text{RDY}}$  output and goes low when  $\overline{\text{CS}}$  goes low.  $\overline{\text{BUSY}}$  goes low after  $\overline{\text{RD}}$  goes low, indicating the beginning of a signal acquisition cycle, and can be used to place the  $\mu\text{P}$  into a wait state. When the conversion is complete,  $\overline{\text{BUSY}}$  releases the  $\mu\text{P}$  from its wait state. The  $\mu\text{P}$  can then access the conversion result with a read instruction. For 16-bit bus operation,  $\text{HBEN} = 0$ , and the 12-bit result is read directly. For 8-bit bus operation,  $\text{HBEN} = 0$  during the conversion, and the read instruction returns the 8 LSBs. A second read with  $\text{HBEN} = 1$  returns the 4 MSBs in the low nibble. Note: In any mode,  $\text{HBEN} = 1$  disables conversion start.

#### ROM Mode, Parallel Read (MODE = 0)

ROM mode avoids using  $\mu\text{P}$  wait states. A conversion starts with a read instruction, and the 12 data bits from the previous conversion appear at D11-D0. The data from the first read in a sequence is often disregarded when ROM mode is used. A second read accesses the results of the first conversion and starts a new conversion. The time between successive reads must be longer than the conversion time of the MAX180/MAX181 (Figure 9a, 16-bit bus).

#### ROM Mode, 2-Byte Read (MODE = 0)

As in memory mode, only D7-D0 are used for a 2-byte read. A conversion starts with a read instruction when  $\text{HBEN}$  is low. At this point, the data outputs contain the 8 LSBs from the previous conversion. Two more read operations are needed to access the conversion result. The first, with  $\text{HBEN}$  high, accesses the 4 MSBs with 4 leading zeros. The second read, with  $\text{HBEN}$  low, outputs the 8 LSBs and starts a new conversion. Figure 9b (8-bit bus) details this mode.

#### Asynchronous Hold Mode (MODE = Open)

Asynchronous hold mode is helpful when a precise or repeatable sample timing is required. Asynchronous hold is very similar to the I/O port mode, except two write instructions are required. The first write, with  $\text{HBEN} = 1$ , configures the MAX180/MAX181 and connects the selected channel to the T/H input; the second write, with  $\text{HBEN} = 0$ , places the T/H into hold and starts the conversion. In other words, the three-clock cycle delay for T/H acquisition can be changed by controlling when the second write instruction occurs. The falling edge of the second  $\overline{\text{WR}}$  pulse places the T/H into hold (Figure 10).

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

## External Clock

The range for the external clock duty cycle is between 20% and 80%. A precise square wave is not required.

## Clock and Control Synchronization

For best analog performance, the MAX180/MAX181 clock should be synchronized to the  $\overline{RD}$ ,  $\overline{WR}$ , and  $\overline{CS}$  inputs (Figure 11) with at least 100ns separating convert start from the nearest clock edge. This synchronization ensures that transitions at CLKIN are not coupled to the analog input and sampled by the T/H. The magnitude of this feedthrough is only a few millivolts. If CLKIN and convert start ( $\overline{CS}$ ,  $\overline{WR}$  and  $\overline{RD}$ ) are asynchronous, frequency components caused by mixing of the clock and convert signals can increase the apparent input noise.

When the clock and convert signals are synchronized, small end-point errors (offset and full-scale) are the most that can be generated by clock feedthrough, but even these errors are eliminated by ensuring that the start of a conversion ( $\overline{RD}$  or  $\overline{WR}$  and  $\overline{CS}$  falling edge) does not occur within 100ns of a clock transition (Figure 11).

## Output Data Format

The 12 data bits can be output either in full parallel or as two 8-bit bytes. Table 2 shows the data-bus output format. To obtain parallel output for 16-bit  $\mu$ Ps, HBEN is tied low. Note: The output data, D11-D0, is right-justified (i.e. D0, the LSB, is the right-most bit in the 16-bit word).

A two-byte read makes use of outputs D7-D0. Byte selection is controlled by HBEN, which multiplexes the data outputs. When HBEN is low, the lower 8 bits appear at the data outputs. When HBEN is high, the upper 4 bits appear at D0-D3 with the leading 4 bits low in locations D4-D7. Note: The 4 MSBs always appear at D11-D8 when the outputs are enabled, regardless of the state of HBEN.

## Application Hints

### Initialization After Power-Up

In some applications, power is removed from the ADC during periods of inactivity to conserve power. This is increasingly common in battery-powered systems. To initialize the MAX180/MAX181 at power-up, execute a read operation with HBEN low, ignoring the data outputs.

### Minimizing System-Induced Noise

The MAX180/MAX181 are insensitive to most noise sources, especially when the layout, bypass, and grounding recommendations are followed. The following practices should also be considered:

1. Minimize digital activity during conversion, especially activity that is asynchronous with the MAX180/MAX181 clock.
2. Avoid data-bus activity within  $\pm 20$ ns of the CLKIN falling edge.

If the data bus connected to the ADC is active during a conversion, coupling from the data pins to the ADC comparator can cause errors. Using slow-memory mode avoids this problem by placing the  $\mu$ P in a wait state during the conversion. In ROM mode, the bus should be isolated from the ADC using three-state drivers if the data bus is active during the conversion.

In ROM mode, the ADC generates considerable digital noise when  $\overline{RD}$  or  $\overline{CS}$  go high and the output data drivers are disabled after conversion start. This noise can affect the ADC comparator and cause large errors if it coincides with the SAR latching a comparator decision. To prevent this,  $\overline{RD}$  and  $\overline{CS}$  should be active for less than one clock cycle. If this is not possible,  $\overline{RD}$  or  $\overline{CS}$  should go high on a rising edge of CLKIN because the comparator output is latched on the falling edge of CLKIN.

Table 2. Data-Bus Output,  $\overline{CS}$  &  $\overline{RD}$  = LOW

DIP Pin #	Pin 16	Pin 17	Pin 18	Pin 19	Pin 21	Pin 22	Pin 23	Pin 24	Pin 25	Pin 26	Pin 27	Pin 28
Pin Label*	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
HBEN = LOW**	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
HBEN = HIGH**	D11	D10	D9	D8	LOW	LOW	LOW	LOW	D11	D10	D9	D8

Note: \* D11-D0 are the ADC data output pin names.  
 \*\* D11-D0 are the 12-bit conversion results. D11 is the MSB.



# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

## Layout, Grounding, Bypassing

Use printed circuit boards for best system performance; wire-wrap boards are not recommended. The board layout should ensure that digital- and analog-signal lines are separated as much as possible. Do not run analog and digital (especially clock) lines parallel to one another or digital lines underneath the ADC package.

Figure 12a shows the recommended system-ground connections. A single-point analog STAR ground should be established at AGND, separate from the logic ground. All other analog grounds and DGND should be connected to this STAR ground, and no other digital system grounds should be connected here. For noise-free operation, the ground return to the power supply from this STAR ground should be low impedance and as short as possible.

The ADC's high-speed comparator is sensitive to high-frequency noise in the  $V_{DD}$  and  $V_{SS}$  power supplies. These supplies should be bypassed to the analog STAR ground with  $0.1\mu\text{F}$  and  $47\mu\text{F}$  bypass capacitors. Minimize capacitor lead length for best supply-noise rejection. If the  $5\text{V}$  power supply is very noisy, connect a small ( $10\Omega$ ) resistor to filter the noise (Figure 12b).

## Gain and Offset Adjustment

Figure 13 plots the nominal unipolar I/O transfer function of the MAX180/MAX181. Code transitions occur halfway between successive integer LSB values. Output coding for unipolar operation is natural binary with  $1\text{LSB} = 1.22\text{mV}$  ( $5\text{V}/4096$ ). Figure 14 shows the bipolar-input transfer function, where output coding is twos-complement.

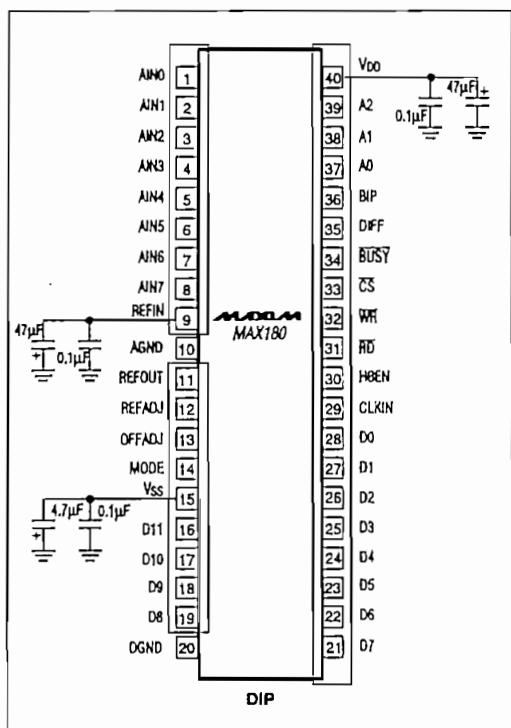


Figure 12a. Recommended Grounding and Ground Plane

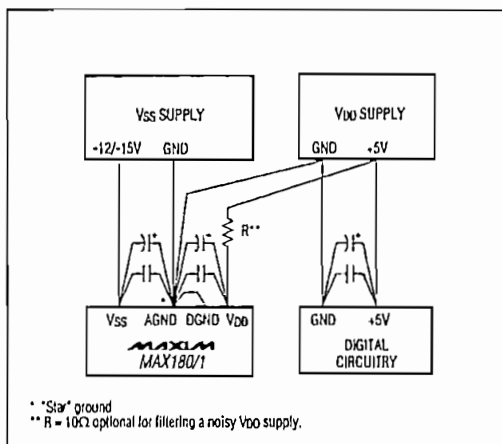


Figure 12b. Power-Supply Grounding

## Complete, 8-Channel, 12-Bit Data-Acquisition Systems

If offset and gain adjustments are not desired, connect OFFADJ and REFADJ to VDD. Figure 15's circuit provides  $\pm 1.2\%$  ( $\pm 50$  LSBs) of adjustment range for gain and  $\pm 0.44\%$  ( $\pm 18$  LSBs) of adjustment range for offset. This is ideal for applications that require gain (full-scale range) or offset adjustment. If the adjustment inputs are used, bypass to AGND with a  $0.1\mu\text{F}$  capacitor. Offset should be adjusted before gain. For the 0V to 5V input range, apply LSB (0.61mV) to the analog input, and adjust R1 so the digital output code changes between 0000 0000 0000 and 0000 0000 0001. To adjust full scale, apply FS - 1LSB (4.99817V), and adjust R2 until the output code changes between 1111 1111 1110 and 1111 1111 1111. There may be a slight interaction between the adjustments.

To adjust bipolar ( $\pm 2.5\text{V}$ ) offset, apply LSB (0.61mV) to the analog input, and adjust R1 until the output code switches between 0000 0000 0000 and 0000 0000 0001. For full scale, apply FS - 1LSB (2.49817V) to the input, and adjust R2 so the output code switches between 0111 1110 and 0111 1111 1111 (Figure 15). There may be some interaction between these adjustments. If an external reference is used, adjust gain by varying the value of the reference instead of R2.

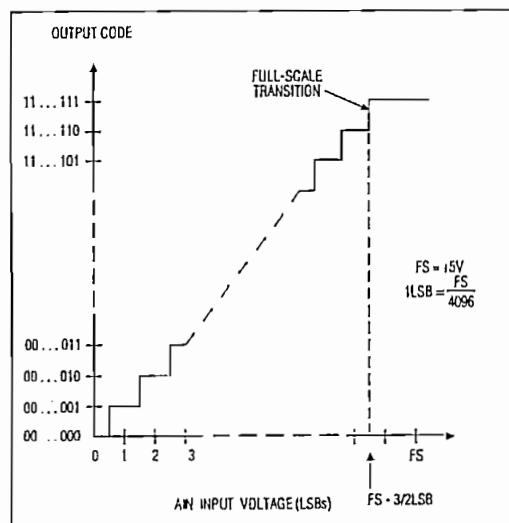


Figure 13. MAX180/MAX181 Unipolar Transfer Function

### Dynamic Performance

Wide-bandwidth analog input and 100kHz throughput make the MAX180/MAX181 ideal for wideband-signal processing. To support these and other related applications, fast Fourier transform (FFT) test techniques guarantee the ADC's dynamic frequency response, distortion, and noise at the rated throughput. Specifically, this involves applying a low-distortion sine wave to the ADC input and recording the digital conversion results for a specified time. The data is then analyzed using an FFT algorithm that determines its spectral content. Conversion errors are seen as spectral elements outside of the fundamental input frequency.

ADCs have traditionally been evaluated by specifications such as zero and full-scale error and integral (INL) and differential (DNL) nonlinearity. Such parameters are widely accepted for specifying performance with DC and slowly varying signals, but less useful in signal-processing applications where the ADC's impact on the system transfer function is the main concern. The significance of the various DC parameters does not translate well to the dynamic case, so different tests are required.

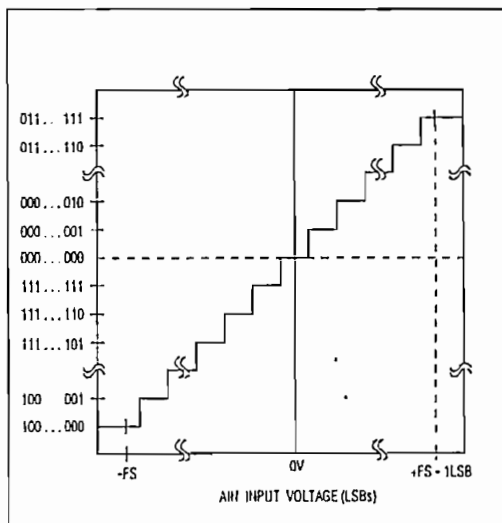


Figure 14. MAX180/MAX181 Bipolar Transfer Function

# Complete, 8-Channel, 12-Bit Data-Acquisition Systems

MAX180/MAX181

## Signal-to-Noise Ratio and Effective Number of Bits

Signal-to-noise ratio (SNR) is the ratio between the RMS amplitude of the fundamental frequency to the RMS amplitude of all other ADC spectral components, excluding harmonics. The output band is limited to frequencies above DC and below one-half the ADC sample (conversion) rate. This band includes both distortion and noise components. For this reason, the signal-to-noise and distortion ratio (SINAD) is a better measure of the ADC's performance.

The theoretical minimum ADC noise is caused by quantization error and is a direct result of the ADC's resolution:

$$SNR = (6.02N + 1.76)dB$$

where N is the number of bits of resolution. A perfect 12-bit ADC can therefore do no better than 74dB. Figure 16 shows the result of sampling a pure 10kHz sinusoid at a 100kHz rate with the MAX180/MAX181. An output FFT plot shows the relative output amplitude at discrete spectral frequencies (Figure 16).

By transposing the equation that converts resolution to SNR, we can determine the effective resolution (effective number of bits) the ADC provides from the measured SNR:  $N = (SNR - 1.76)/6.02$ . Figure 17 shows the effective number of bits as a function of the input frequency for the MAX180/MAX181.

## Total Harmonic Distortion

Total harmonic distortion (THD) is the ratio of the RMS sum of all the harmonics (in the frequency band above DC and below one-half the sample rate) to the RMS

amplitude of the fundamental frequency. This is expressed as:

$$THD = 20\text{Log} \left[ \frac{\sqrt{V_2^2 + V_3^2 + \dots + V_N^2}}{V_1} \right]$$

where  $V_1$  is the fundamental RMS amplitude, and  $V_2$  to  $V_N$  are the amplitudes of the 2nd through Nth harmonics.

## Spurious-Free Dynamic Range

Spurious-free dynamic range is the ratio of the fundamental RMS amplitude to the amplitude of the next largest spectral component (in the frequency band above DC and below one-half the sample rate). Usually this peak occurs at some harmonic of the input frequency. But if

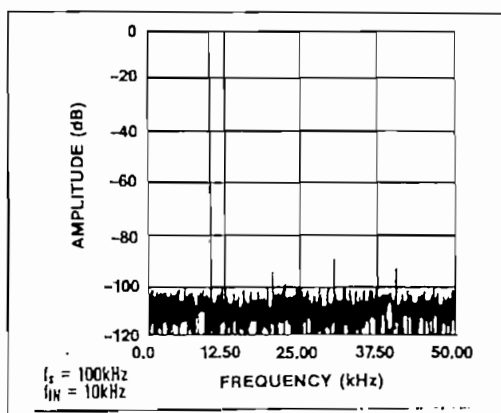


Figure 16. FFT Plot for the MAX180/MAX181

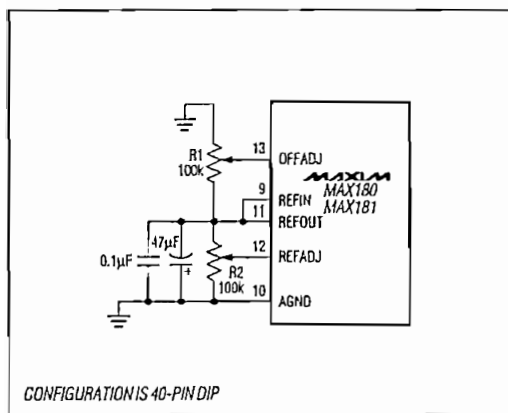


Figure 15. Offset and Gain Adjustment

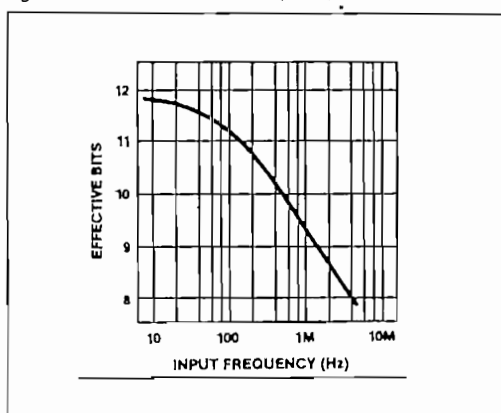


Figure 17. MAX180/MAX181 Effective Bits vs. Input Frequency

**ANEXO 5**  
**EL MICROCONTROLADOR AT89C52**

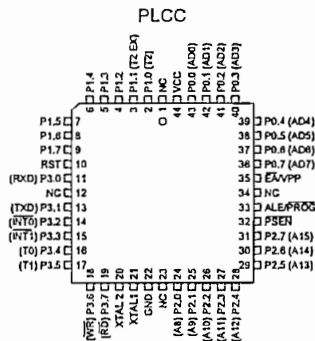
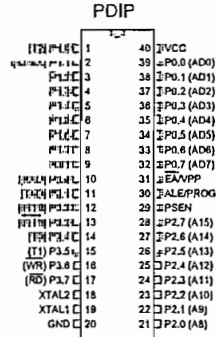
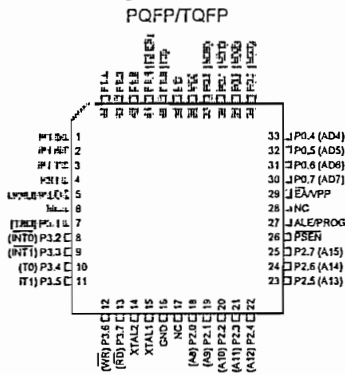
## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

## Description

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

## Pin Configurations



## 8-bit Microcontroller with 8K Bytes Flash

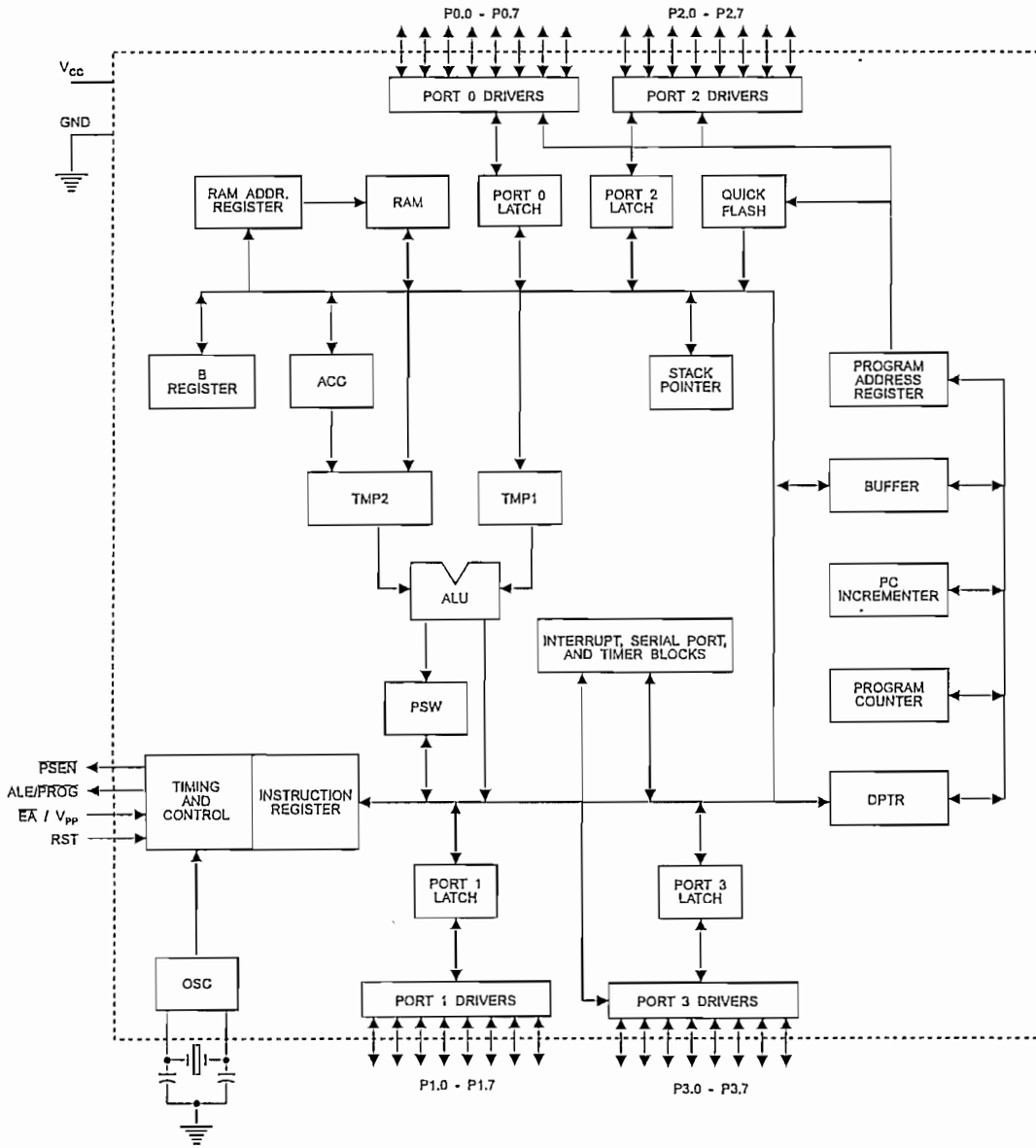
### AT89C52

Not Recommended  
for New Designs.  
Use AT89S52.





# Block Diagram



The AT89C52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full-duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

### Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

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

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

### ALE/ $\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ( $\overline{PROG}$ ) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external



timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

#### PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89C52 is executing code from external program memory,  $\overline{\text{PSEN}}$  is activated twice each machine cycle, except that two  $\overline{\text{PSEN}}$  activations are skipped during each access to external data memory.

#### EA/VPP

External Access Enable.  $\overline{\text{EA}}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed,  $\overline{\text{EA}}$  will be internally latched on reset.

$\overline{\text{EA}}$  should be strapped to  $V_{CC}$  for internal program executions.

This pin also receives the 12-volt programming enable voltage ( $V_{PP}$ ) during Flash programming when 12-volt programming is selected.

#### XTAL1

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

#### XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H



**Special Function Registers**

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

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

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

**Interrupt Registers** The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

**Table 2. T2CON – Timer/Counter 2 Control Register**

T2CON Address = 0C8H					Reset Value = 0000 0000B			
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function, C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

**Data Memory**

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction

specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```





Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

## Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit  $C/\overline{T2}$  in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

RCLK +TCLK	CP/ $\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external

input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

## Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

## Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 1. Timer in Capture Mode

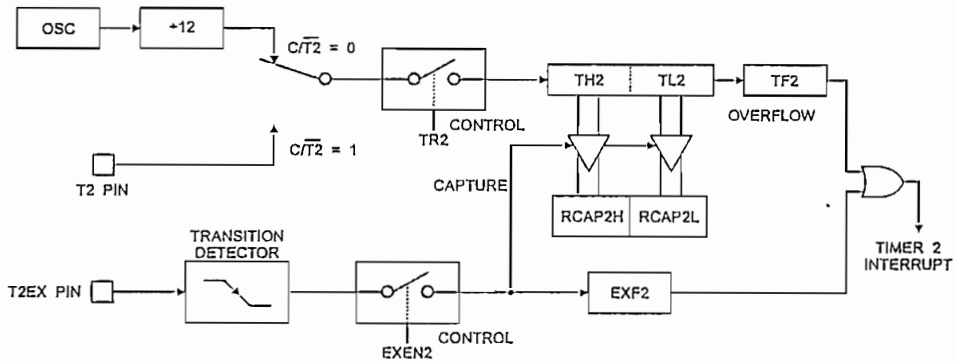


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture Mode RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls

the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

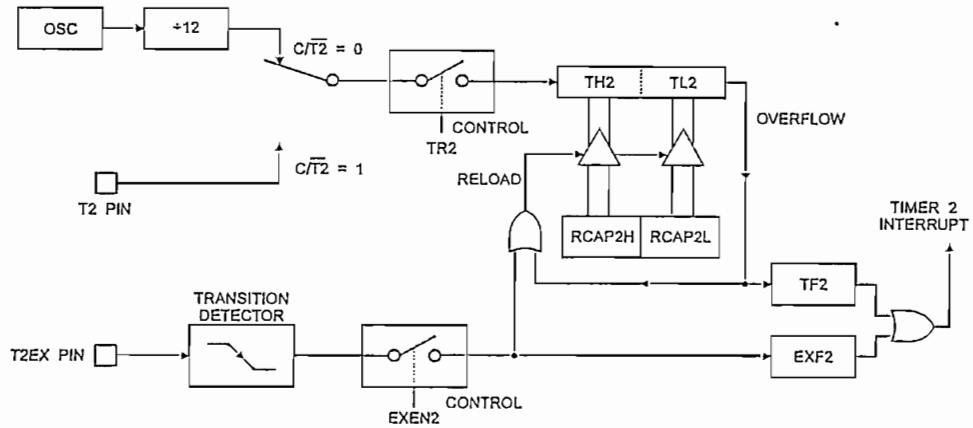


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN
	-	-	-	-	-	-		

Symbol	Function
-	Not implemented, reserved for future
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

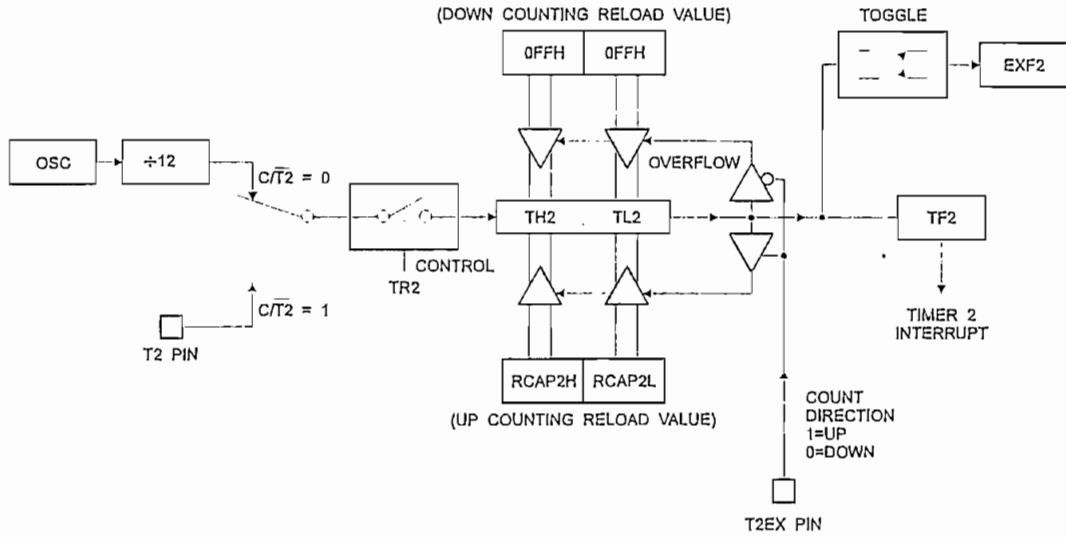
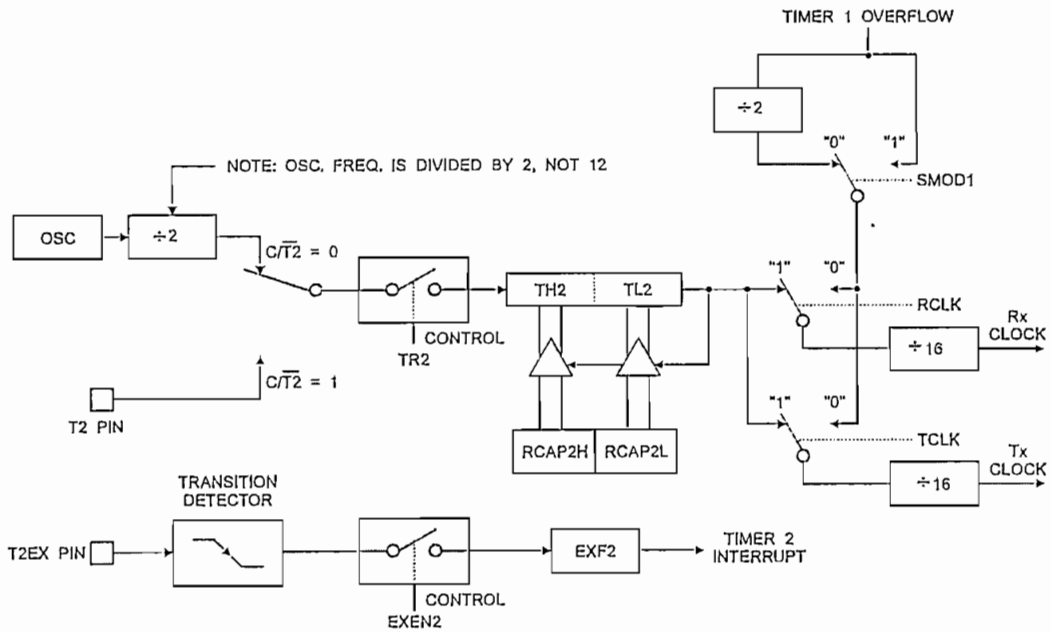


Figure 4. Timer 2 in Baud Rate Generator Mode



## Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ( $CP/\overline{T2} = 0$ ). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it

increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

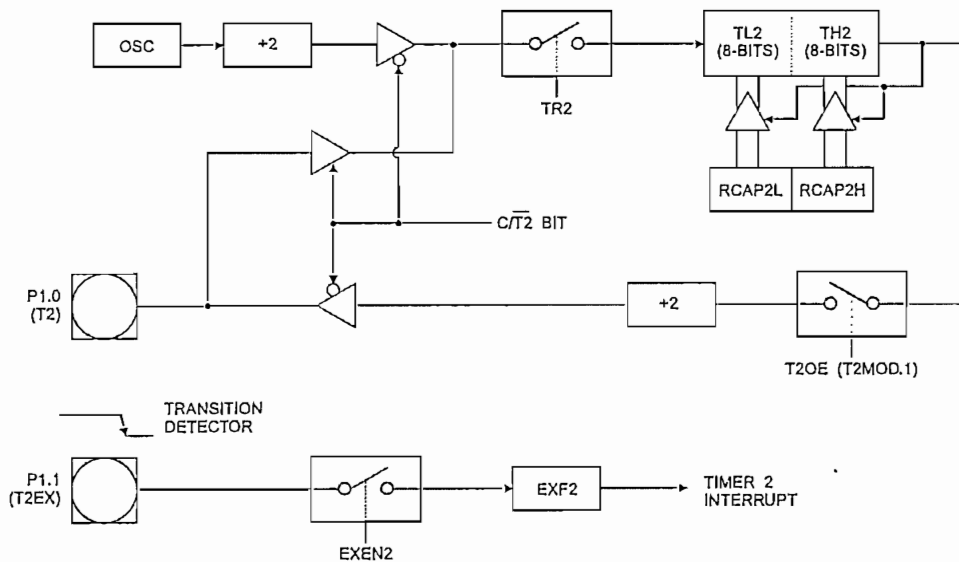
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running ( $TR2 = 1$ ) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-out Mode



**Programmable Clock Out**

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/√2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer. The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

**UART**

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

**Interrupts**

The AT89C52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However,

the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

**Table 5. Interrupt Enable (IE) Register**

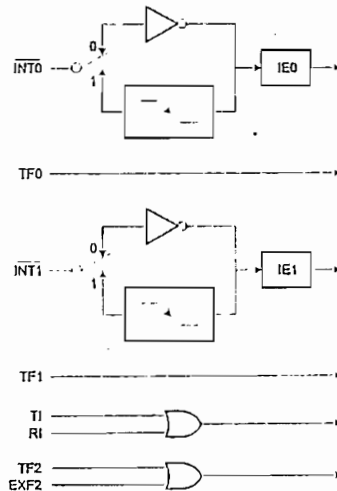
(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port Interrupt enable bit.
ET1	IE.3	Timer 1 Interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

**Figure 6. Interrupt Sources**



**ANEXO 6****GLOSARIO**



## GLOSARIO

<b>AIMS:</b>	Auto indexación para almacenamiento en masa
<b>AP:</b>	Punto de Acceso
<b>ARP:</b>	Protocolo de resolución de direcciones
<b>ATM:</b>	Modo de transferencia asincrónica
<b>BIP:</b>	Selección del modo unipolar/bipolar
<b>BPS:</b>	Bits por segundo
<b>BSS:</b>	Conjunto de servicios básicos
<b>BUSY:</b>	Salida de indicación de chip ocupado
<b>CAD:</b>	Convertidor analógico digital
<b>CIS:</b>	Estructura de la tarjeta de información
<b>CCITT:</b>	Comité Consultivo Internacional de Telegrafía y Telefonía
<b>CLKIN:</b>	Reloj de entrada
<b>CN:</b>	Nodo correspondiente
<b>CPU:</b>	Unidad Central de Procesos
<b>CS:</b>	Entrada para selección del chip
<b>CSMA/CA:</b>	Acceso Múltiple sensible a portadora evitación de colisiones
<b>CSMA/CD:</b>	Acceso Múltiple sensible a portadora con detección de colisiones
<b>DARPA:</b>	Agencia de Proyectos de Investigación Avanzada para la Defensa
<b>DBPSK:</b>	Modulación por desplazamiento de fase binario diferencial
<b>DCE:</b>	Equipo terminal del circuito de datos
<b>DGND:</b>	Tierra Digital
<b>DHCP:</b>	Protocolo de configuración dinámica de servidor
<b>DIFF:</b>	Selección del modo diferencial
<b>DNS:</b>	Sistema de nombre de dominio
<b>DSSS:</b>	Espectro ensanchado de secuencia directa
<b>DTE:</b>	Equipo terminal de datos
<b>DQPSK:</b>	Modulación por desplazamiento de fase por cuadratura diferencial
<b>E/S:</b>	Entrada/Salida
<b>EISA:</b>	Arquitectura Estándar Industrial Extendida
<b>EIA:</b>	Asociación de Industrias Electrónicas

<b>EPROM:</b>	Memoria solo lectura borrable programable
<b>ETSI:</b>	Instituto Europeo de estándares de telecomunicaciones
<b>ESS:</b>	Conjunto servicios extendidos
<b>FA:</b>	Agente externo
<b>FCC:</b>	Comisión Federal de Comunicaciones
<b>FDDI:</b>	Interfaz para distribución de datos en fibra
<b>FHSS:</b>	Espectro ensanchado de salto de frecuencia
<b>FSK:</b>	Modulación de frecuencia
<b>FTP:</b>	Protocolo de transferencia de archivos
<b>GND:</b>	Tierra Analógica
<b>HA:</b>	Agente local
<b>HBA:</b>	Adaptador del bus del host
<b>HBEN:</b>	Entrada de habilitación del Byte mayor
<b>HDD:</b>	Unidad de disco duro
<b>HFC:</b>	Redes Híbridas Fibra-Coaxial
<b>HRFWG:</b>	Grupo de trabajo HomeRF
<b>HTTP:</b>	Protocolo de transferencia de hipertexto
<b>ICMP:</b>	Protocolo de mensajes de control Internet
<b>IEEE:</b>	Instituto de Ingenieros Electrónicos y Eléctricos
<b>IETF:</b>	Grupo de Trabajo en Ingeniería de Internet
<b>IP:</b>	Protocolo Internet
<b>ISA:</b>	Arquitectura Estándar Industrial
<b>JEIDA:</b>	La Asociación de Desarrollo de Industrias Electrónicas Japonesas
<b>LAN:</b>	Red de Área Local
<b>LED:</b>	Diodo emisor de luz
<b>LLC:</b>	Enlace de control lógico
<b>LMDS:</b>	Sistema de distribución multipunto local
<b>LSB:</b>	Bit menos significativos
<b>MAC:</b>	Control de acceso al medio
<b>MN:</b>	Nodo móvil
<b>MMDS:</b>	Sistema de Distribución Multipunto de Multicanales
<b>MSB:</b>	Bit más significativo
<b>OFFADJ:</b>	Ajuste del Offset
<b>PAN:</b>	Red de área personal

<b>PC:</b>	Computador personal
<b>PCB:</b>	Tarjeta de circuito impreso
<b>PCI:</b>	Componente de interconexión Física
<b>PCMCIA:</b>	Asociación Internacional de Tarjetas de Memoria de Computadoras Personales
<b>PING:</b>	Rastreador de Paquetes Internet
<b>PON:</b>	Redes Ópticas Pasivas
<b>POP3:</b>	Protocolo de la oficina de correos
<b>PPP:</b>	protocolo punto a punto
<b>PSTN:</b>	La red telefónica pública conmutada
<b>RAS:</b>	Sistema de acceso remoto
<b>RAM:</b>	Memoria de acceso randomico
<b>RARP:</b>	Protocolo reverso de resolución de direcciones
<b>RD:</b>	Lectura
<b>RDSI:</b>	Red Digital de Servicios Integrados
<b>REFADJ:</b>	Ajuste de Referencia
<b>REFIN:</b>	Entrada de Referencia
<b>REFOUT:</b>	Salida de Referencia
<b>ROM:</b>	Memoria de solo lectura
<b>RS 232:</b>	Puerto serial
<b>RTD:</b>	Resistencias Detectoras de Temperatura
<b>SAD:</b>	Sistema de Adquisición de Datos
<b>SAP's:</b>	Puntos de Acceso al Servicio
<b>SIG:</b>	Grupo de interes especial
<b>SLIP:</b>	Protocolo Internet de línea serie
<b>SMTP:</b>	Protocolo simple de transferencia de correo
<b>SNMP:</b>	Protocolo simple de administración de red
<b>SSID:</b>	Identificador del conjunto de servicios
<b>SWAP-CA:</b>	Protocolo de acceso compartido inalámbrico – Acceso sin cables
<b>T/H:</b>	Seguidor/Retenedor
<b>TCP:</b>	Protocolo de control de transmisión
<b>TDMA:</b>	Acceso multiplexado por división de tiempo
<b>TELNET:</b>	Nombre de un protocolo que sirve para acceder remotamente a otra máquina

<b>TFTP:</b>	Protocolo trivial de transferencia de archivos
<b>TTL:</b>	Tiempo de vida
<b>UDP:</b>	Protocolo de datagrama de usuario
<b>UIT:</b>	Unión Internacional para la Reglamentación de las Telecomunicaciones
<b>VPN:</b>	Red privada virtual
<b>WECA:</b>	Alianza de compatibilidad para Ethernet inalámbrico
<b>WDM:</b>	Multiplexación por división de longitud de onda
<b>WEP:</b>	Protocolo equivalente inalámbrico
<b>Wi-Fi:</b>	Fidelidad inalámbrica
<b>WINS:</b>	Servicio de nombre para Internet de Windows
<b>WLAN:</b>	Red de Área Local Inalámbrica
<b>WLL:</b>	Lazo local inalámbrico
<b>WR:</b>	Escritura