



UNIVERSIDAD CATÓLICA DE TEMUCO
FACULTAD DE CIENCIAS.

Conectividad inalámbrica entre Microcontroladores y equipos de Telefonía Celular y sus aplicaciones practicas en Sistemas Embebidos y Bases de Datos.

Por

GUIDO ROBERTO BASCUÑAN LEONELLI

Trabajo de titulo presentado a la
Facultad de Ciencias de la Universidad Católica de Temuco
Para Optar al Titulo de Ingeniero en Ejecución en Informática

-Temuco, año 2005-

UNIVERSIDAD CATÓLICA DE TEMUCO
FACULTAD DE CIENCIAS

COMISIÓN EXAMEN DE TÍTULO

Este Examen de Título ha sido realizado en la Escuela de Ingeniería en Informática.

Presidente Comisión:
Oriol Herrera G.
Director Escuela de Ingeniería en Informática.

Profesor Guía:
Alberto Caro S.
Secretario Académico Escuela de
Ingeniería en Informática.

Profesor Informante:
Alejandro Mellado G.
Docente Universidad Católica de Temuco

Secretario Académico:
De la Escuela. Alberto Caro
Secretario Académico Escuela de
Ingeniería en Informática.

Temuco, 19 de Agosto de 2005



INFORME TRABAJO DE TÍTULO

TÍTULO : “Conectividad inalámbrica entre Microcontroladores y equipos de Telefonía Celular y sus aplicaciones practicas en Sistemas Embebidos y Bases de Datos”

ALUMNO : Guido Roberto Bascuñan Leonelli

En mi condición de profesor guía de este trabajo puedo efectuar las siguientes observaciones:

- El trabajo esta ordenado y bien estructurado y se han usado herramientas metodológicas adecuadas.
- Se aprecia un buen nivel de dedicación y un profundo dominio sobre el tema.
- La complejidad del trabajo es de un nivel avanzado, no obstante, el alumno logra motivar y enfocar su trabajo de manera amena y clara.

De acuerdo a estas consideraciones califico este trabajo con nota 7.0 (Siete coma cero)

Luis Alberto Caro Saldivia
Profesor Guía

Temuco, 19 de Agosto de 2005



INFORME TRABAJO DE TÍTULO

TÍTULO : “Conectividad inalámbrica entre Microcontroladores y equipos de Telefonía Celular y sus aplicaciones practicas en Sistemas Embebidos y Bases de Datos”

ALUMNO : Guido Roberto Bascuñan Leonelli

En mi condición de profesor informante de este trabajo puedo efectuar las siguientes observaciones:

- El trabajo esta bien estructurado y ordenado. Denota dedicación de esmero.
- Este trabajo revela un buen nivel técnico en la integración de distintos elementos y sus relaciones, para llegar a una fusión de distintas tecnologías.
- Debe mejorar un poco sus conclusiones.

De acuerdo a estas consideraciones califico este trabajo con nota 7.0 (Siete coma cero)

Alejandro Mauricio Mellado Gatica
Profesor Informante

Temuco, 19 de Agosto de 2005

INDICE DE CONTENIDOS

1. Microcontroladores.....	5
1.1. Introducción a los Microcontroladores.....	5
1.2. Controlador y Microcontrolador.....	5
1.3. Historia de los Microcontroladores.....	7
1.4. Recursos Comunes de todos los Microcontroladores.....	9
1.5. Arquitectura Básica.....	9
1.5.1. Arquitectura de Von Neumann.....	9
1.5.2. Arquitectura de Harvard.....	11
1.6. El Procesador.....	12
1.6.1. CISC.....	13
1.6.2. RISC.....	13
1.6.3. SISC.....	13
1.7. Memoria.....	14
1.7.1. Rom Con Mascara.....	15
1.7.2. OTP.....	15
1.7.3. EPROM.....	15
1.7.4. EEPROM.....	16
1.7.5. FLASH.....	17
1.8. Puertas de Entrada y Salida.....	17
1.9. Reloj Principal.....	18
1.10. Recursos Especiales.....	18
1.10.1. Temporizadores o “Timers”.....	20
1.10.2. Perro Guardián o “Watchdog”.....	20
1.10.3. Protección ante fallo de alimentación o “Brownout”.....	21
1.10.4. Estado de reposo ó de bajo consumo.....	21
1.10.5. Conversor A/D (CAD).....	21
1.10.6. Conversor D/A (CDA).....	22
1.10.7. Comparador analógico.....	22
1.10.8. Modulador de anchura de impulsos o PWM.....	22
1.10.9. Puertas de E/S digitales.....	22
1.10.10. Puertas de comunicación.....	23
1.10.10.1. UART.....	23
1.10.10.2. USART.....	23
1.10.10.3. Puerta Paralela.....	23
1.10.10.4. USB.....	23
1.10.10.5. Bus I ² C.....	23
1.10.10.6. CAN.....	23
1.11. Ventajas y Defectos de los Microcontroladores.....	24
2. Tipos de Microcontroladores.....	26
2.1. Intel 8051.....	26
2.2. Motorota 68HCXX.....	28
2.3. Microcontroladores PIC.....	30
2.3.1. Características relevantes.....	31
2.3.1.1. Arquitectura.....	31
2.3.1.2. Segmentación.....	31
2.3.1.3. Formato de las instrucciones.....	32
2.3.1.4. Juego de instrucciones.....	32
2.3.1.5. Todas las instrucciones son ortogonales.....	32
2.3.1.6. Arquitectura basada en un “banco de registros”.....	33

2.3.1.7. Diversidad de modelos de Microcontroladores.....	33
2.3.1.8. Herramientas de soporte potentes y económicas.....	33
2.3.2. Las gamas de PIC.....	34
2.3.2.1. Gamma enana : PIC12C(F)XXX de 8 patitas.....	34
2.3.2.2. Gamma baja : PIC16C5X.....	35
2.3.2.2.1. Sistema POR.....	35
2.3.2.2.2. Perro Guardián.....	36
2.3.2.2.3. Código de protección.....	36
2.3.2.2.4. Líneas de E/S de alta corriente.....	36
2.3.2.2.5. Modo de Reposo (sleep).....	36
2.3.2.3. Gamma Media : PIC16CXXX con instrucciones de 14 Bits.....	37
2.3.2.4. Gamma Alta: PIC17CXXX con instrucciones de 16 Bits.....	38
3. Presentación del Microcontrolador PIC16X84.....	39
3.1. Descripción General.....	39
3.2. Aspecto externo.....	41
3.3. La Frecuencia de Funcionamiento : Clock.....	42
3.4. Tipos de Osciladores.....	43
3.5. Reinicialización o Reset.....	44
3.6. Arquitectura de los Microcontroladores PIC16X84.....	44
3.7. Puertas de E/S.....	46
3.7.1. La Puerta A.....	46
3.7.2. La Puerta B.....	47
3.8. La Memoria EEPROM de Datos.....	48
3.9. Interrupciones.....	49
3.9.1. Causas de Interrupción.....	50
4. Comunicación Inalámbrica.....	52
4.1. Introducción.....	52
4.2. Comunicación de Radiofrecuencia.....	53
4.3. Comunicación de Microondas.....	54
4.4. Comunicación Vía Satélite.....	55
4.5. Comunicación infrarroja.....	57
4.5.1. Modo Punto a Punto.....	58
4.5.2. Modo Cuasi-Difuso.....	59
4.5.3. Modo Difuso.....	60
4.6. Modulación.....	61
4.6.1. Modulación para facilidad de radiación.....	61
4.6.2. Modulación para reducir el ruido y la interferencia.....	62
4.6.3. Modulación por asignación de frecuencia.....	62
4.6.4. Modulación para multicanalización.....	63
4.6.5. Modulación para superar las limitaciones del equipo.....	63
4.6.6. Modulaciones amplitud y bandas laterales.....	64
4.6.7. Modulación de frecuencia.....	65
5. Protocolos de Comunicación.....	67
5.1. Introducción.....	67
5.2. Protocolo Estándar.....	69
5.3. Funciones de un Protocolo.....	70
5.3.1. Segmentación y ensamblado.....	70
5.3.2. Encapsulado.....	70
5.3.3. Control de conexión.....	70
5.3.4. Entrega ordenada.....	70

5.3.5.	Control de flujo.....	71
5.3.6.	Control de errores.....	71
5.3.7.	Direccionamiento.....	72
5.3.8.	Multiplexacion.....	72
5.3.9.	Servicios de transmisión.....	72
5.3.9.1.	Prioridad.....	73
5.3.9.2.	Grado de servicio.....	73
5.3.9.3.	Seguridad.....	73
5.4.	Tipos de Protocolos.....	73
5.4.1.	Protocolos orientados a la coneccion.....	73
5.4.2.	Protocolos orientados a la no coneccion.....	76
5.4.3.	Protocolos orientados a Byte.....	77
5.4.4.	Protocolos orientados a Bits.....	77
6.	Bluetooth.....	78
6.1.	Introducción.....	78
6.2.	Características.....	79
6.3.	Banda de frecuencia libre.....	81
6.4.	Salto de frecuencia.....	81
6.5.	Seguridad.....	82
6.6.	beneficios de Bluetooth.....	83
6.7.	Otros escenarios usando Bluetooth.....	85
6.8.	Interferencia en las redes Bluetooth.....	86
6.9.	Consideraciones en las conexiones.....	87
6.9.1.	Trabajo con anchos de bandas limitados.....	87
6.9.2.	Consumo de energía y alcance.....	87
6.9.2.1.	Manejando un alcance aceptable.....	88
6.9.2.2.	Máxima salida de poder versus el alcance.....	88
6.10.	Dispositivos y adaptadores basados en Bluetooth.....	90
7.	Comunicación entre dispositivos.....	93
7.1.	Introducción.....	93
7.2.	Esquema principal.....	93
7.3.	Transmisores.....	95
7.4.	Desarrollo.....	96
7.5.	Transmisión y Trama de datos.....	97
7.6.	Equipo celular y Bluetooth.....	98
7.7.	Base de Datos.....	100
8.	Códigos Fuentes.....	106
8.1.	Códigos en el Microcontrolador.....	106
8.1.1.	Explicación de códigos.....	106
8.1.2.	Código completo.....	107
8.2.	Códigos de Equipo Servidor.....	109
8.2.1.	Explicación de códigos.....	109
8.2.2.	Código completo.....	113
9.	Conclusión.....	119
10.	Bibliografía.....	121

INDICE DE IMÁGENES

1. Microcontroladores	
1.5.1. Esquema de Arquitectura de Von Neumann.....	10
1.5.2. Esquema de Arquitectura de Harvard.....	12
2. Tipos de Microcontroladores	
2.1. Intel 8051.....	26
2.2. Motorola 68HC11.....	28
3. Presentación del Microcontrolador PIC16X84	
3.1. Microcontrolador PIC16F84A.....	40
3.2. Microcontrolador PIC16X8X.....	41
4. Comunicación Inalámbrica	
4.2. Transmisor y receptor de radio.....	53
4.3. Antena microondas.....	55
4.4. Ejemplo de satélite.....	57
4.5.1. Modo Punto a Punto.....	58
4.5.2. Modo Cuasi-Difuso.....	59
4.5.3. Modo Difuso.....	60
4.6.7. Ejemplo de señal modulada.....	65
5. Protocolos de comunicación	
5.4.1. Esquema de protocolo orientado a la conexión.....	75
5.4.2. Esquema de protocolo orientado a la no conexión.....	76
6. Bluetooth	
6.3. Campo de uso de algunas frecuencias.....	81
6.6.1. Mapa de interconexiones de dispositivos.....	84
6.9.2.2.1. Diseño de PCB modulo clase 1.....	89
6.9.2.2.2. Diseño de PCB modulo clase 2.....	89
6.9.2.2.3. Modulo clase 2.....	90
6.10.1. Adaptador Bluetooth para dispositivos con USB.....	91
6.10.2. Adaptador Bluetooth para PC.....	91
6.10.3. PDA con auricular Bluetooth.....	92
7. Comunicación entre dispositivos	
7.2. Diagrama de flujo de información.....	94
7.3.1. Transmisor inalámbrico de 433Mhz.....	95
7.3.2. Receptor inalámbrico de 433Mhz.....	95
7.4. Aplicación de control en el Equipo central.....	96
7.5. Trama de datos.....	97
7.6. Adaptador USB a Bluetooth.....	99
7.7.1. Motor de bases de datos.....	101
7.7.2. Dialogo de ingreso de consulta.....	102
7.7.3. Respuesta generada por la consulta.....	103
7.7.4. Menú de control de robot.....	104

1. Microcontroladores

1.1. Introducción a los Microcontroladores

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricaremos y usamos los humanos.

1.2. Controlador y Microcontrolador

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la

actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

1. Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
2. Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.

3. Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.

4. Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna.

1.3. Historia de los microcontroladores

1971: Intel fabrica el primer microprocesador (el 4004) de tecnología PMOS. Este era un microprocesador de 4 bits y fue fabricado por Intel a petición de Datapoint Corporation con el objeto de sustituir la CPU de terminales inteligentes que eran fabricadas en esa fecha por Datapoint mediante circuitería discreta. El dispositivo fabricado por Intel resultó 10 veces más lento de lo requerido y Datapoint no lo compró, de esta manera Intel comenzó a comercializarlo. El 4004 era un microprocesador de 4 bits, contenía 2,300 transistores y corría a 108 Khz podía direccionar sólo 4096 (4k) localidades de memoria de 4 bits, reconocía 45 instrucciones y podía ejecutar una instrucción en 20 µseg en promedio. Este procesador se utilizó en las primeras calculadoras de escritorio.

1972. Las aplicaciones del 4004 estaban muy limitadas por su reducida capacidad y rápidamente Intel desarrolló una versión más poderosa (el 8008), el cual podía manipular Bytes completos, por lo cual fue un microprocesador de 8 bits. La memoria que este podía manejar se incrementó a 16 KBytes, sin embargo, la velocidad de operación continuó igual.

1973 Intel lanza al mercado el 8080 el primer microprocesador de tecnología NMOS, lo cual permite superar la velocidad de su predecesor (el 8008) por un factor de diez, es decir, el 8080 puede realizar 500 000 operaciones por segundo, además se incrementó la capacidad de direccionamiento de memoria a 64 KBytes. A partir del 8080 de Intel se produjo una revolución en el diseño de microcomputadoras y varias compañías fabricantes de circuitos integrados comenzaron a producir microprocesadores. Algunos ejemplos de los primeros microprocesadores son: el IMP-4 y el SC/MP de National Semiconductors, el PPS-4 y PPS-8 de Rockwell International, el MC6800 de Motorola, el F-8 de Fairchild.

1975 Zilog lanza al mercado el Z80, uno de los microprocesadores de 8 bits más poderosos. En ese mismo año, Motorola abate dramáticamente los costos con sus microprocesadores 6501 y 6502 (este último adoptado por APPLE para su primera microcomputadora personal). Estos microprocesadores se comercializan en \$20 y \$25 Dólares respectivamente. Esto provoca un auge en el mercado de microcomputadoras de uso doméstico y un caos en la proliferación de lenguajes, sistemas operativos y programas (ningún producto era compatible con el de otro fabricante).

1976 Surgen las primeras microcomputadoras de un sólo chip, que más tarde se denominarán microcontroladores. Dos de los primeros microcontroladores, son el 8048 de Intel y el 6805R2 de Motorola.

198x En la década de los 80's comienza la ruptura entre la evolución tecnológica de los microprocesadores y la de los microcontroladores, Ya que los primeros han ido incorporando cada vez más y mejores capacidades para las aplicaciones en donde se requiere el manejo de grandes volúmenes de información y por otro lado, los segundos han incorporado más capacidades que les permiten la interacción con el mundo físico en tiempo real, además de mejores desempeños en ambientes de tipo industrial.

1.4. Recursos comunes de todos los microcontroladores

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales: Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

En este apartado se hace un recorrido de todos los recursos que se hallan en todos los microcontroladores describiendo las diversas alternativas y opciones que pueden encontrarse según el modelo seleccionado.

1.5. Arquitectura básica

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en el momento presente se impone la arquitectura de Harvard.

1.5.1. Arquitectura Von Neumann

La arquitectura tradicional de computadoras y microprocesadores se basa en el esquema propuesto por John Von Neumann, en el cual la unidad central de proceso, o CPU, esta conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de la memoria. Es decir que un microprocesador de 8 bits, que tiene además un bus de 8 bits que lo conecta con la memoria, deberá manejar datos e instrucciones de una o más unidades de 8

bits de longitud. Cuando deba acceder a una instrucción o dato de más de un Byte de longitud, deberá realizar más de un acceso a la memoria. Por otro lado este bus único limita la velocidad de operación del microprocesador, ya que no se puede buscar de memoria una nueva instrucción, antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior. Es decir que las dos principales limitaciones de esta arquitectura tradicional son:

- a) Que la longitud de las instrucciones esta limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones complejas.
- b) Que la velocidad de operación (o ancho de banda de operación) esta limitada por el efecto de cuello de botella que significa un bus único para datos e instrucciones que impide superponer ambos tiempos de acceso.

La arquitectura Von Neumann permite el diseño de programas con código auto modificables, práctica bastante usada en las antiguas computadoras que solo tenían acumulador y pocos modos de direccionamiento, pero innecesaria, en las computadoras modernas.

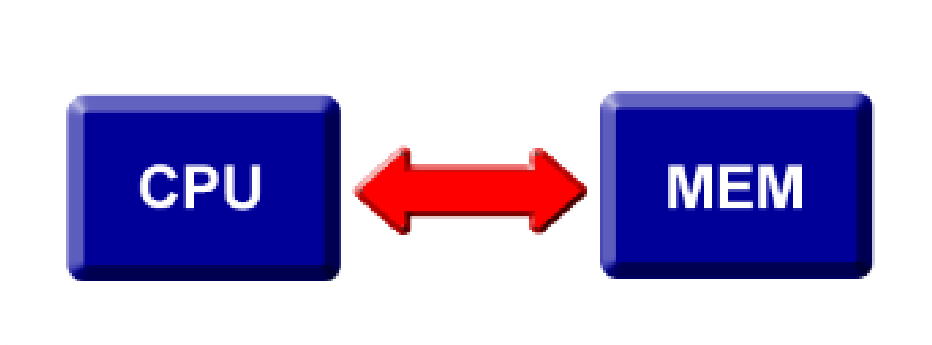


Figura 1.5.1. Esquema de Arquitectura de Von Neumann.

1.5.2. La arquitectura Harvard

La arquitectura conocida como Harvard, consiste simplemente en un esquema en el que el CPU esta conectado a dos memorias por intermedio de dos buses separados. Una de las memorias contiene solamente las instrucciones del programa, y es llamada Memoria de Programa. La otra memoria solo almacena los datos y es llamada Memoria de Datos. Ambos buses son totalmente independientes y pueden ser de distintos anchos. Para un procesador de conjunto de Instrucciones Reducido, o RISC (computador con un conjunto reducido de instrucciones), el conjunto de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, como los buses son independientes, el CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar. Se puede observar claramente que las principales ventajas de esta arquitectura son:

- a) Que el tamaño de las instrucciones no esta relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa,
- b) Que el tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad de operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que pueda ser necesario incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

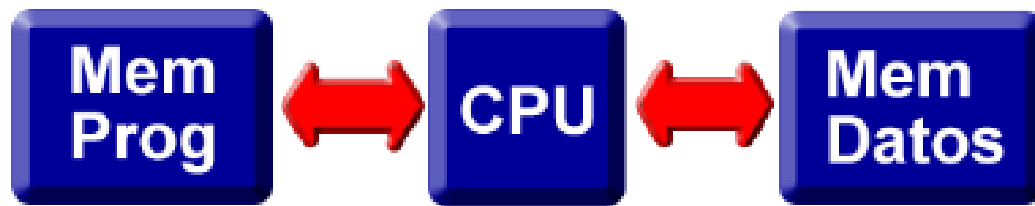


Figura 1.5.2. Esquema de Arquitectura de Harvard.

Los microcontroladores PIC 16C5X, 16CXX y 17CXX poseen arquitectura Harvard, con una memoria de datos de 8 bits, y una memoria de programa que, según el modelo, puede ser de 12 bits para los 16C5X, 14 bits para los 16CXX y 16 bits para los 17CXX.

1.6.El procesador o UCP (CPU)

Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

1.6.1. CISC

Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (computador con un conjunto complejo de instrucciones). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

1.6.2. RISC

Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de conjunto de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

1.6.3. SISC

En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de conjunto de Instrucciones Específico).

1.7. Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

- a) No existen sistemas de almacenamiento masivo como disco duro o disquetes.
- b) Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 Bytes y 8 k Bytes y de RAM comprendidas entre 20 y 512 Bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1.7.1. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

1.7.2. OTP

El microcontrolador contiene una memoria no volátil de sólo lectura “programable una sola vez” por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

1.7.3. EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son

más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

1.7.4. EEPROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan “grabadores en circuito” que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno.

1.7.5. FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados “en circuito”, es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

1.8. Puertas de Entrada y Salida

La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

1.9. Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

1.10. Recursos Especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o “Timers”.
- Perro guardián o “Watchdog”.
- Protección ante fallo de alimentación o “Brownout”.
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

1.10.1. Temporizadores o “Timers”

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

1.10.2. Perro guardián o “Watchdog”

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, “ladrará y ladrará” hasta provocar el reset.

1.10.3. Protección ante fallo de alimentación o “Brownout”

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor

1.10.4. Estado de reposo ó de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando sumido en un profundo “sueño” el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

1.10.5. Conversor A/D (CAD)

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

1.10.6. Conversor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

1.10.7. Comparador analógico

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

1.10.8. Modulador de anchura de impulsos o PWM

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

1.10.9. Puertas de E/S digitales

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertas.

Las líneas digitales de las Puertas pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el BIT correspondiente de un registro destinado a su configuración.

1.10.10. Puertas de comunicación

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- 1.10.10.1.** UART, adaptador de comunicación serie asíncrona.
- 1.10.10.2.** USART, adaptador de comunicación serie síncrona y asíncrona
- 1.10.10.3.** Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- 1.10.10.4.** USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- 1.10.10.5.** Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.
- 1.10.10.6.** CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

1.11. Las ventajas y defectos de los Microcontroladores

En primer lugar, un microcontrolador integra en un único encapsulado lo que antes necesitaba una o varias decenas de elementos distintos. Como resultado de estos, se ha obtenido una evidente disminución en el volumen del hardware y del circuito impreso. Esta integración también ha tenido como consecuencia inmediata la simplificación del diseño del circuito impreso, ya que no es necesario llevar los buses de direcciones y de datos de un componente a otro.

En segundo lugar, de todo lo anterior se deriva un aumento de la fiabilidad del sistema ya que, al disminuir el número de componentes, el número potencial de fuentes de error también disminuye. Además, la cantidad de conexiones entre componentes/zócalos o componentes/circuito impreso también disminuye, lo que aumenta la fiabilidad del sistema. Así mismo, la disminución de componentes reduce el consumo global de todo el sistema, lo que según en que aplicaciones y tipos de alimentación se utilice el microcontrolador puede resultar ventajoso.

Los mayores inconvenientes de los microcontroladores son bastante pocos y, principalmente, se encuentran en el nivel de la programación, pero en dos planos diferentes. El primer inconveniente es el sistema de almacenamiento de los programas que lo hacen funcionar, como ya hemos visto, las opciones de almacenamiento pasan por utilizar una memoria ROM en alguna de sus variantes (ROM por máscara, UV PROM, OTPROM, EEPROM, etc.), esto implica que la modificación de los programas realizados va a suponer un esfuerzo de borrado de la memoria completa (o de bloques en el mejor de los casos) o la sustitución del chip de memoria por uno nuevo, lo cual conlleva gastos adicionales en material o en esfuerzo.

El otro inconveniente es el de que en los microcontroladores, como cualquier sistema programado, hay que disponer de una herramienta o medio de desarrollo, es necesario escribir los programas, probarlos y comprobarlos sobre el

hardware que rodea al microcontrolador, antes de instalarlos y hacer funcionar el sistema. Este sistema de desarrollo representa, por tanto, una inversión que hay que tener en cuenta en el coste del producto final. Si se prevé la realización de aparatos diversos que utilicen microcontroladores de la misma familia, es bastante fácil de amortizar; en caso contrario puede ser más difícil.

2. Tipos de microcontroladores

A continuación una lista de los microcontroladores más comunes y usados durante estos últimos años.

2.1. Intel 8051

El 8051 miembro original de la familia de los microcontroladores MCS(R) 51, y es el corazón de todos los dispositivos basados en MCS 51. Las principales características del 8051 son:

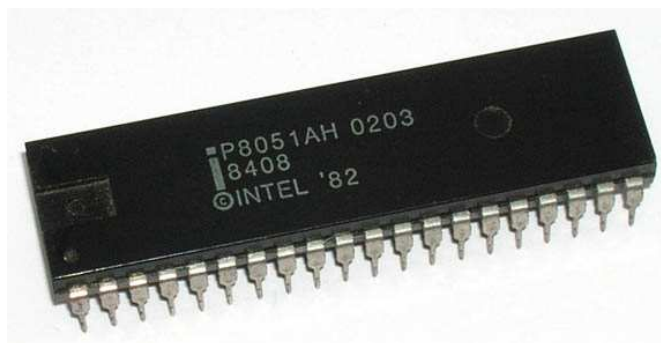


Figura 2.1. Intel 8051.

- 8-BIT CPU Optimizada para el control de aplicaciones
- Capacidades de proceso extendido de números booleanos (Lógica de un solo BIT)
- 64K de memoria Programable
- 64K memoria para datos
- 4K Bytes de memoria de programa en el chip
- 128 Bytes para datos internos

-
- 32 direcciones individuales y bidireccionales
 - 2 contadores de 16-BIT
 - Full duplex UART
 - Oscilador interno

El 8051 es un microprocesador de 8 BIT diseñado al principio del año 1980 por Intel, la cual ha ganado la gran popularidad desde su introducción. Su diseño estándar incluye varios periférico estándar chip, incluyendo temporizadores, contadores, y una UART, además de 4kbytes de memoria de programa en el chip y 128 Bytes de memoria de datos, poniendo en práctica la implementación de un solo chip posibles. Sus cientos de derivados, fabricados por varias empresas diferentes (como Philips) incluyen más periférico integrados en el chip, como convertidores análogos digitales, moduladores de anchura de pulso, I2C interfaces de bus, etc.

La arquitectura del 8051 incluye memoria de datos de 128 Bytes que son accesibles directamente por sus instrucciones. Un segmento de 32 Bytes de este bloque de memoria de 128 Bytes es direccionable por un subconjunto de las instrucciones del 8051, llamada las instrucciones de BIT. La memoria externa es de hasta 64 KBytes la cual es accesible por la instrucción especial "movx". Hasta 4 KBytes para instrucciones de programa que pueden ser almacenados en la memoria interna del 8051, o también puede ser configurado para usar hasta 64 KBytes de memoria de programa externa, la mayoría de las instrucciones del 8051 es ejecutada dentro de 12 ciclos de reloj.

2.2. Motorola 68HCXX

A la familia 68HCXX pertenecen los siguientes microcontroladores Motorola 68HC05, 68HC08 y 68HC11 de 8 bits, 68HC12 y 68HC16 de 16, y 68300 y M-CORE de 32. Así, la familia 6805 (de gama baja y media) es la más vendida en la historia de los microcontroladores; el 68HC08 se introdujo para usuarios de la 05 que necesitan más prestaciones. Entre las más populares está la familia 68HC11 (gama alta), que por su relación prestaciones/precio y gran facilidad de uso resulta ideal para el aficionado a la electrónica. Para usuarios del HC11 que requieren más prestaciones se han introducido las familias de 16 bits 68HC12 y el 68HC16 (con operaciones DSP).



Figura 2.2. Motorola 68HC11.

Todos los miembros de las familias 68XX parten de la arquitectura del clásico microprocesador M6800: en algunos casos el núcleo del microcontrolador (o CPU) es un 6800 reforzado (68HC11) y en otros simplificado (6805). Sus juegos

de instrucciones se basan en el del 6800, y los modos de direccionamiento son prácticamente los mismos, con lo que la programación de todas las familias resulta muy similar.

El 68HC11, uno de los microcontroladores de 8 bits más populares en el mundo, que integra en una única pastilla un elevado número de periféricos y memoria. Una de las grandes ventajas del 68HC11 es que con él pueden llevarse a cabo aplicaciones importantes sin necesidad de herramientas especiales, ya que Motorola ha introducido en su diseño muchos detalles que facilitan el desarrollo de sistemas.

2.3. Microcontroladores PIC

Los PIC tienen algo que atrae a los diseñadores, puede ser la velocidad, el precio, la facilidad de uso, la información o las herramientas de apoyo. Quizás un poco de todo eso es lo que produce esa imagen de sencillez y utilidad.

Queremos constatar que para las aplicaciones más habituales (casi un 90%) la elección de una versión adecuada de PIC es la mejor solución; sin embargo, dado su carácter general, otras familias de microcontroladores son más eficaces en aplicaciones específicas, especialmente si en ellas predomina una característica concreta, que puede estar muy desarrollada en otra familia.

Los detalles más importantes que los profesionales de la microelectrónica y microinformática les importan son los siguientes:

1. Sencillez de manejo: Tienen un juego de instrucciones reducido; 35 en la gama media.
2. Buena información, fácil de conseguir y económica.
3. Precio: Su coste es comparativamente inferior al de sus competidores.
4. Poseen una elevada velocidad de funcionamiento. Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.
5. Herramientas de desarrollo fáciles y baratas. Muchas herramientas software se pueden recoger libremente a través de Internet desde Microchip.
6. Existe una gran variedad de herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
7. Diseño rápido.
8. La gran variedad de modelos de PIC permite elegir el que mejor responde a los requerimientos de la aplicación.

Una de las razones del éxito de los PIC se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

2.3.1. Características relevantes

Descripción de las características más representativas de los PIC:

2.3.1.1. Arquitectura

La arquitectura del procesador sigue el modelo Harvard. En esta arquitectura, la CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos.

La arquitectura Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema como se irán describiendo.

2.3.1.2. Segmentación

Se aplica la técnica de segmentación (“pipe-line”) en la ejecución de las instrucciones.

La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

2.3.1.3. Formato de las instrucciones

El formato de todas las instrucciones es de la misma longitud.

Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

2.3.1.4. Juego de instrucciones

Procesador RISC (Computador de conjunto de Instrucciones Reducido).

Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

2.3.1.5. Todas las instrucciones son ortogonales

Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

2.3.1.6. Arquitectura basada en un “banco de registros”

Esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

2.3.1.7. Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes

La gran variedad de modelos de microcontroladores PIC permite que el usuario pueda seleccionar el más conveniente para su proyecto.

2.3.1.8. Herramientas de soporte potentes y económicas

La empresa Microchip y otras que utilizan los PIC ponen a disposición de los usuarios numerosas herramientas para desarrollar hardware y software. Son muy abundantes los programadores, los simuladores software, los emuladores en tiempo real, Ensambladores, Compiladores C, Intérpretes y Compiladores Basic, etc.

La arquitectura Harvard y la técnica de segmentación son los principales recursos en los que se apoya el elevado rendimiento que caracteriza estos dispositivos programables, mejorando dos características esenciales:

1. Velocidad de ejecución.
2. Eficiencia en la compactación del código.

2.3.2. Las gamas de PIC

Una de las labores más importantes del ingeniero de diseño es la elección del microcontrolador que mejor satisfaga las necesidades del proyecto con el mínimo presupuesto.

Para resolver aplicaciones sencillas se precisan pocos recursos, en cambio, las aplicaciones grandes requieren numerosos y potentes. Siguiendo esta filosofía Microchip construye diversos modelos de microcontroladores orientados a cubrir, de forma óptima, las necesidades de cada proyecto. Así, hay disponibles microcontroladores sencillos y baratos para atender las aplicaciones simples y otros complejos y más costosos para las de mucha envergadura.

Microchip dispone de cuatro familias de microcontroladores de 8 bits para adaptarse a las necesidades de la mayoría de los clientes potenciales.

En la mayor parte se encontrarán tan solo tres familias de microcontroladores, con lo que habrán despreciado la llamada gama enana, que es en realidad una subfamilia formada por componentes pertenecientes a las otras gamas. En nuestro caso hemos preferido comentarla dado que los PIC enanos son muy apreciados en las aplicaciones de control de personal, en sistemas de seguridad y en dispositivos de bajo consumo que gestionan receptores y transmisores de señales. Su pequeño tamaño los hace ideales en muchos proyectos donde esta cualidad es fundamental.

2.3.2.1. La gama enana: PIC12C(F)XXX de 8 patitas

Se trata de un grupo de PIC de reciente aparición que ha acaparado la atención del mercado. Su principal característica es su reducido tamaño, al disponer todos sus componentes de 8 patitas. Se alimentan con un voltaje de

corriente continua comprendido entre 2,5 V y 5,5 V, y consumen menos de 2 mA cuando trabajan a 5 V y 4 MHz. El formato de sus instrucciones puede ser de 12 o de 14 bits y su repertorio es de 33 o 35 instrucciones, respectivamente.

Aunque los PIC enanos sólo tienen 8 patitas, pueden destinar hasta 6 como líneas de E/S para los periféricos porque disponen de un oscilador interno R-C.

2.3.2.2. Gama baja o básica: PIC16C5X con instrucciones de 12 bits

Se trata de una serie de PIC de recursos limitados, pero con una de las mejores relaciones precio/rendimiento. Sus versiones están encapsuladas con 18 y 28 patitas y pueden alimentarse a partir de una tensión de 2,5 V, lo que les hace ideales en las aplicaciones que funcionan con pilas teniendo en cuenta su bajo consumo (menos de 2 mA a 5 V y 4 MHz). Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo de interrupción y la Pila sólo dispone de dos niveles.

Al igual que todos los miembros de la familia PIC16/17, los componentes de la gama baja se caracterizan por poseer los siguientes recursos:

2.3.2.2.1. Sistema POR ("Power On Reset")

Todos los PIC tienen la facultad de generar una autoreinicialización o autoreset al conectarles la alimentación.

2.3.2.2.2. Perro guardián (Watchdog o WDT)

Existe un temporizador que produce un reset automáticamente si no es recargado antes que pase un tiempo prefijado. Así se evita que el sistema quede “colgado” dado en esa situación el programa no recarga dicho temporizador y se genera un reset.

2.3.2.2.3. Código de protección

Cuando se procede a realizar la grabación del programa, puede protegerse para evitar su lectura. También disponen los PIC de posiciones reservadas para registrar números de serie, códigos de identificación, prueba, etc.

2.3.2.2.4. Líneas de E/S de alta corriente

Las líneas de E/S de los PIC pueden proporcionar o absorber una corriente de salida comprendida entre 20 y 25 mA, capaz de excitar directamente ciertos periféricos.

2.3.2.2.5. Modo de reposo (Bajo consumo o “sleep”)

Ejecutando una instrucción (SLEEP), la CPU y el oscilador principal se detienen y se reduce notablemente el consumo.

Para terminar el comentario sobre los componentes de la gama baja conviene nombrar dos restricciones importantes:

1. La pila o “stack” sólo dispone de dos niveles lo que supone no poder encadenar más de dos subrutinas.

2. Los microcontroladores de la gama baja no admiten interrupciones.

2.3.2.3. Gama media: PIC16CXXX con instrucciones de 14 bits

Es la gama más variada y completa de los PIC. Abarca modelos con encapsulado desde 18 patitas hasta 68, cubriendo varias opciones que integran abundantes periféricos. Dentro de esta gama se halla el fabuloso PIC16X84 y sus variantes.

En esta gama sus componentes añaden nuevas prestaciones a las que poseían los de la gama baja, haciéndoles más adecuados en las aplicaciones complejas. Admiten interrupciones, poseen comparadores de magnitudes analógicas, convertidores A/D, puertos serie y diversos temporizadores.

El repertorio de instrucciones es de 35, de 14 bits cada una y compatible con el de la gama baja. Sus distintos modelos contienen todos los recursos que se precisan en las aplicaciones de los microcontroladores de 8 bits. También dispone de interrupciones y una Pila de 8 niveles que permite el anidamiento de subrutinas.

Encuadrado en la gama media también se halla la versión PIC14C000, que soporta el diseño de controladores inteligentes para cargadores de baterías, pilas pequeñas, fuentes de alimentación ininterrumpibles y cualquier sistema de adquisición y procesamiento de señales que requiera gestión de la energía de alimentación. Los PIC 14C000 admiten cualquier tecnología de las baterías como Li-Ion, NiMH, NiCd, Ph y Zinc.

El temporizador TMR1 que hay en esta gama tiene un circuito oscilador que puede trabajar asincrónicamente y que puede incrementarse aunque el

microcontrolador se halle en el modo de reposo (“sleep”), posibilitando la implementación de un reloj en tiempo real.

Las líneas de E/S presentan una carga “pull-up” activada por software.

2.3.2.4. Gama alta: PIC17CXXX con instrucciones de 16 bits

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectorizadas muy potente.

También incluyen variados controladores de periféricos, puertas de comunicación serie y paralelo con elementos externos, un multiplicador hardware de gran velocidad y mayores capacidades de memoria, que alcanza los 8 k palabras en la memoria de instrucciones y 454 bytes en la memoria de datos.

Quizás la característica más destacable de los componentes de esta gama es su arquitectura abierta, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos. Para este fin, las patitas sacan al exterior las líneas de los buses de datos, direcciones y control, a las que se conectan memorias o controladores de periféricos. Esta facultad obliga a estos componentes a tener un elevado número de patitas comprendido entre 40 y 44. Esta filosofía de construcción del sistema es la que se empleaba en los microprocesadores y no suele ser una práctica habitual cuando se emplean microcontroladores.

3. Presentación del Microcontrolador PIC16X84.

3.1. Descripción general

Microcontrolador diseñado por la empresa Microchip (<http://www.microchip.com>), que se fabrica en varias versiones de las cuales las más simples, pero interesantes, son la 16C84 y la 16F84. Son idénticas en cuanto a su arquitectura interna a excepción de la memoria ROM y la memoria RAM. En el primer caso contiene una memoria EEPROM de 1Kbytes de 14 bits cada una, en el segundo diseño la memoria tiene la misma capacidad pero es de tipo Flash.

Tal y como se ha comentado, la memoria EEPROM y la Flash son eléctricamente grabables y borrables, lo que permite escribir y borrar el programa bajo prueba manteniendo el microcontrolador en el mismo zócalo y usando el mismo dispositivo para grabar y borrar. Esta característica supone una gran ventaja frente a la mayoría de los microcontroladores, que tienen como memoria de programa reescribible una tipo EPROM. Se graba eléctricamente, pero para borrarlas hay que someterlas durante cierto tiempo a rayos ultravioleta, lo que implica sacar del zócalo el circuito integrado y colocarlo en un borrador de EPROM.

Otra ventaja del PIC16X84 en cuanto a simplificar el proceso de escritura, borrado y reescritura de programas, tan necesario en la fase de diseño, es su sistema de grabación de datos, que se realiza en serie. Para escribir un programa en la memoria se manda la información en serie usando sólo dos patillas, una para la señal de reloj y otra para los datos serie. A continuación exponemos las características más significativas:



Figura 3.1. Microcontrolador PIC16F84A.

Memoria de Programa: 1K x14

Memoria de Datos: 36 Bytes (PIC16C84) y 68 Bytes (PIC16F84)

Memoria de datos EEPROM: 64 Bytes

Pila: De 8 Niveles

Interrupciones: 4 tipos diferentes

Juego de Instrucciones: 35

Encapsulado: Plástico DIP de 18 Patillas

Frecuencia de Trabajo: 10Mhz Máxima

Temporizadores: Solo uno el TMR0. También tiene Perro Guardián

Líneas de E/S Digitales: 13 (5 Puerta A y 8 Puerta B)

Voltaje de Alimentación (V_{DD}): De 2 a 6 V DC

Voltaje de Grabación (V_{DD}): de 12 a 14 V DC

Existen otras variantes que se comercializan y responden a la nomenclatura genérica PIC16X8X, atendiendo a diversas características como pueden ser la frecuencia máxima de funcionamiento, el tipo de oscilador externo o el margen del voltaje de alimentación.

3.2. Aspecto externo

EL PIC16C(F)84 está fabricado con tecnología CMOS de altas prestaciones y encapsulado en plástico con 18 patillas, con la nomenclatura que se muestra. La misión de cada patilla comentada brevemente es:

V_{DD} : Tensión positiva de alimentación.

V_{SS} : Tensión conectada a tierra o negativa de alimentación.

OSC1/CLKIN: Entrada del circuito oscilador externo que proporciona la frecuencia de trabajo del microcontrolador.

OSC2/CLKOUT: Patilla auxiliar del circuito oscilador.

MCLR#: Patilla activa con nivel lógico bajo, lo que se representa con el símbolo #. Su activación origina la reinicialización o Reset del PIC. También se usa durante la grabación de la memoria para introducir por ella la tensión V_{PP} .

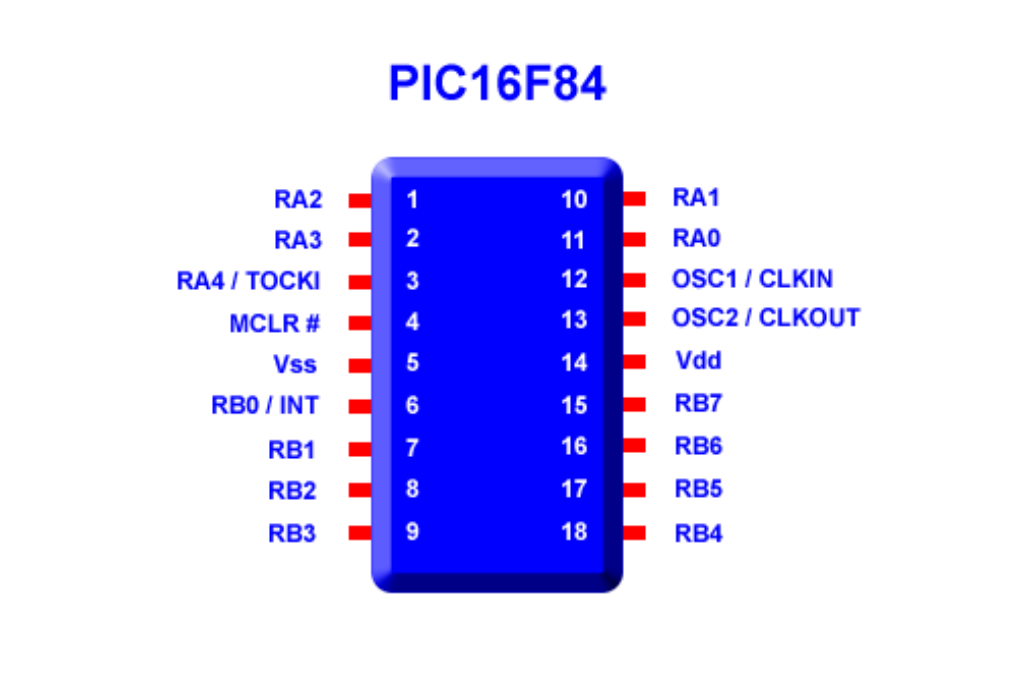


Figura 3.2. Microcontrolador PIC16X8X.

RA0-RA4: Son las 5 líneas de E/S digitales correspondientes a la Puerta A. La línea RA4 multiplexa otra función expresada por TOCKI. En ese caso sirve para recibir una frecuencia externa para alimentar al temporizador TMR0.

RB0-RB7: Son las 8 líneas de E/S digitales de la Puerta B. La línea RB0 multiplexa la función de servir como entrada a una petición externa de una interrupción.

3.3. La frecuencia de funcionamiento. El reloj o “Clock”

La frecuencia de trabajo del microcontrolador es un parámetro fundamental a la hora de establecer la velocidad de ejecución de instrucciones y el consumo de energía. Cuando un PIC16X84 funciona a 10 MHz, que es su máxima frecuencia, le corresponde un ciclo de instrucción de 400 ns, puesto que cada instrucción tarda en ejecutarse 4 períodos de reloj, o sea, $4 \times 100 \text{ ns} = 400 \text{ ns}$. Todas las instrucciones del PIC se realizan en un ciclo de instrucción, menos las de salto que tardan el doble.

Los impulsos de reloj entran por la patilla OSC1/CLKIN y se dividen por 4 internamente, dando lugar a las señales Q1, Q2, Q3 y Q4. Durante un ciclo de instrucción, que comprende las cuatro señales mencionadas, se desarrollan las siguientes operaciones:

Q1: Durante este impulso se incrementa el Contador de Programa.

Q4: Durante este impulso se busca el código de la instrucción en la memoria del programa y se carga en el registro de Instrucciones.

Q2-Q3: Durante la activación de estas dos señales se produce la decodificación y la ejecución de la instrucción.

Para conseguir ejecutar cada instrucción en un ciclo de instrucción (excepto las de salto), se aplica la técnica de segmentación o *pipe-line*, que consiste en realizar en paralelo las dos fases que comprende cada instrucción.

La estructura segmentada del procesador permite realizar al mismo tiempo la fase de ejecución de una instrucción y la de búsqueda de la siguiente. Cuando la instrucción ejecutada corresponde a un salto no se conoce cuál será la siguiente hasta que se realice, por eso en esta situación se sustituye la fase de búsqueda por un ciclo “vacío”, originando que las instrucciones de salto tarde en realizarse dos ciclos de instrucción.

3.4. Tipos de osciladores

Los PIC admiten cuatro tipos de osciladores externos para aplicarles la frecuencia de funcionamiento, se colocan entre las patillas OSC1 y OSC2. El tipo empleado debe especificarse en dos bits de la “Palabra de Configuración”, como se comentará más adelante. Los tipos que se pueden emplear son:

Oscilador RC: Es un oscilador de bajo costo formado por una simple resistencia y un condensador. Proporciona una estabilidad mediocre de la frecuencia, cuyo valor depende de los valores de los dos elementos R-C.

Oscilador HS: Es un oscilador que alcanza una alta velocidad comprendida entre 4 y 10 MHz y está basado en un cristal de cuarzo o un resonador cerámico.

Oscilador XT: Es un oscilador de cristal o resonador para frecuencias estándar comprendidas entre 100 KHz y 4 MHz.

Oscilador LP: Oscilador de bajo consumo con cristal o resonador diseñado para trabajar en un rango de frecuencias de 35 a 200 KHz.

3.5. Reinicialización o RESET

Cuando se aplica un nivel lógico bajo a la patilla MCLR# el microcontrolador reinicializa su estado. Dos acciones importantes se producen en la reinicialización o RESET:

- El Contador de Programa se carga con la dirección 0, apuntando a la primera dirección de la memoria de programa en donde deberá estar situada la primera instrucción del programa de aplicación.
- La mayoría de los registros de estado y control del procesador toman un estado conocido y determinado.

Se puede provocar el RESET de varias maneras, pero si se desea realizar manualmente, habrá que colocar, conectado a la patilla MCLR#, un circuito con un pulsador, que al ser apretado genere un nivel lógico bajo.

3.6. Arquitectura de los microcontroladores PIC16X84

Para lograr una óptima compactación de código y una velocidad superior a la de sus competidores, los microcontroladores PIC incorporan en su procesador tres de las características más avanzadas en los grandes computadores:

Procesador tipo RISC.

Ejecución segmentada.

Arquitectura HARVARD.

El juego de instrucciones se reduce a 35 y sus modos de direccionamiento se han simplificado al máximo. Con la estructura segmentada se pueden realizar simultáneamente las dos fases en que se descompone cada instrucción. Con la

arquitectura HARVARD se puede acceder de forma simultánea e independiente a la memoria de datos y a la de programa. El aislamiento y la diferenciación de los dos tipos de memoria permiten que cada uno tenga la longitud y el tamaño más adecuado. De esta forma, en el PIC16X84 la longitud de datos es de un Byte, mientras que la de las instrucciones es de 14 bits.

Otra característica relevante de los PIC es el manejo intensivo del Banco de Registros, los cuales participan de manera muy activa en la ejecución de las instrucciones. De igual forma, la memoria RAM complementa los registros internos implementando en sus posiciones registros de propósito específico y de propósito general.

La arquitectura interna del PIC16X84 se presenta a continuación y consta de siete bloques fundamentales.

1. memoria de programa EEPROM o Flash de 1 K x 14 bits.
2. memoria de datos formada por dos áreas. Una RAM donde se alojan 22 registros de propósito específico (SFR) y 36 de propósito general (GDR), y otro de tipo EEPROM de 64 Bytes.
3. Camino de datos con una ALU de 8 bits y un registro de trabajo W del que normalmente recibe un operando y envía el resultado. El otro operando puede provenir el bus de datos o del propio código de la instrucción.
4. Diversos recursos conectados al bus de datos, tales como Puertas de Entrada/Salida, Temporizador TMR0, etc.
5. Base de tiempo y circuitos auxiliares.
6. Direccionamiento de la memoria de programa e base al contador de Programa ligado a una Pila de 8 niveles de profundidad.
7. Direccionamiento directo e indirecto de la memoria RAM.

3.7. Las Puertas de E/S

Los PIC16X84 sólo disponen de dos puertas de E/S. La Puerta A posee 5 líneas, RA0-RA4, y una de ellas soporta dos funciones multiplexadas. Se trata de RA4/TOCKI, que puede actuar como línea de E/S o como la patilla por la que se reciben los impulsos que debe contar TMR0. La Puerta B tiene 8 líneas, RB0-RB7, y también tiene una con funciones multiplexadas, la RB0/INT, que, además de línea típica de E/S, también sirve como patilla por la que reciben los impulsos externos que provocan una interrupción.

Cada línea de E/S puede configurarse independientemente como entrada o como salida, según se ponga a 1 o a 0, respectivamente, el bit asociado del registro de configuración de cada puerta (TRISA y TRISB). Se llaman PUERTAA y PUERTAB los registros que guardan la información que entra o sale por la puerta y ocupan las direcciones 5 y 6 del banco 0 de la memoria de datos. Los registros de configuración TRISA y TRISB ocupan las mismas direcciones pero en el banco 1. Al reiniciarse el PIC todos los bits de los registros TRIS quedan a 1, con lo que las líneas de las puertas quedan configuradas como entradas. Cada línea de salida puede suministrar una corriente máxima de 20 mA y si es de entrada puede absorber hasta 25 mA. Al existir una limitación en la disipación máxima de la potencia del chip se restringe la corriente máxima de absorción de la Puerta A a 80 mA y la de suministro a 50 mA. La Puerta B puede absorber un máximo de 150 mA y suministrar un total de 100 mA.

3.7.1. La Puerta A

Las líneas RA3-RA0 admiten niveles de entrada TTL y de salida CMOS. La línea RA4/TOCKI dispone de un circuito Trigger Schmitt que proporciona una buena inmunidad al ruido y la salida tiene drenador abierto. RA4 multiplexa su función de E/S con la entrada de impulsos externos para el TMR0. En el circuito

se muestra la adaptación de las patillas RA3-RA0 a las señales internas del procesador.

Cuando se lee una línea de la Puerta A se recoge el nivel lógico que tiene en ese momento. Las líneas cuando actúan como salidas están conectadas a unos biestable, lo que significa que sus patillas sacan el nivel lógico que se haya cargado por última vez en el registro PUERTAA. La escritura de una puerta, tal y como se ve, implica que primero se deposita el nivel lógico en la línea correspondiente del bus interno de datos y se activa la señal WRITE, lo que origina el almacenamiento de dicho nivel en el biestable. En esta situación, el biestable de configuración (TRIS) debería tener valor 0 para que actuase como salida. Con estos valores la puerta OR tendría un 0 a su salida y la AND también. Estos valores producen la conducción de los transistores MOS, en función del valor escrito, dando como salida final V_{DD} o V_{SS} (1 ó 0 respectivamente).

Si una línea actúa como entrada, el nivel lógico depositado en ella desde el exterior pasa a la línea correspondiente del bus interno de datos cuando se activa la señal READ y se hace conductor el dispositivo triestado que les une. Al programarse como entrada, los dos transistores MOS de salida quedan bloqueados y la línea de alta impedancia. Téngase en cuenta que cuando se lee una línea de entrada se obtiene el estado actual que tiene su patilla correspondiente (almacenado en el tercer biestable) y no el que haya almacenado en el biestable de datos. La información presente en una línea de entrada se muestrea al iniciarse el ciclo de instrucción y debe mantenerse estable durante su desarrollo.

3.7.2. La Puerta B

Consta de 8 líneas bidireccionales de E/S, RB7-RB0, cuya información se almacena en el registro PUERTAB, que ocupa la dirección 6 del banco 0. El registro de configuración TRISB ocupa la misma dirección pero del banco 1. La

línea RB0/INT tiene dos funciones multiplexadas; además de patilla de E/S actúa como la patilla para la petición de una interrupción externa, cuando se autoriza esta función mediante la adecuada programación del registro INTCON.

A todas las líneas de esta puerta se las permite conectar una resistencia *pull-up* de elevado valor con el positivo de la alimentación. Para este fin hay que programar en el registro OPTION el BIT RBPU# = 0, afectando la conexión de la resistencia a todas las líneas. Con el *Reset* todas las líneas quedan configuradas como entradas y se desactivan las resistencias de *pull-up*.

Las cuatro líneas de más peso, RB7-RB4, pueden programarse para soportar una misión especial. Cuando las 4 líneas actúan como entradas se las puede programar para generar una interrupción si alguna de ellas cambia su estado lógico. Esta posibilidad es muy práctica en el control de teclados. A continuación se muestra el esquema de conexión entre las patillas RB7-RB4 y las líneas correspondientes del bus interno.

El estado de las patillas RB7-RB4 en modo de entrada se compara con el valor antiguo que tenían y que se había almacenado en un biestable durante la última lectura de la Puerta B. El cambio de estado en alguna de esas líneas origina una interrupción y la activación del señalizador RBIF. La línea RB6 también se utiliza para la grabación serie de la memoria de programas y sirve para soportar la señal de reloj. La línea RB7 constituye la entrada de los datos en serie.

3.8. La memoria EEPROM de Datos

Los PIC16X84 tienen 64 Bytes de memoria EEPROM de datos, donde se pueden almacenar datos y variables que interesa que no se pierdan cuando se desconecta la alimentación al sistema. Soporta 1.000.000 de ciclos de

escritura/borrado y es capaz de guardar la información sin alterarla más de 40 años. La memoria EEPROM no está mapeada en la zona de la memoria de datos donde se ubican los registros SFR y GPR.

3.9. Interrupciones

Las llamadas a las subrutinas mediante instrucciones del tipo CALL son desviaciones del flujo de control del programa originadas por instrucciones, por lo que se consideran síncronas. Se producen cada vez que se ejecuta dicha instrucción. Las interrupciones son desviaciones del flujo de control del programa originadas asíncronamente por diversos sucesos que no se hallan bajo la supervisión de las instrucciones. Dichos sucesos pueden ser externos al sistema, como la generación de un flanco o nivel activo en una patilla del microcontrolador, o bien, internos, como el desbordamiento de un contador.

El comportamiento del microcontrolador ante la interrupción es similar al de la instrucción tipo CALL de llamada a subrutina. En ambos casos se detiene la ejecución del programa en curso, se salva la dirección actual del PC en la Pila y se carga el PC con una dirección, que en el caso de CALL viene acompañando a la misma instrucción, y en el caso de una interrupción es una dirección “reservada” de la memoria de código, llamada Vector de Interrupción que da paso a un trozo de código denominado Rutina de Servicio de la Interrupción (RSI).

La RSI suele comenzar guardando en la memoria de datos algunos registros específicos del procesador. Concretamente aquellos que la RSI va a emplear y va a alterar su contenido. Antes del retorno al programa principal se recuperan los valores guardados y se restaura completamente el estado del procesador. Algunos procesadores salvan estos registros en la Pila, pero los PIC no disponen de instrucciones para meter (*push*) y sacar (*pop*) información de la Pila, utilizando para este fin registros de propósito general de la memoria de datos.

Los PIC16X84 pueden ser interrumpidos por 4 causas diferentes, pero todas ellas desvían el flujo de control a la dirección 0004 H, por lo que otra de las operaciones iniciales de la RSI es averiguar cuál de las posibles causas ha sido la responsable de la interrupción en curso, para ello se exploran los señalizadores de las fuentes de interrupción. Otro detalle importante en la RSI de los PIC16X84 es que estos microcontroladores poseen un BIT GIE (*Global Interrupt Enable*) que cuando vale 0 prohíbe todas las interrupciones. Pues bien, al comenzar la RSI dicho BIT GIE se pone automáticamente a 0, con objeto de no atender nuevas interrupciones hasta que se termine la que ha comenzado. En el retorno final de la interrupción, GIE pasa a valer automáticamente 1 para volver a tener en cuenta las interrupciones.

Antes del retorno conviene borrar el señalizador de la causa de interrupción que se ha atendido, porque si bien los señalizadores se ponen a 1 automáticamente en cuanto se produce la causa que indican, la puesta a 0 se hace por programa.

3.9.1. Causas de Interrupción

Los PIC16X84 tienen 4 causas o fuentes posibles de interrupción:

1. Activación de la patilla RB0/INT.
2. Desbordamiento del temporizador TMR0.
3. Cambio de estado de una de las patillas de las peso (RB7-RB4) de la Puerta B.
4. Finalización de la escritura en la EEPROM de datos.

Cuando ocurre cualquiera de los 4 sucesos indicados se origina una petición de interrupción, que si se acepta y se atiende comienza depositando el valor del PC actual en la Pila, poniendo el BIT GIE = 0 y cargando en el PC el valor 0004 H, que es el Vector de Interrupción donde se desvía el flujo de control. Cada fuente de interrupción dispone de un señalizador o “flag”, que es un BIT que se pone automáticamente a 1 cuando se produce. Además cada fuente de interrupción tiene otro BIT de permiso, que según su valor permite o prohíbe la realización de dicha interrupción.

4. Comunicación Inalámbrica

4.1. Introducción

El simple hecho de ser seres humanos nos hace desenvolvernos en medios donde tenemos que estar comunicados. La conexión inalámbrica de computadoras mediante Ondas de Radio o Luz Infrarroja, actualmente está siendo ampliamente investigada. Las Redes Inalámbricas facilitan la operación en lugares donde las computadoras no pueden permanecer en un solo lugar, como en universidades o en oficinas. Pero la realidad es que esta tecnología está todavía en pañales ya que tiene un enorme potencial, aun existen problema de regulación y normas que solo los países mas avanzados están integrando recientemente, es el caso de nuestro país el cual casi no tiene legislación sobre el uso del campo aéreo para transmisiones inalámbricas.

Actualmente el uso de equipos inalámbricos va aumentando enormemente, ya sea el uso de teléfonos celulares, controles a distancia, alarmas, agendas electrónicas, periféricos para computadores como teclados, ratones, joystick, etc.

No se espera que las redes inalámbricas lleguen a remplazar a las redes cableadas. Estas ofrecen velocidades de transmisión mayores que las logradas con la tecnología inalámbrica. Mientras que las redes inalámbricas actuales ofrecen velocidades de 11, 22, 54, 108 Mbps, las redes cableadas ofrecen velocidades de 1 Gbps y se espera que alcancen velocidades aun superiores. Los sistemas de Cable de Fibra Óptica logran velocidades aún mayores.

Sin embargo se pueden mezclar las redes cableadas y las inalámbricas, y de esta manera generar una "Red Híbrida" y poder resolver los últimos metros hacia la estación. Se puede considerar que el sistema cableado sea la parte principal y la inalámbrica le proporcione movilidad adicional al equipo y el operador se pueda desplazar con facilidad dentro de un almacén o una oficina.

4.2. Comunicación de Radiofrecuencia

Se le llama así a las ondas aéreas electromagnéticas las que sirven para comunicar datos desde un punto a otro. Son portadoras de radio porque desempeñan la función de entregar energía al receptor. Los datos que se transmiten son sobrepuestos sobre la portadora de radio para que pueda extraer de manera precisa por el receptor. Es a lo que se conoce como la modulación de la portadora por la información que se transmite. Después de que los datos son sobrepuestos (modulados) en el transportador de radio, la señal de radio ocupa más de una sola frecuencia, donde la frecuencia de la información modulada se agrega a la portadora. Múltiples portadoras de radio pueden coexistir en el mismo espacio a la vez, sin que haya interferencia, así las ondas de radio se transmiten sobre radiofrecuencias diferentes.



Figura 4.2. Transmisor y receptor de radio.

4.3. Comunicación de Microondas

Las comunicaciones inalámbricas mediante microondas permiten reducir al mínimo la posibilidad de interferencia con otras señales de radio, ya que funcionan en frecuencias mayores de un gigahertz. De hecho, las bandas de frecuencias más comunes son las de 2,4, 6 y 6,8 Ghz. En un enlace de microondas a 140 Mbits/s podemos tener hasta 1920 canales de voz o bien varias comunicaciones de 2 Mbits/s.

El problema es que para este tipo de comunicaciones es necesario que haya una línea de visión directa, por lo que se debe disponer de antenas en torres elevadas que aseguren un camino directo con pocos repetidores. Es por esto que en cada lugar donde instalemos una red de microondas se debe tener muy en cuenta la orografía del terreno. A parte de esto, también pueden influir en la comunicación lluvias intensas o el paso de una bandada de pájaros, aunque no es muy frecuente que esto ocurra.

El MMDS (Microwave Multipoint Distribution System, Sistema de Distribución Multipunto de Microondas) constituye una alternativa viable al cable. Se puede utilizar las MMDS para proveer acceso de alta velocidad a Internet. En un sistema MMDS los datos son transmitidos vía microondas utilizando un esquema de multiplexión por división de tiempo (TDM, Time Division Multiplexing). Cada suscriptor dispone de un módem inalámbrico, el cual monitorea la señal recibida en espera de la información dirigida a un usuario particular. Los datos del canal de retorno son enviados utilizando la línea. El canal de información dirigida al usuario (downstream) está compartido, por lo que es necesario algún tipo de algoritmo para administrar el empleo del canal por parte de cada usuario, aunque la implementación de este algoritmo no conlleva más dificultad.



Figura 4.3. Antena Microondas.

4.4. Comunicación Vía Satélite

Un satélite de comunicaciones hace la labor de repetidor: una estación terrestre envía la información al satélite en forma de señales de una frecuencia determinada y el satélite las retransmite a otra estación terrestre mediante una frecuencia distinta.

Las transmisiones de los satélites pueden ser recibidas por todos aquellos que se encuentren dentro del cono de radiación del satélite. El utilizar frecuencias distintas al enviar los datos al satélite y al recibirlos evita que se produzcan interferencias. En general, la frecuencia con la que se trabaja en una comunicación por satélite suele encontrarse en la banda de los gigahertz.

La mayoría de los satélites de comunicaciones están situados en una órbita denominada geostacionaria (a unos 35000 Km.) en la que el satélite gira a la

misma velocidad que la tierra, de modo que parece que no se mueva. Esto permite que las antenas terrestres puedan permanecer orientadas hacia un satélite que mantendrá una posición casi estable. En la actualidad hay un problema de ocupación de la órbita geostacionaria, por lo que se hace necesario desplazar de esta órbita los satélites que no funcionan. Debido a esta sobre ocupación, se ha acordado disminuir el espacio entre los satélites, lo que obliga a centrar una mayor atención en evitar que se produzcan interferencias entre los satélites que están próximos.

En una comunicación vía satélite existen una serie de características:

- Existe un retardo de unos 0.5 segundos, debido lógicamente a la distancia que han de recorrer las señales
- La comunicación con el satélite puede ser interceptada por cualquiera que disponga de un receptor en las proximidades de la estación, por lo que es necesario utilizar técnicas de encriptación para garantizar la privacidad de los datos
- Los satélites geostacionarios pasan por períodos en los que no pueden funcionar, como en un eclipse de Sol en el que se corta el suministro de energía solar con la que se alimenta el satélite, lo que provoca el uso de unas baterías de reserva que en general no permiten el correcto funcionamiento del satélite
- Los satélites tienen una vida media de siete a 10 años, pero pueden sufrir fallos que provocan su salida de servicio. Es, por tanto, necesario disponer de un medio alternativo de servicio en caso de cualquier eventualidad
- Las estaciones terrestres suelen estar lejos de los usuarios y a menudo se necesitan enlaces caros de alta velocidad

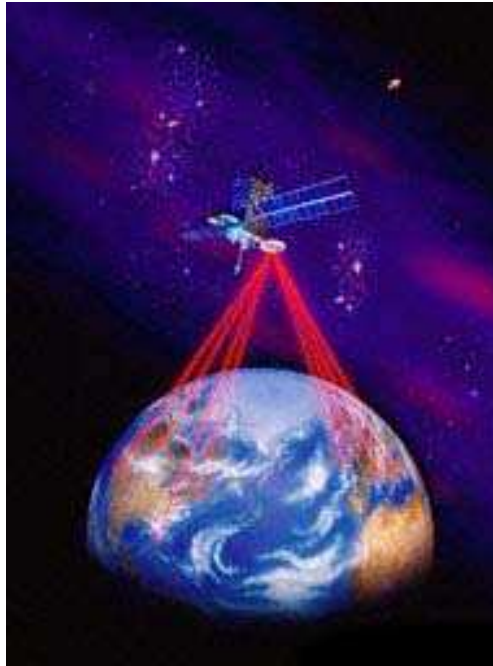


Figura 4.4. Ejemplo de Satélite.

4.5. Comunicación Infrarroja

Las redes de luz infrarroja están muy limitadas por el espacio, por lo que únicamente se utilizan en una habitación o en un piso, aunque se pueden crear redes un poco más grandes colocando los receptores en las ventanas de los edificios. Además es necesario que exista un contacto visual, ya que la luz viaja en línea recta. Pero se puede superar este inconveniente colocando espejos que modifiquen la dirección de la luz o utilizando otras técnicas. Por otra parte, este sistema no tiene el problema de las frecuencias.

La comunicación mediante infrarrojos está bastante extendida en nuestra vida cotidiana, encontramos infrarrojos en los mandos a distancia, en la alarma del

automóvil y ya están bastante extendidos los periféricos informáticos que funcionan mediante este sistema.

En realidad, las redes de comunicación mediante infrarrojos utilizan el mismo principio: un "transreceptor" envía un haz de luz infrarroja hacia otro que la recibe. La luz que se envía se codifica y se decodifica utilizando un protocolo existente. Con este sistema se consiguen velocidades de 10 Mbits/seg e incluso más. Ya existen multitud de productos que crean redes de este tipo, teniendo en cuenta que son redes de área local.

Las estaciones con tecnología infrarroja pueden usar tres modos diferentes de radiación para comunicarse: punto-a-punto, cuasi-difuso y difuso.

4.5.1. Modo Punto a Punto

En el modo punto-a-punto los patrones de radiación del emisor y del receptor deben de estar lo más cerca posible, para que su alineación sea correcta. Como resultado, el modo punto-a-punto requiere una línea-de-vista entre las dos estaciones a comunicarse.

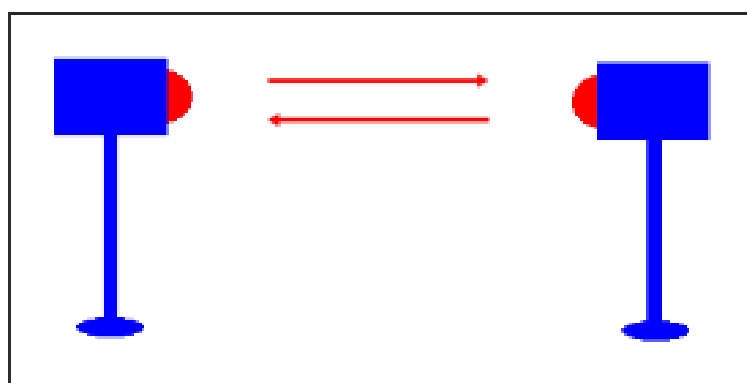


Figura 4.5.1. Modo Punto a Punto.

4.5.2. Modo Cuasi-Difuso

A diferencia del modo punto-a-punto, el modo cuasi-difuso y difuso son de emisión radial, o sea que cuando una estación emite una señal Óptica, ésta puede ser recibida por todas las estaciones al mismo tiempo en la célula. En el modo cuasi-difuso las estaciones se comunican entre si, por medio de superficies reflejantes. No es necesaria la línea-de-vista entre dos estaciones, pero si deben de estarlo con la superficie de reflexión. Además es recomendable que las estaciones estén cerca de la superficie de reflexión.

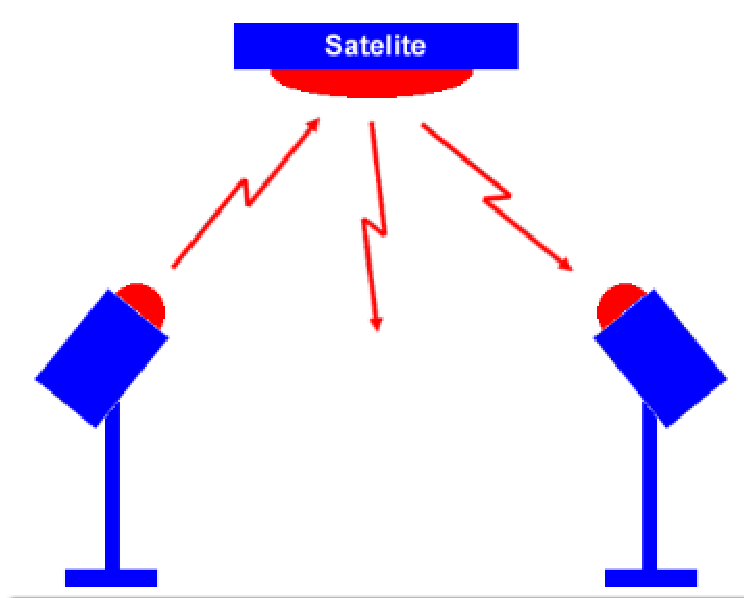


Figura 4.5.2. Modo Cuasi-Difuso.

4.5.3. Modo Difuso

En el modo difuso, el poder de salida de la señal óptica de una estación, debe ser suficiente para llenar completamente el total del cuarto, mediante múltiples reflexiones, en paredes y obstáculos del cuarto. Por lo tanto la línea-de-vista no es necesaria y la estación se puede orientar hacia cualquier lado. El modo difuso es el más flexible, en términos de localización y posición de la estación, sin embargo esta flexibilidad esta a costa de excesivas emisiones ópticas.

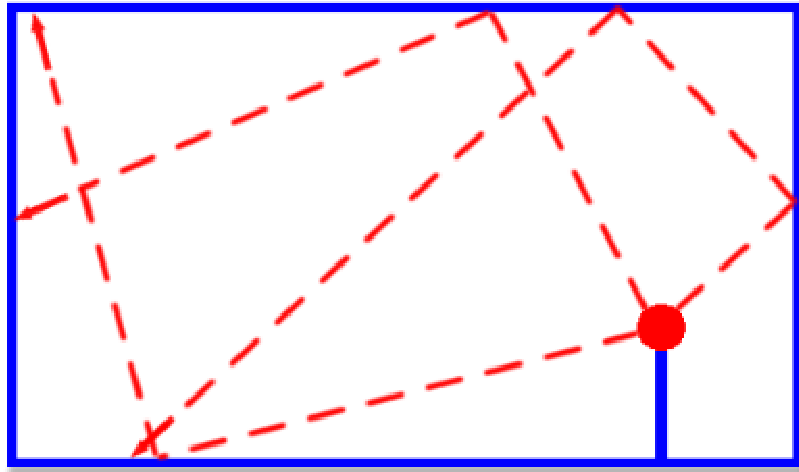


Figura 4.5.3. Modo Difuso.

4.6. Modulación

La modulación es el proceso que sufre la información, ya sea esta un sonido, una imagen o bien solo datos que se desea hacer llegar a un receptor a través de una onda electromagnética o un cable conductor.

La modulación nace de la necesidad de transportar una información a través de un canal de comunicación a la mayor distancia y menor costo posible. Este es un proceso mediante el cual dicha información (onda moduladora) se inserta a un soporte de transmisión.

Ventajas de la Modulación

- Facilita la PROPAGACIÓN de la señal de información por cable o por el aire.
- Ordena el RADIOESPECTRO, distribuyendo canales a cada información distinta.
- Disminuye DIMENSIONES de antenas.
- Optimiza el ancho de banda de cada canal.
- Evita INTERFERENCIA entre canales.
- Protege a la Información de las degradaciones por RUIDO.
- Define la CALIDAD de la información transmitida.

4.6.1. Modulación para facilidad de radiación:

Una radiación eficiente de energía electromagnética requiere de elementos radiadores (antenas) cuyas dimensiones físicas serán por lo menos de 1/10 de su longitud de onda. Pero muchas señales, especialmente de audio, tienen componentes de frecuencia del orden de los 100 Hz o menores, para lo cual

necesitarían antenas de unos 300 Km. de longitud si se radiaran directamente. Utilizando la propiedad de traslación de frecuencias de la modulación, estas señales se pueden sobreponer sobre una portadora de alta frecuencia, con lo que se logra una reducción sustancial del tamaño de la antena. Por ejemplo, en la banda de radio de FM, donde las portadoras están en el intervalo de 88 a 108 MHz, las antenas no deben ser mayores de un metro.

4.6.2. Modulación para reducir el ruido y la interferencia

Se ha dicho que es imposible eliminar totalmente el ruido del sistema. Y aunque es posible eliminar la interferencia, puede no ser práctico. Por fortuna, ciertos tipos de modulación tiene la útil propiedad de suprimir tanto el ruido como la interferencia. La supresión, sin embargo, ocurre a un cierto precio, generalmente requiere de un ancho de banda de transmisión mucho mayor que el de la señal original, de ahí la designación del ruido de banda ancha. Este convenio de ancho de banda para la reducción del ruido es uno de los intereses y a veces desventajosos aspectos del diseño de un sistema de comunicación.

4.6.3. Modulación por asignación de frecuencia

El propietario de un aparato de radio o televisión puede seleccionar una de varias estaciones, aún cuando todas las estaciones estén transmitiendo material de un programa similar en el mismo medio de transmisión. Es posible seleccionar y separar cualquiera de las estaciones, dado que cada una tiene asignada una frecuencia portadora diferente. Si no fuera por la modulación, solo operaría una estación en un área dada. Dos o más estaciones que transmitan directamente en el mismo medio, sin modulación, producirán una mezcla inútil de señales interferentes.

4.6.4. Modulación para multicanalización

A menudo se desea transmitir muchas señales en forma simultánea entre dos puntos. Las técnicas de multicanalización son formas intrínsecas de modulación, permiten la transmisión de múltiples señales sobre un canal, de tal manera que cada señal puede ser captada en el extremo receptor. Las aplicaciones de la multicanalización comprenden telemetría de datos, emisión de FM estereofónica y telefonía de larga distancia. Es muy común, por ejemplo, tener hasta 1,800 conversaciones telefónicas de ciudad a ciudad, multicanalizadas y transmitidas sobre un cable coaxial de un diámetro menor de un centímetro.

4.6.5. Modulación para superar las limitaciones del equipo

El diseño de un sistema queda generalmente a la disponibilidad de equipo, el cual a menudo presenta inconvenientes en relación con las frecuencias involucradas. La modulación se puede usar para situar una señal en la parte del espectro de frecuencia donde las limitaciones del equipo sean mínimas o donde se encuentren más fácilmente los requisitos de diseño. Para este propósito, los dispositivos de modulación se encuentran también en los receptores, como ocurre en los transmisores.

4.6.6. Modulación en Amplitud y Bandas laterales de la Modulación en Amplitud

Las estaciones de radio de banda de onda larga estándar (540kHz a 1620kHz) utilizan la modulación en amplitud (AM) para transmitir información de audio (voz, música, etc.) en la onda portadora de RF. AM es una mezcla de señales de AF y RF, de manera que las variaciones de amplitud de la señal de AF (modulación) alteran la amplitud de la señal de RF (portadora). La onda modulada de RF tiene la forma simétrica arriba y debajo de la línea de referencia cero. Las frecuencias laterales superior e inferior iguales a la suma y diferencia de las dos frecuencias originales se generan en el proceso de modulación y aparecen en el espectro de frecuencias inmediatamente arriba y debajo de la frecuencia de la portadora. Por ejemplo, si la frecuencia de la portadora es de 1MHz y la frecuencia de modulación es de 1kHz, se producen arriba y debajo de la frecuencia de portadora las frecuencias laterales superior e inferior de 1.001Mhz y 999kHz respectivamente. Si la señal de modulación varía en frecuencia, crea una banda de frecuencias a cada lado de la portadora, conocida como la banda lateral superior y banda lateral inferior. El ancho de banda total de la señal modulada queda determinado por la frecuencia de modulación más alta y es igual a la suma de las dos bandas laterales. El ancho de canal total de la señal portadora es igual a la frecuencia de portadora más y menos las dos bandas laterales.

4.6.7. Modulación de frecuencia

La modulación en frecuencia (FM) es el proceso de combinar una señal de AF (Audio Frecuencia) con otra de RF (Radio Frecuencia) en el rango de frecuencias entre 88MHz y 108MHz, tal que la amplitud de la AF varíe la frecuencia de la RF.

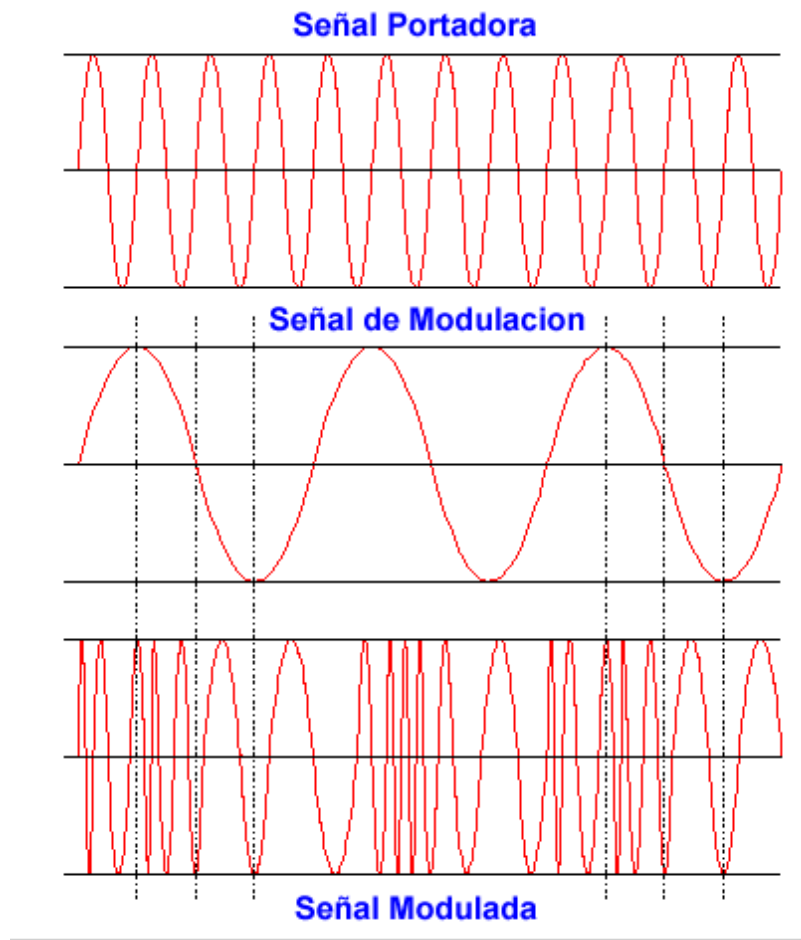


Figura 4.6.7. Ejemplo de señal Modulada.

Si la señal de modulación varía en frecuencia, no tiene efecto en las excursiones máxima y mínima de la frecuencia de portadora, sino que solo determina la rapidez o lentitud con que ocurren las variaciones en la frecuencia. Es decir, que una frecuencia mas baja de modulación provoca que ocurran variaciones a una tasa más lenta, y una frecuencia mas alta de modulación hace que ocurran a una tasa más rápida. Sin embargo, las variaciones en amplitud de la señal de modulación si afectan las excursiones máxima y mínima de la frecuencia portadora. Una señal de mayor amplitud provoca un mayor cambio en la frecuencia y una señal más pequeña provoca un cambio menor en la frecuencia.

5. PROTOCOLOS DE COMUNICACIÓN

5.1. Introducción

Así como existen normas de conductas y comunicación para las personas, también las hay para la comunicación entre dispositivos y se conocen con el nombre de protocolos. Los protocolos de comunicaciones definen las normas que posibilitan que se establezca una comunicación entre varios equipos o dispositivos, ya que estos equipos pueden ser diferentes entre sí.

Por ejemplo computadores, teléfonos, Routers, etc. Los protocolos gobiernan el formato, sincronización, secuencia y control de errores. Sin estas reglas, los dispositivos no podrían detectar la llegada de bits. Los protocolos cuentan con algunos elementos que lo definen, como los siguientes:

- **Sintaxis:** formato, codificación y niveles de señal de datos.
- **Semántica:** información de control y gestión de errores.
- **Temporización:** coordinación entre la velocidad y orden secuencial de las señales.

Pero los protocolos van más allá que sólo una comunicación básica. Por ejemplo, supongamos que se desea enviar algún archivo de un computador a otro, se podría enviar todo el archivo de una sola vez, pero si un error ocurre durante la transmisión, todo el archivo tendría que enviarse de nuevo. Para resolver este tipo de problemas, el archivo es partido o dividido en piezas pequeñas llamados "paquetes" agrupados de una cierta manera. Esto significa que cierta información debe ser agregada al paquete para decirle al receptor

donde pertenece cada grupo en relación con los otros (por ejemplo en número del paquete, el tamaño, etc.). Para mejorar la confiabilidad de la información de sincronización y corrección se le deberá ser agregada al paquete algún tipo de información útil, algunos bits para poder tener algún tipo de corrección.

Debido a su complejidad, la comunicación entre dispositivos es separada en pasos. Cada paso tiene sus propias reglas de operación y, consecuentemente, su propio protocolo. Esos pasos deben de ejecutarse en un cierto orden, de arriba hacia abajo en la transmisión y de abajo hacia arriba en la recepción (Por ejemplo en el modelo OSI, en donde los datos van desde la capa de aplicación hasta la capa física cuando transmite y desde la capa física hasta la capa de aplicación cuando recibe.). Debido al arreglo jerárquico de los protocolos, el término "pila de protocolos" (protocol Stack) es comúnmente usado para describir esos pasos. Una pila de protocolos, por lo tanto, es un conjunto de reglas de comunicación, y cada paso en la secuencias tiene su propio subconjunto de reglas.

Es software que reside en la memoria de un computador o en la memoria de un dispositivo de transmisión, como una tarjeta de red. Cuando los datos están listos para transmitirse, este software es ejecutado. EL software prepara los datos para la transmisión y configura la transmisión en movimiento. En la parte receptora, el software toma los datos y los prepara para el computador, desechando toda la información agregada, y tomando sólo la información útil.

Existen un montón de protocolos, por ejemplo en una red usando sistemas operativos Novell se comunican a través de sus propias reglas (su propio protocolo llamado IPX/SPX), Microsoft lo hace a su manera (NetBEUI), así como IBM (NetBIOS) y así cada sistema de cómputo tiene sus propios protocolos y sus propias reglas para comunicarse. Debido a que el transmisor y el receptor tienen que hablar el mismo idioma (el mismo protocolo sino no se van a entender).

5.2. Protocolo Estándar

Un protocolo estándar es un conjunto de reglas que han sido ampliamente usadas e implementadas por diversos fabricantes, usuarios, y organismos oficiales (Por ejemplo la IEEE, ISO, ITU). Idealmente, un protocolo estándar debe permitir a los computadores o dispositivos comunicarse entre sí, aún cuando estos sean de diferentes fabricantes. Los computadores o dispositivos no tienen que usar un protocolo estándar para comunicarse, pero si estos utilizan un protocolo propietario entonces ellos pueden sólo comunicarse con los equipos de su mismo tipo.

Existen muchos protocolos estándares, ninguno de ellos puede ser universal, pero los que son exitosos se están moviendo para cumplir con el modelo de referencia OSI. Los estándares y protocolos asociados con el modelo de referencia OSI están bajo el concepto de Sistemas Abiertos para enlazar literalmente a decenas de computadoras distintas encontradas en oficinas a través del mundo.

5.3. FUNCIONES DE UN PROTOCOLO

5.3.1. Segmentación y ensamblado

Generalmente es necesario dividir los bloques de datos en unidades pequeñas e iguales en tamaño, y este proceso se le llama segmentación. El bloque básico de segmento en una cierta capa de un protocolo se le llama PDU (Unidad de datos de protocolo).

5.3.2. Encapsulado

Se trata del proceso de adherir información de control al segmento de datos. Esta información de control es el direccionamiento del emisor/receptor, código de detección de errores y control de protocolo, etc.

5.3.3. Control de conexión

Hay bloques de datos sólo de control y otros de datos y control. Cuando se utilizan datagramas, todos los bloques incluyen control y datos ya que cada paquete de datos se trata como independiente. En circuitos virtuales hay bloques de control que son los encargados de establecer la conexión del circuito virtual. Hay protocolos más sencillos y otros más complejos, por lo que los protocolos de los emisores y receptores deben de ser por lo menos compatible.

5.3.4. Entrega ordenada

El envío de paquete de datos puede acarrear el problema de que si hay varios caminos posibles, lleguen al receptor paquetes de datos desordenados o

repetidos, por lo que el receptor debe de tener un mecanismo para reordenarlos. Hay sistemas que tienen un mecanismo de numeración.

5.3.5. Control de flujo

Hay controles de flujo de parada y espera o de ventana deslizante. El control de flujo es necesario en varios protocolos o capas, ya que el problema de saturación del receptor se puede producir en cualquier capa del protocolo. Por ejemplo:

- Manejo de contención de bloques.
- Regulación del tráfico.
- Retransmisión de bloques.
- Control por pasos y de extremo a extremo (el error puede verificarse en cada paso o al final del enlace depende del algoritmo de control de error).

5.3.6. Control de errores

Generalmente se utiliza un temporizador para retransmitir una trama una vez que no se ha recibido confirmación después de expirar el tiempo del temporizador. Cada capa de protocolo debe de tener su propio control de errores mediante algún tipo de algoritmo de verificación y corrección de errores como por ejemplo:

-
- **Checksum.**
 - **VRC.**
 - **LRC.**
 - **CRC.**
 - **Etc.**

5.3.7. Direccionamiento

Cada estación o dispositivo intermedio de almacenamiento debe tener una dirección única. A su vez, en cada terminal o sistema final puede haber varios agentes o programas que utilizan la red, por lo que cada uno de ellos tiene asociado un puerto.

5.3.8. Multiplexación

Es posible multiplexar las conexiones de una capa hacia otra, es decir que de una única conexión de una capa superior, se pueden establecer varias conexiones en una capa inferior (y al revés).

5.3.9. Servicios de transmisión

Los servicios que puede prestar un protocolo son los siguientes:

5.3.9.1. Prioridad

Hay mensajes (los de control) que deben tener prioridad respecto a otros.

5.3.9.2. Grado de servicio

Hay datos que deben de retardarse y otros acelerarse (vídeo).

5.3.9.3. Seguridad.

5.4. Tipos de Protocolos

5.4.1. Protocolos orientados a la conexión

Los protocolos orientado a la conexión mantienen la información de estado, acerca del dialogo que están manteniendo. Esta información del estado de la conexión soporta control de error, secuencia y control de flujo entre las correspondientes entidades. Es decir, la entidad receptora le avisa a la entidad transmisora si la información útil llego correctamente, si no es así también le avisa que vuelva a retransmitir (un ejemplo de protocolo orientado a la conexión, es el protocolo TCP).

Los protocolos orientados a conexión operan en tres fases. La primera fase es la fase de configuración de la conexión, durante la cual las entidades correspondientes establecen la conexión y negocian los parámetros que definen la conexión (Por ejemplo el protocolo TCP usa Hand Shake de tres vías para establecer la comunicación).

La segunda fase es la fase de transferencia de datos, durante la cual las entidades correspondientes intercambian mensajes (información útil) bajo el amparo de la conexión. Finalmente, la última fase, fase de liberación de la conexión, en la cual ambas entidades se ponen de acuerdo para terminar la conexión.

Un ejemplo de la vida diaria de un protocolo orientado a la conexión es una llamada telefónica. La parte originadora (el que llama) deberá primero marcar el número del teléfono usuario (abonado) destino. La infraestructura telefónica deberá asignar el circuito extremo-extremo, entonces hace timbrar el teléfono del usuario destino. Al momento que éste levanta el teléfono se establece la llamada o conexión y ambos empiezan a conversar. En algún momento, alguno de los dos cuelga, y la conexión de termina y se libera el circuito. Entonces se termina la llamada.

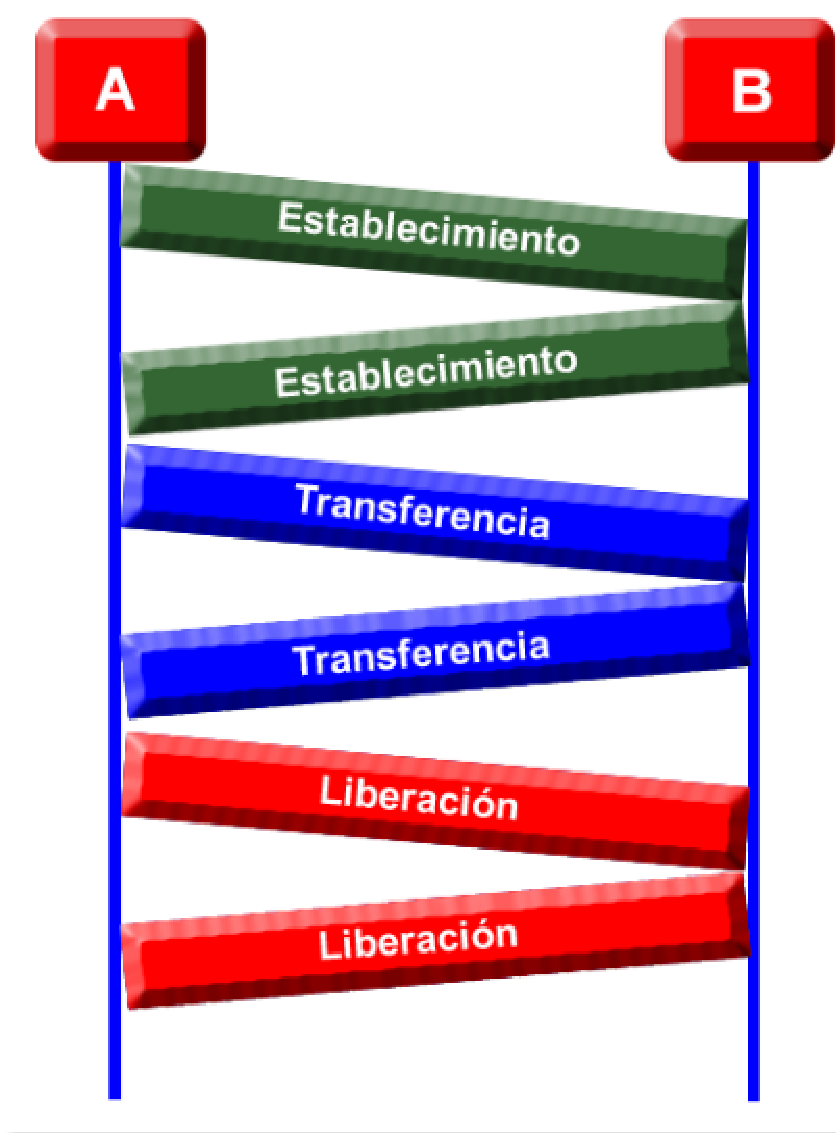


Figura 5.4.1. Esquema de Establecimiento, transferencia y liberación de un protocolo orientado a la conexión.

5.4.2. Protocolos orientados a la no conexión

Los protocolos orientados a la no conexión, están siempre en la fase de transferencia de datos, y no les interesa las fases restantes de configuración y liberación de una conexión. Se emplean en aplicaciones donde no se requiera mucha precisión. Tal es el caso de la voz, música o el video. Pero en cambio en aplicaciones donde se requiera mucha precisión (Por ejemplo transacciones electrónicas bancarias, archivos de datos, comercio electrónico, etc.) se utilizarían los protocolos orientados a conexión. Unos ejemplos de protocolos orientados a la no conexión son el protocolo UDP e IP.

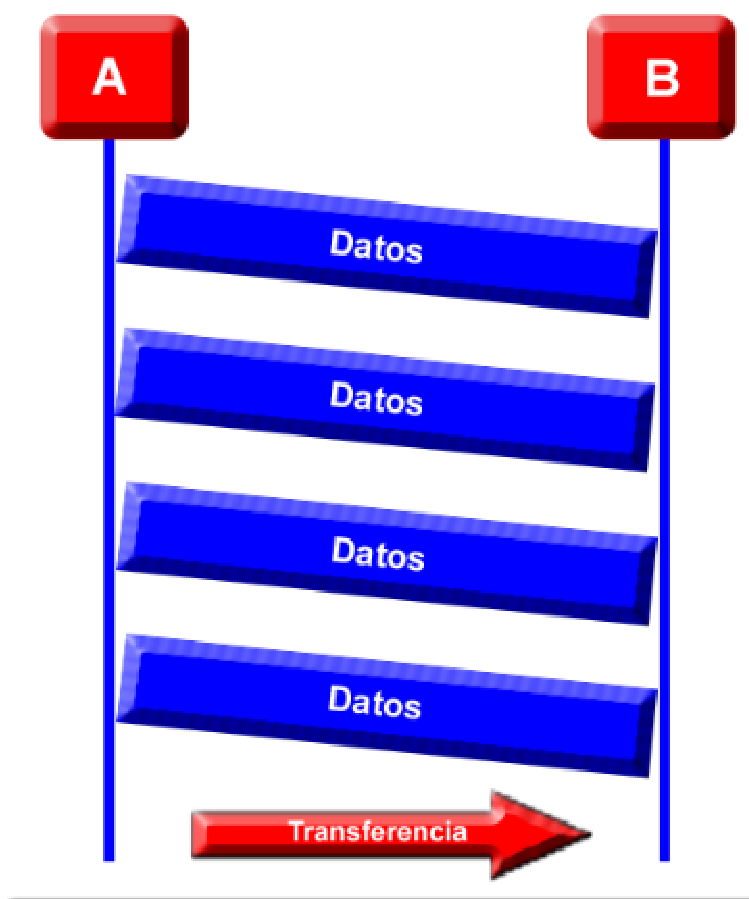


Figura 5.4.2. Esquema de Protocolo Orientado a la no conexión.

5.4.3. Protocolos orientados a Byte

Los protocolos orientados a Byte o carácter utilizan Bytes completos para representar códigos de control establecidos tales como los definidos por el código ASCII (American Standard Code for Information Interchange) o código EBCDIC (Extended Binary Coded Decimal Interchange Code). En contraste, los protocolos orientados a bits confían en bits individuales para códigos de control. Los protocolos orientados a Byte transmiten los datos como si fueran cadenas de caracteres. El método de transmisión es asíncrono. Cada carácter es separado de un BIT de inicio y un BIT de término, y no es necesario un mecanismo de reloj.

5.4.4. Protocolos orientados a Bits

En una transmisión orientada a BIT, los datos son transmitidos como constantes ráfagas de bits. Antes de que la transmisión de datos empiece, caracteres especiales de sincronía son transmitidos por el transmisor, así el receptor puede sincronizarse a sí mismo con la ráfaga de bits. Este patrón de bits es comúnmente representado en una cadena de 8 bits. Por ejemplo, SDLC (Synchronous Data Link Control) de IBM es un protocolo orientado a bits. Su carácter de sincronía (sync) es la cadena de bits 01111110, y esto es seguido por una dirección de 8 bits, un campo de control y por los datos (información útil). Una vez que el sistema receptor recibe esas tramas iniciales, empieza a leer 8 bits a la vez (1 Byte) desde la cadena de bits hasta que aparezca un error o una bandera de término.

6. Bluetooth

6.1. Introducción

Bluetooth es una tecnología de comunicación inalámbrica de corto alcance y bajo consumo de potencia que soporta tanto datos como audio. La idea inicial de la tecnología surge en 1994 con la empresa de telecomunicaciones *Telefonaktiebolaget LM Ericsson* al querer conectar sin cables los teléfonos celulares con sus accesorios. Luego en 1998 se forma el Bluetooth SIG (Grupo de Interés Especial Bluetooth) conformado por cinco compañías promotoras: Ericsson, Intel, IBM, Toshiba y Nokia para impulsar la tecnología y desarrollar la especificación Bluetooth, cuya primera versión (v1.0) fue publicada oficialmente en julio de 1999. La especificación siguiente (v1.1) constaba de dos volúmenes: el volumen 1 o especificación central que contiene la pila de protocolo Bluetooth y el volumen 2 o especificación de perfiles que establece casos de uso para promover interoperabilidad entre las muchas aplicaciones de la pila de protocolo. Debido al nuevo y constante surgimiento de perfiles, hubo una gran reestructuración de la versión 1.2 de la especificación para hacerla más flexible a los constantes cambios de los documentos.

Debido a sus orígenes en Escandinava y a la intención de unir los mundos de la computación y las telecomunicaciones, la tecnología fue nombrada en honor al rey vikingo danés Harald Blatand el cual unió Dinamarca y Noruega alrededor del siglo X.

La tendencia actual hacia los sistemas inalámbricos hace de Bluetooth una tecnología ideal para la conformación de redes inalámbricas, con enlaces seguros, de alta inmunidad al ruido, con una especificación totalmente abierta al público y el uso de la banda ICM libre de licencia, que ayudan a llegar a un estimado bajo costo. Estas características, entre otras, permiten un sin número de aplicaciones aparte del simple reemplazo de un cable.

6.2. Características

Especificaciones:

- El dispositivo en una red PAN comparte el canal de comunicación de datos, este canal tiene una capacidad de 1Mbps. Las cabeceras y el control de flujo consumen alrededor de un 20% de esta capacidad.
- En Estados Unidos y Europa, los rangos de frecuencia usados son de 2.400 a 2.483.5Mhz, con 79 canales de radio frecuencia de 1Mhz, en la práctica el rango es de 2.402Mhz a 2.480Mhz. En Japón el rango de frecuencia es de 2.472 a 2.497Mhz. con 23 canales de 1Mhz.
- El canal de datos salta o varía 1.600 veces por segundo entre 79 (ó 23) canales de radio frecuencia.
- Cada canal esta dividido en espacios de tiempo de 625 microsegundos.
- Una PAN posee un maestro y hasta 7 esclavos. El equipo maestro transmite en tiempos pares y los esclavos en slots de tiempo impares.
- Los paquetes de datos pueden tener hasta 5 veces el tamaño de un slot.
- Un paquete de datos puede tener hasta 2.745 bits de largo.
- Existen dos tipos de transferencia de datos entre dispositivos: SCO (conexión orientada a la sincronización) y ACL (sin conexión sincronizada).

-
- En una PAN, puede existir hasta 3 enlaces SCO de 64.000 bits por segundo cada uno. Para eliminar el problema de las colisiones, el enlace SCO usa slots reservados que el equipo maestro se encarga de distribuir.
 - Un equipo maestro puede soportar hasta 3 enlaces SCO con uno, dos o tres esclavos.
 - Los slots que no están siendo usados para enlaces SCO pueden ser perfectamente usados para enlaces ACL.
 - Un equipo maestro y uno esclavo pueden tener un enlace ACL.
 - Un enlace ACL es también punto a punto (un maestro a un esclavo) o simplemente un maestro a todos los esclavos usando punto a multipunto.
 - Los esclavos que usan enlace ACL pueden transmitir solamente cuando el equipo maestro lo solicita.

6.3. Banda de frecuencia libre

El transmisor está integrado en un pequeño microchip de 9x9 milímetros y opera en una banda de frecuencia abierta a cualquier sistema de radio independientemente del lugar del planeta donde nos encontremos. Sólo la banda ISM (médico–científica internacional) de 2,45 Ghz cumple con éste requisito, con rangos que van de los 2.400 Mhz a los 2.500 Mhz, y solo con algunas restricciones en países como Francia, España y Japón.

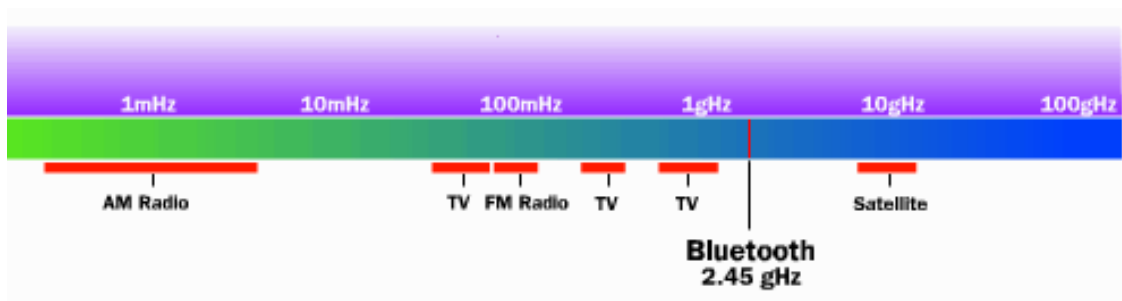


Figura 6.3. Campo de uso de algunas Frecuencias comunes.

6.4. Salto de frecuencia

Debido a que la banda ISM está abierta a cualquiera, el sistema de radio Bluetooth deberá estar preparado para evitar las múltiples interferencias que se pudieran producir. Éstas pueden ser evitadas utilizando un sistema que busque una parte no utilizada del espectro o un sistema de salto de frecuencia. En los sistemas de radio Bluetooth se suele utilizar el método de salto de frecuencia debido a que ésta tecnología puede ser integrada en equipos de baja potencia y bajo coste. Éste sistema divide la banda de frecuencia en varios canales de salto, donde, los transceptores, durante la conexión van cambiando de uno a otro canal

de salto de manera pseudo-aleatoria. Con esto se consigue que el ancho de banda instantáneo sea muy pequeño y también una propagación efectiva sobre el total de ancho de banda. En conclusión, con el sistema FH (Salto de frecuencia), se pueden conseguir transceptores de banda estrecha con una gran inmunidad a las interferencias.

6.5. Seguridad

Para asegurar la protección de la información se ha definido un nivel básico de encriptación, que se ha incluido en el diseño del clip de radio para proveer de seguridad en equipos que carezcan de capacidad de procesamiento, las principales medidas de seguridad son:

- Una rutina de pregunta-respuesta, para autenticación .
- Una corriente cifrada de datos, para encriptación .
- Generación de una clave de sesión (que puede ser cambiada durante la conexión) .

Tres entidades son utilizadas en los algoritmos de seguridad: la dirección de la unidad Bluetooth, que es una entidad pública. Una clave de usuario privada, como una entidad secreta y un número aleatorio, que es diferente por cada nueva transacción.

Como se ha descrito anteriormente, la dirección Bluetooth se puede obtener a través de un procedimiento de consulta. La clave privada se deriva durante la inicialización y no es revelada posteriormente. El número aleatorio se genera en un proceso pseudo-aleatorio en cada unidad Bluetooth.

6.6. Beneficios de Bluetooth

El principal beneficio de esta tecnología es que nos permite conectar dispositivos sin la necesidad de cables, en una PAN (red de área personal) con muchos dispositivos podemos tener un PC con su correspondiente Mouse y teclado, a veces un equipo portátil (notebook), también una PDA con sistema basado Palm o PocketPC y un teléfono móvil con manos libres y un impresora conectada al PC. Como comunicamos estos dispositivos entre si, la respuesta es simple: Usando muchos cables hubs y adaptadores entre ellos.

Esta es solo una parte de todo el proceso, ya que debemos conectar y desconectar cables frecuentemente, sincronizar los dispositivos con el equipo maestro y tratar cada dispositivo individualmente y así suma y sigue este tedioso proceso.

El caso anterior es el de una típica PAN que podemos encontrar en muchas casas y oficinas.

Bluetooth nos plantea un escenario diferente y diseñado para ayudarnos a no perder tiempo repitiendo acciones como es el buscar un cable, conectarlo y sincronizar el dispositivo. Además de permitirnos movilidad en un rango de 10 metros, al no poseer una distancia fija como es el caso de la conexión con cable.

La figura siguiente nos presenta un diagrama de un escenario Bluetooth en el que los dispositivos simplemente se interconectan entre si usando esta tecnología en este caso en modo ad-hoc y además la interconexión entre el equipo central y las demás redes como Internet, esto nos permite prescindir de los cables, eliminando así el problema de las conexiones físicas en el cual el usuario no interviene en nada (previa configuración de software en el equipo maestro y en el dispositivo).

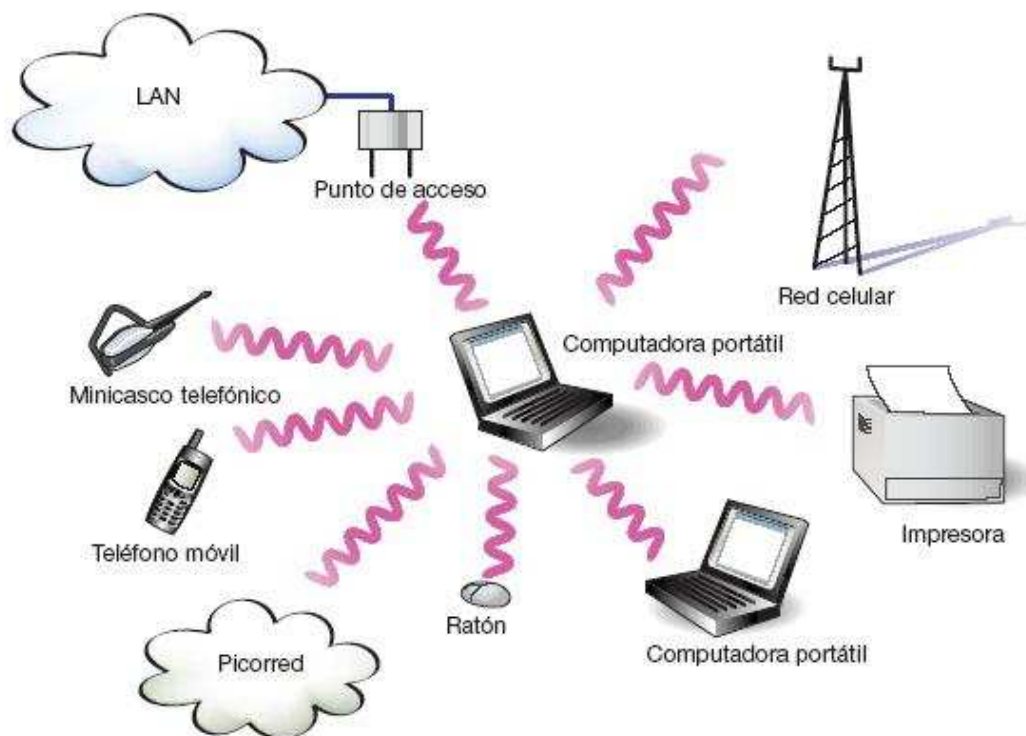


Figura 6.6.1. Mapa de interconexión de dispositivos.

Este escenario sin cables es posible gracias a la forma de trabajo Maestro / esclavo de la tecnología Bluetooth. Cada dispositivo es un punto o par el cual es identificado con una dirección única de 48 bits, también puede ser asignada como maestro a través de la configuración del usuario.

Un dispositivo en modo maestro puede conectarse hasta con 7 dispositivos en modo esclavo al mismo tiempo, un número que nos permite trabajar holgadamente. Formando así una red “Punto a Multipunto”.

Si extraemos algunas conclusiones, podemos encontrar que la conectividad inalámbrica entre dispositivos nos ofrece una gran libertad de movimiento y conveniencia, esto nos permite ejecutar tediosas tareas con una intervención mínima, además con la posibilidad de que los dispositivos tengan la opción de trabajar en modo dual (maestro y esclavo) nos ayuda a ahorrar muchos cables.

Bluetooth nos hará cambiar la forma de ver la conectividad entre dispositivos electrónicos, la idea de tener un cable especial para conectar un dispositivo con otro va a quedar en el pasado.

Con la enorme cantidad de dispositivos que existe hoy con esta habilidad de incorporar la tecnología Bluetooth estamos cada vez más cerca al ejemplo anterior donde todos estos aparatos electrónicos se conectaban sin cables.

6.7. Otros escenarios usando Bluetooth

Para una red en nuestro hogar hay un par de ideas. La primera de ellas cuenta con un dispositivo maestro que controla todo en la red, desde el video, televisor, cámaras de seguridad, etc. Y este reemplaza al PC, pasando a tomar posición de concentrador de todos los dispositivos.

El otro escenario es uno similar en el que el PC es el dispositivo maestro y a través de un software puede controlar todos los dispositivos.

La figura siguiente nos muestra como esta PAN puede ser extendida en nuestros hogares y combinada con la estructura de cables para así formar una HAN (red de área de hogar) la que utiliza la tecnología inalámbrica para el control de los dispositivos audiovisuales. En estos momentos en algunos países de Europa y Estados Unidos ya se esta promoviendo fuertemente el uso de la casa

inalámbrica la que nos demuestra varios dispositivos que usualmente usamos a diario conectados con cables como es el televisor y el video, esta vez usando Bluetooth.

6.8. Interferencia en las redes Bluetooth

Las conexiones inalámbricas están sujetas a interferencias, la interferencia puede tener un importante impacto en la calidad de la conexión de audio y en la transferencia de datos. Altos niveles de interferencia pueden interrumpir la comunicación por un tiempo suficiente para que la pila del protocolo llegue a time out y se termine la conexión estable que existía.

El fabricante y desarrollador del estándar Bluetooth explica que esta tecnología no debe de ser usada para los casos críticos en los que los datos deben llegar si o si, ya que siempre esta la posibilidad de que una interferencia detenga el enlace constante que se establece para la transferencia de datos, la interferencia puede venir de muchas fuentes: micro ondas, tormentas eléctricas y otros sistemas de comunicación (como es el caso de la IEEE 802.11b) o también otros dispositivos Bluetooth(estos no tienen un gran efecto por que están diseñados para responder para estas situaciones).

6.9. Consideraciones en las conexiones

6.9.1. Trabajo con anchos de banda limitados

En muchos casos, inalámbrico significa “lento”, una conexión a Internet vía Bluetooth esta limitada a un máximo teórico de transferencia de 723.2Kbps. El control de tráfico, los protocolos y datos adjuntos hacen que este máximo se vea muy disminuido, dado esto a Bluetooth no se le puede considerar como una conexión inalámbrica de alta velocidad, como para enviar y recibir grandes cantidades de datos, este no es el método correcto para hacerlo.

Esto tiene un impacto importante en la calidad de audio, la tecnología Bluetooth simplemente no posee el ancho de banda para trabajar con la compresión RAW CD de calidad de sonido a 1411.2Kbps, sin embargo si es manejable la técnica de compresión que es empleada en el formato MP3.

6.9.2. Consumo de energía y alcance

El consumo de energía es un factor crítico para los dispositivos inalámbricos. Si un producto se hace llamar inalámbrico es obvio que no esta conectado a la toma de corriente y la pregunta, de donde viene la energía? De una batería, es por eso que el consumo energético del transmisor Bluetooth incorporado en este dispositivo debe usar un mínimo de recursos energéticos. Acá es donde se considera la duración de la batería, el tiempo de suspensión y las dimensiones físicas también.

Por ejemplo: los audífonos o manos libres no necesitan abastecimiento de energía cuando están conectados con cables. Las señales de audio simplemente vienen en el cable. Un sistema tan simple como este que no necesita ser energizado con conectores extras.

Cuando es reemplazado por un enlace Bluetooth, comienza a ser necesaria una fuente de energía para establecer el enlace, también para que el chip controle el protocolo Bluetooth y energía para amplificar la señal de audio a un nivel necesario para que el usuario pueda escucharla claramente. Con dispositivos móviles pequeños lo que menos se desea es usar grandes baterías, es por ello lo importante de mantener el consumo de energía a un nivel mínimo.

6.9.2.1. Manejando un alcance aceptable

La especificación del protocolo Bluetooth define 3 tipos de estados de entrega de energía para transmisores con una salida de 1mW, 2.5 mW y 100mW.

La salida de energía define el alcance en metros para este dispositivo, es por ello que se debe considerar el uso que le queremos dar al dispositivo antes de adquirirlo.

6.9.2.2. Máxima salida de poder versus el alcance

1. Clase 1 a 100mW y 100 Metros aprox.
2. Clase 2 a 2.5mW y 10 Metros aprox.
3. Clase 3 a 1mW y 1 Metro aprox.

Es importante que estos rangos son considerando un uso típico de hogar, oficina, etc.

Cuando se esta en un espacio abierto se puede alcanzar mucho mas distancia si no existen objetos que se interpongan o absorban la señal.

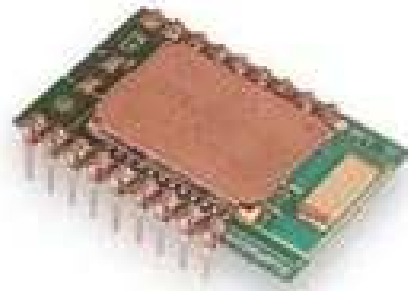


Figura 6.9.2.2.1. Diseño de PCB modulo clase 1.



Figura 6.9.2.2.2. Diseño de PCB modulo clase 2.

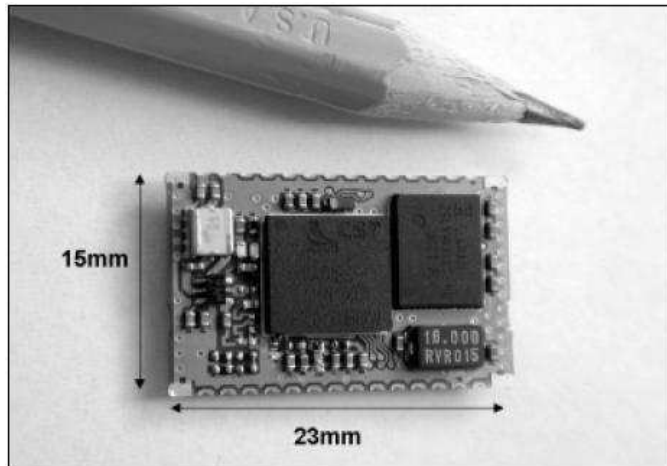


Figura 6.9.2.2.3. Modulo de Clase 2.

6.10. Dispositivos y adaptadores basados en Bluetooth

Ya están en el mercado muchos dispositivos que usan la tecnología Bluetooth, como son: impresoras, teclados, Mouse, scanner, PDA, equipos celulares, GPS, etc.

Este dispositivo permite imprimir documentos en nuestra impresora USB habitual sin necesidad de cables de conexión. Este adaptador nos permite imprimir documentos vía Bluetooth a una distancia de hasta 100 metros en condiciones óptimas. Si la impresora esta apagada en el momento de la impresión, el adaptador se encarga de enviar la señal para encenderla.



Figura 6.10.1. Adaptador Bluetooth para dispositivos con USB.

El dispositivo que va conectado al equipo central o PC es un adaptador USB a Bluetooth, a continuación una fotografía del pequeño adaptador de solo 3cm de tamaño que nos permite un alcance de 10 metros aprox. Ya que posee un transmisor de Clase 2 de 2.5mW de potencia.



Figura 6.10.2. Adaptador Bluetooth para PC o Laptops.

Un ejemplo de una PDA con su auricular inalámbrico que además permite controlar algunas funciones de la PDA a través del auricular.



Figura 6.10.3. PDA con Auricular Bluetooth.

7. Comunicación entre dispositivos

7.1. Introducción

La temática del trabajo práctico aborda la conectividad entre varios dispositivos, la forma de interactuar entre ellos. La idea es demostrar que no importa que tipo de dispositivo se use, con tal de que posea interfaces que nos permitan un flujo de datos, un método de programación y un equipo que actúe como centro de control de datos. Esto hace posible que podamos enviar y recibir datos en cualquier dirección y así sacarle provecho a esta mezcla de tecnologías. En este caso se trabajó con microcontroladores PIC, servo motores, transmisores y receptores inalámbricos de 433 Mhz. un PC, con adaptadores USB a Bluetooth y equipos celulares.

7.2. Esquema Principal

Este diagrama nos muestra el flujo de datos entre los diferentes dispositivos que participan en este sistema, el equipo central posee la tarea de sincronizar la comunicación entre los extremos, conectándose a la base de datos cuando sea necesario, esta base de datos puede ser local o remota.

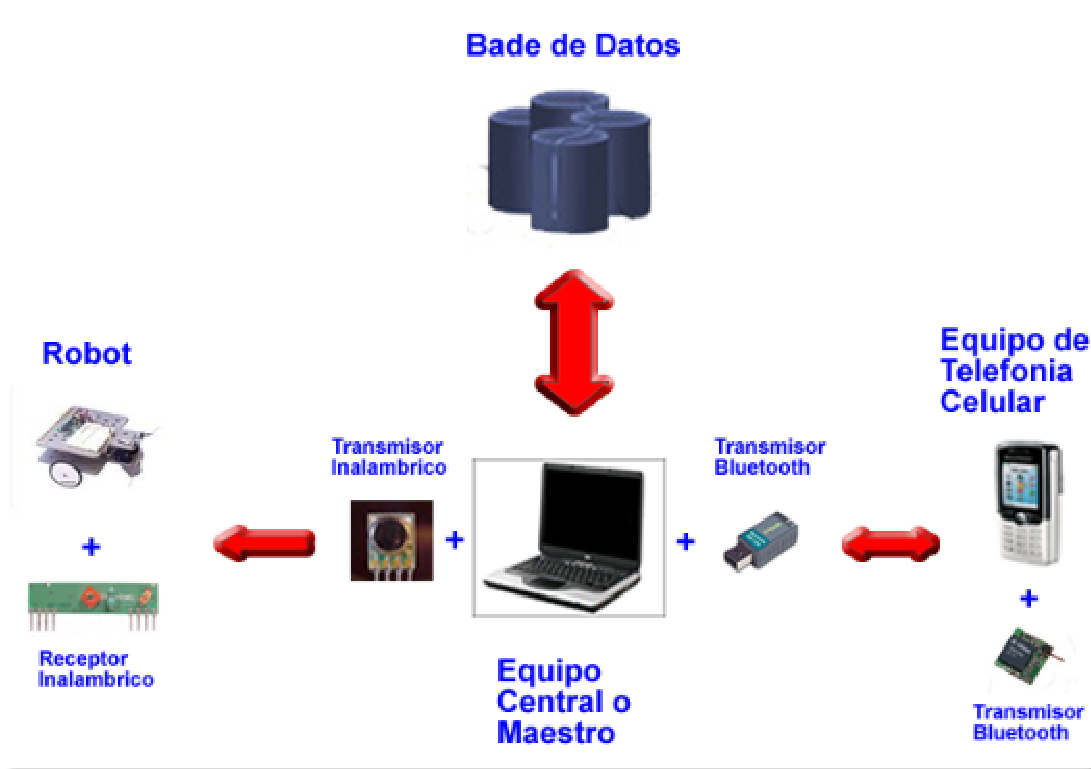


Figura 7.2. Diagrama de flujo de información.

El software que se ejecuta en el equipo central está desarrollado en Borland Delphi 6, el software es el encargado de la comunicación entre todos los dispositivos.

7.3. Transmisores

Las interfaces que participan en este sistema son el puerto serial del PC, que tiene conectado un transmisor inalámbrico de 433Mhz que se encarga de enviar el dato inalámbrica menté hasta el receptor que esta en el robot.



Figura 7.3.1. Transmisor inalámbrico de 433Mhz.

En el otro extremo el robot tiene un receptor de 433Mhz. que recibe datos que vienen desde el transmisor que tiene conectado el PC.



Figura 7.3.2. Receptor Inalámbrico de 433Mhz.

7.4. Desarrollo

El robot esta siendo controlado por el chip PIC 16F84A que funciona con 5Volts, este se encarga de tomar los datos que le entrega el receptor inalámbrico, analizar la trama de datos que llega y ver si esta completa para posteriormente ejecutar las ordenes que correspondan.

El primer paso era poder controlar el robot usando el equipo central. Este es un robot con funciones básicas, hablamos de movimientos básicos como son los movimientos similares a los de un auto de juguete, ya que la idea central del proyecto es probar la conectividad entre distintos dispositivos, no el hacer un gran robot que tengan muchas funciones.



Figura 7.4. Aplicación de Control en el Equipo Central.

De esta forma se logra enviar datos desde el PC al robot y este los ejecuta en forma casi instantánea.

7.5. Transmisión y Trama de Datos

Anteriormente se hablo de una trama de datos, la explicación para ellos es la siguiente: el par de transmisor y receptor de 433Mhz que se uso en este proyecto, son equipos muy básicos y económicos. No traen buffer (memoria para almacenar datos temporalmente), esto significa que la comunicación entre el PC y el robot tiene que ser con absoluta sincronización y precisión, ya que en el momento de salir un dato del PC hacia el robot, si el robot no estaba preparado para recepción, simplemente se pierde el dato.

1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
<i>Activación de canal</i>	<i>Sincronización</i>	<i>Dato de tipo de Acción</i>	<i>Dato de Duración de Acción</i>	<i>Checksum</i>

Figura 7.5. Trama de datos enviada desde el PC al Robot.

En el momento en que se le quiere enviar una orden al robot, se debe enviar la trama que se muestra en el cuadro anterior, el robot esta esperando un Byte de activación de canal, que en el momento de recibirlo, se activa una rutina en el microcontrolador del robot que espera el Byte siguiente que es el de sincronización, luego recibe los 3 Bytes siguientes que contienen datos y checksum de la trama. Cada dato que se envía desde el PC al robot, debe ir con un retardo entre 30ms y 15ms entre Bytes, 30ms es el máximo ya que con más ms de retardo se comienza a pierden datos y basado en pruebas técnicas de distancia, he comprobado que 15ms es el mínimo y con el cual se logra más alcance con este par de transmisor y receptor inalámbrico.

Una vez recibida toda la trama, el microcontrolador descompone la trama y la analiza, suma dato a dato y este resultado se compara con el Byte de checksum, si los datos llegaron bien y el proceso de comprobación de checksum fue satisfactorio, el robot procede a ejecutar la acción.

Sino llegaron bien los datos, la comprobación de checksum no debería ser satisfactoria y por ende el robot vuelve al estado de espera de otra trama completa. También posee un TimeOut en el caso de que quede esperando un dato que no va a llegar.

Dado aquello el tema de la sincronización y el evitar perder datos obligo a crear esta trama de datos para que el robot pudiera analizar antes de ejecutar y así ver si todos los datos llegaron bien antes de empezar a interpretarlos. Con esto evitamos que el robot quede esperando datos que no van a llegar o que ya se enviaron y simplemente se perdieron al no estar listo para recibirlos.

7.6. Equipo Celular y Bluetooth

Por otro lado el equipo central de control (PC) se conecta con un equipo de telefonía celular mediante la tecnología inalámbrica Bluetooth.

Como la mayoría de los PC que hay en el mercado no traen la tecnología Bluetooth incorporada como estándar, se debió usar un adaptador para integrar esta habilidad al PC, se uso un adaptador USB a Bluetooth marca D-Link modelo DBT-120. Este adaptador nos permite conectarnos al equipo celular que también es capaz de manejar esta tecnología. Gracias a esto podemos crear un canal estable de transmisión de datos entre el PC y el celular usando la tecnología inalámbrica Bluetooth.



Figura 7.6. Adaptador USB a Bluetooth.

Como se explico anteriormente, los transmisores Bluetooth poseen buffer, control de flujo y todas las herramientas necesarias para que cuando tengamos el enlace de datos funcionando, no tengamos que preocuparnos si el dato llega o no por que la capa inferiores de Bluetooth se preocupan de la entrega segura de información.

Como segundo paso la idea era transmitir datos entre el PC y el equipo celular, lo que se logro investigando sobre los tipos de programación que era capas de manejar el modelo de celular que usamos, el SonyEricsson T616, el cual trae soporte para JavaMicroEdition, más conocido como JME2, además soporte para aplicaciones basadas en Morpheus. Este último se usa para programar juegos básicamente. Y un tercer método que fue el que finalmente usamos para poder acceder a todas las funciones de este equipo que es el de la programación a nivel del hardware del celular (Comandos AT).

Usando comandos AT que soporta este equipo celular, podemos acceder a absolutamente todas las funciones que existen en el teléfono móvil.

Por que elegir el tercer método de programación, la respuesta es simple.

Por que la versión de JME2 que tiene el SonyEriccson T616 no soporta Bluetooth y Morpheus tampoco, así que era la única alternativa y resulto ser la mejor.

Como funciona?. Se establece un enlace de comunicación de datos entre el celular y el PC emulando un puerto serial y usando velocidades que van desde los 300bps hasta los 115200bps. Entre más alta la velocidad la comunicación es más fluida y el tiempo de respuesta es mucho más rápido pero consume más batería el equipo celular al usar altas velocidades.

La forma de trabajo fue enviar desde el PC al celular comandos AT mencionados anteriormente y el celular los ejecuta instantáneamente. Esto nos permite crear menús, diálogos de entrada de datos, mensajes, advertencias, controlar el teclado del celular, usar funciones multimedia, enviar datos hacia y desde el celular, etc.

7.7. Base de Datos

Para probar este flujo de datos se creo una base de datos en el PC usando el motor de base de datos que trae Borland Delphi 6 integrado, se uso formato Dbase para Windows para la tabla de datos, esta tabla posee 4 campos: Rut, Nombre, Deuda, Teléfono.

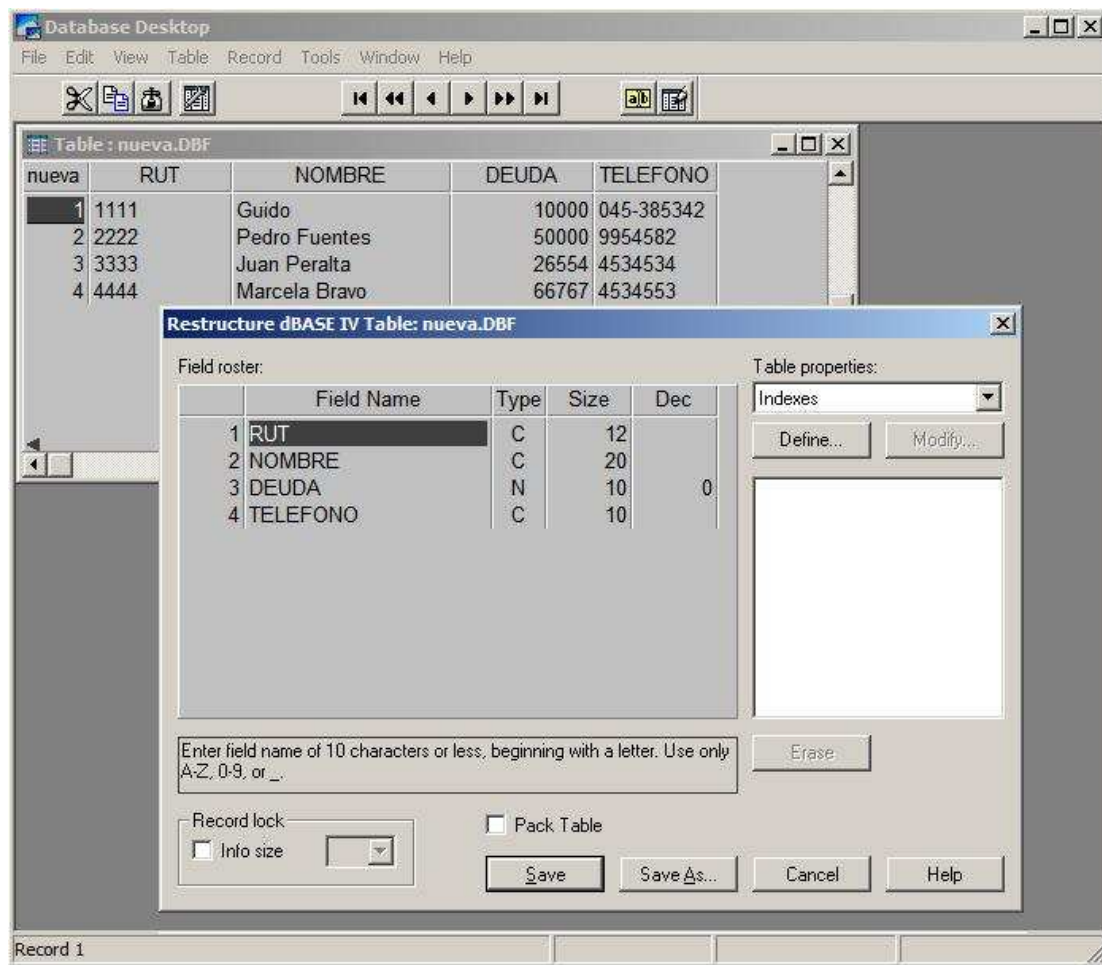


Figura 7.7.1. Motor de Bases de Datos de Borland Delphi 6.

En la imagen anterior se muestra el editor de bases de datos que trae Borland Delphi 6 integrado. Para el tema de la velocidad en el desarrollo y pruebas se uso el campo Rut como carácter además de datos simulados que nos permitieran probar rápidamente el avance.

Luego de establecer el enlace entre el celular y el PC, es cuando el software que esta ejecutándose en el PC comienza a enviar los comandos

necesarios para crear una interfaz en el celular que nos permita interactuar en la recepción y envío de datos, dado aquello se puede acceder desde el celular a la base de datos que esta en el PC y hacer consultas sobre la tabla que contiene los campos anteriormente mencionados.

Diálogo de consulta en la Base de Datos usando el campo Rut para búsqueda de Coincidencias.



Figura 7.7.2. Diálogo de ingreso de Consulta por RUT en el equipo Celular.

Luego de ejecutar la consulta, el PC busca en la base de datos y envía la información al Celular que la muestra en pantalla. La imagen siguiente nos ilustra la respuesta sobre la búsqueda hecha sobre la base de datos que esta en el PC.



Figura 7.7.3. Respuesta generada por la consulta sobre la Base de Datos.

Luego como tercer pasó y último se intento controlar el robot satisfactoriamente desde el equipo celular, usando Bluetooth entre el celular y el PC, luego entre el PC y el robot una red inalámbrica con transmisores de 433Mhz.

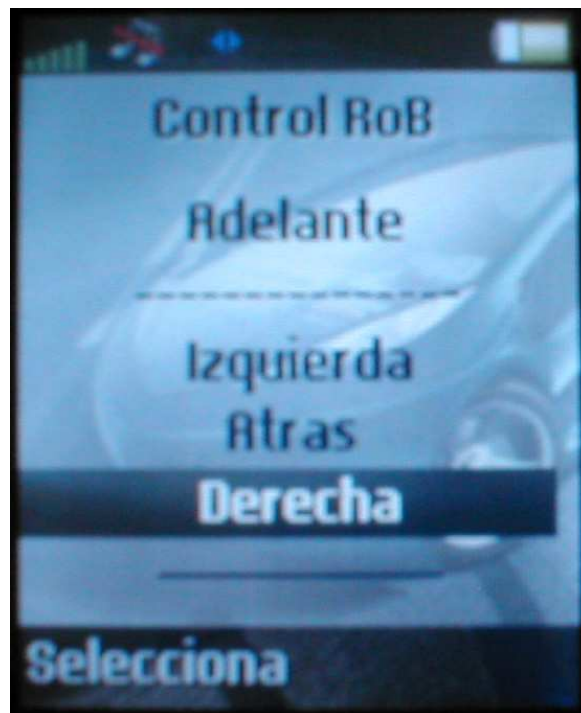


Figura 7.7.4. Menú de control de Robot en el Equipo celular.

El equipo central de control (PC) es el que se lleva toda la tarea complicada de sincronizar, enviar y recibir datos por diferentes puertos, realizar acciones sobre la base de datos y además crear y administrar toda la interfase que esta en el celular. El equipo central es el nexa entre los dispositivos que queremos comunicar.

Estando todo conectado y encendido el robot, el usuario solo debe usar el celular, navegar por los menús y con el controlar el robot o ver la base de datos como si fuera un terminal cualquiera.

Queda probado el flujo de datos entre el celular y el robot pasando por el PC a una velocidad de 115200bps entre el celular y el PC, entre el PC y el robot a

1200bps, para el tipo y cantidad de datos que estamos manejando el tiempo de respuesta es casi despreciable, lo que hace este un sistema muy atractivo para seguir investigando y desarrollando sobre la base estudiada.

8. Códigos Fuentes

8.1. Códigos en el Microcontrolador

8.1.1. Explicación de Códigos

Este es el código que va en el microcontrolador, esta escrito en Microcode Studio Plus y luego compilado en EPICtm Plus PICmicro MCU Programmer 2.42, este mismo compilador se encarga de bajar el código al microcontrolador, por medio de un quemador.

A continuación la declaración de variables y su correspondiente explicación de uso.

```
include "modedefs.bas"    ' definiciones de MODOS Serin/out, Shiftin/out, etc.'
B var byte                ' variable que almacena la dirección del movimiento'
i var byte                ' variable usada para ciclo FOR'
Arreglo var byte[3]      ' arreglo que almacena la trama que viene desde el PC '
Total var word           ' variable que almacena suma de instrucciones de Checksum'
```

La instrucción que se muestra a continuación es la encargada de recibir el dato por la interfaz inalámbrica que posee el robot, esta lee byte a byte una cadena de datos desde el Puerto **A.1**, a 1200 Baudios con las siguientes características: Bit15 Driven, Bit14 Inverted y Bit13 Paridad None.

```
serin2 PORTA.1, 17197, [wait (1),str arreglo\3]
```

Otras instrucciones importantes en este código son las que especifican los movimientos de los motores que tiene conectado el robot, estas instrucciones son representadas por movimientos básicos como son: Adelante, Atrás, Izquierda, Derecha.

Esta instrucción **PulsOut** específicamente saca por el Puerto B.2 un pulso a una frecuencia de 100 hertz. Esto hace que el motor conectado a la Puerta B.2 se mueva en una determinada dirección.

PulsOut PORTB.2,100

8.1.2. Código Completo

Función principal de este código es recibir datos por el puerto de entrada A.1, que tiene conectado un receptor inalámbrico que trabaja a 433Mhz. Este código espera la llegada de un dato por el puerto de entrada A.1, analiza la trama y comienza a interpretarla.

La información que contiene la trama es básicamente de movimiento y control, lo que se ve interpretado en las sentencias case, las cuales son 4 y representan los movimientos básicos del robot.

```
include "modedefs.bas"
B var byte
i var byte
Arreglo var byte[3]
Total var word

loop:
  arreglo[0] = 0
  arreglo[1] = 0
  arreglo[2] = 0
  arreglo[3] = 0
  i = 0
  total = 0
  serin2 PORTA.1, 17197, [wait (1),str arreglo\3]
  B = arreglo[1]
  for i = 0 to 1
```

```
    total = total + arr[i]
next i
if ( arreglo[2] = total ) then
  Select case b
    case 1
      For i = 1 To 48
        PulsOut PORTB.2,100
        PulsOut PORTB.5,100
        Pause 5
      Next i
      goto loop
    case 2
      For i = 1 To 130
        PulsOut PORTB.2,100
        PulsOut PORTB.5,200
        Pause 5
      Next i
      goto loop
    case 3
      For i = 1 To 130
        PulsOut PORTB.2,200
        PulsOut PORTB.5,100
        Pause 5
      Next i
      goto loop
    case 4
      For i = 1 To 38
        PulsOut PORTB.2,200
        PulsOut PORTB.5,200
        Pause 5
      Next i
      goto loop
  end select
endif
goto loop
end
```

8.2. Código del Equipo Servidor

8.2.1. Explicación de Códigos

A continuación se explica parte del código que se ejecuta en el equipo servidor, este código esta escrito en Borland Delphi 6.

Este lenguaje esta orientado a eventos, se explicara lo que hace cada evento y los procedimientos principales.

El código comienza con la inicialización del menú accesorios en el equipo celular, esto inicia la comunicación con el celular, esta comunicación es de enviar una orden y esperar una respuesta para luego enviar otra orden y esto se lleva a cabo en el siguiente código:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    comport1.WriteString('AT*EAM="Robot Bakan" + #13);  
end;
```

El evento **ComPort1RxFlag** es generado gracias a la respuesta desde el celular hacia el equipo servidor, el siguiente procedimiento lee los datos que llegan por el COM 1 y reconoce si el usuario desea inicializar la función de control del Robot o la función de Consulta de datos sobre una base de datos en SQL.

```
procedure TForm1.ComPort1RxFlag(Sender: TObject);  
begin  
    a:="";  
    b:="";  
    c:="";  
    aux:=comport1.ReadString;  
    a := copy(aux,1,13);      'Extrae parte de la cadena que lee del puerto COM 1'  
    b := copy(aux,1,11);
```

```

c := copy(aux,1,12);
edit1.Text:=aux;
if checkbox1.Checked = true then
    controlrobot           'llama al procedimiento control robot'
else
    if checkbox1.Checked = false then
        basededatos;      'llama al procedimiento base de datos'

end;
```

El procedimiento **controlrobot** es el encargado de seguir con la creación del submenú accesorios si la respuesta de la creación del menú principal es la correcta, esto se lleva a cabo con la siguiente orden, la cual detalla los ítems que van en la pantalla del celular.

```

if aux = (#13#10 + '*EAAI' + #13#10) then
    form1.comport1.WriteString ('AT*EASM="Control      RoB",1,2,6,"-----
    ", "Adelante", "-----", "Izquierda", "Atras", "Derecha ", 1' + #13);
```

Después de creado el submenú con las opciones de movimiento del robot, el usuario en el celular debe elegir alguna de ella, si la segunda opción es seleccionada se llama al procedimiento **DireccionRobot** con el parámetro direc, que contiene un valor que se interpretara en el procedimiento a seguir.

```

if a = (#13#10 + '*EAMI: 2' + #13#10) then
    DireccionRobot(direc);
```

El procedimiento **DireccionRobot** se encarga de asigna un valor al arreglo que contiene en la posición numero 2 el dato de la dirección, luego asigna a la posición 3 la suma de la dirección y un dato auxiliar para poder armar el cheksum,

después llama al procedimiento **MovimientoRobot** el cual envía al robot las ordenes para que el robot ejecute las acciones.

```
case movi of
    2: arr[2]:=2;
    4: arr[2]:=1;
    5: arr[2]:=3;
    6: arr[2]:=4;
end;

arr[3] := arr[1] + arr[2];
MovimientoRobot;
```

El procedimiento **MovimientoRobot** es el encargado de enviar la trama de datos hacia el robot por medio del puerto que tiene conectado el transmisor inalámbrico, se debe enviar un Byte de calentamiento o activación (hot) de canal, luego esperar 15ms para no sobrecargar el canal de comunicación y así seguir con el proceso de enviar Bytes y esperar 15ms entre cada dato.

```
repeat
    if ( x = 1 ) then
        begin
            form1.comport2.WriteByte (Hot);
            sleep(15);
        end;
    if ( x = 2 ) then
        begin
            form1.comport2.WriteByte(Sync);
            sleep(15);
        end;
    if ( x > 2 ) then
        begin
            form1.comport2.WriteByte(arr[j]);
            j := j + 1;
            sleep(15);
        end;
    x := x + 1;
```

```
until j = 4;
```

El procedimiento **BasedeDatos** es el encargado de controlar todo lo que tiene que ver con la consulta de datos en el equipo servidor, las siguientes línea muestran la creación estándar del submenú de la consulta.

```
if aux = (#13#10 + '*EAAI' + #13#10) then
    form1.comport1.WriteString('AT*EASM="Base de DatoS",1,1,1,"Consulta", 1'+
#13);
```

La creación del menú especial de la consulta de datos por RUT se lleva a cabo con el siguiente comando:

```
if a = (#13#10 + '*EAMI: 1' + #13#10) then
    begin
        form1.comport1.WriteString('AT*EAID=11,0,"Consulta","Ingrese
RUT",12,"" + #13);
    end;
```

Una vez ingresado el Rut a buscar, el celular envía la el dato y el equipo servidor comienza a hacer la consulta SQL a la base de datos, si no encuentra datos asociados a ese Rut entrega un aviso de que no existen registro, si encuentra valor manda el string con todos los datos que encontró como se ve en la sentencia **else**.

```
if ( c = (#13#10 + '*EAI: 11,') ) and ( aux <> (#13#10 + '*EAI: 11,0') ) then
    begin
        i:=length(aux);
        Valor:=copy(aux,14,i-16);
        form1.edit2.Text:=Valor;
```

```

form1.Query1.Close;
form1.Query1.SQL.Clear;
form1.Query1.SQL.Add ('Select rut, nombre, deuda, telefono');
form1.Query1.SQL.Add ('FROM nueva.dbf ');
form1.Query1.SQL.Add ('WHERE rut LIKE "' + Valor + '%"');
form1.Query1.Open;

if form1.DBText1.Caption = " then
    form1.comport1.WriteString('AT*EAID=1,0,"No existe Registro
con ese RUT",20' + #13)
else
    form1.comport1.WriteString('AT*EAID=15,1,"Base de
DatoS","Datos Acualizados",4,"Rut:"' + form1.DBText1.Caption +',"Nom:"' +
form1.DBText2.Caption +',"Deuda:"' + form1.DBText3.Caption +',"Fono:"' +
form1.DBText4.Caption +',"1' + #13);
end;

```

8.2.2. Código Completo

La función principal de este código es controlar el equipo celular, recibir datos desde el y interpretarlos, para ello cuenta con varios eventos, uno de ellos es el ComPort1RxFlag que esta encargado de recibir datos y enviarlos a los distintos procedimientos según sea el tipo de dato, este código también esta encargado de enviar datos al robot por la internas inalámbrica y además hacer consultas SQL sobre la base de datos y luego enviar la respuesta al celular.

```

var
Form1      : TForm1;
Aux        : string;
A          : string;
B          : string;
C          : string;
D          : string;
Valor      : string;
I          : integer;

direc      : integer;

```

```
j      : byte;
x      : byte;
Hot    : byte;
Sync   : byte;
Arr    : array [1..3] of byte;
```

implementation

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    comport1.WriteString('AT*EAM="Robot Bakan" + #13);
end;
```

```
procedure MovimientoRobot;
begin
    j:=1;
    x:=1;
    Hot:=23;
    Sync:=1;

    repeat
        if (x = 1) then
            begin
                form1.comport2.WriteByte(Hot);
                sleep(15);
            end;
        if (x = 2) then
            begin
                form1.comport2.WriteByte(Sync);
                sleep(15);
            end;
        if (x > 2) then
            begin
                form1.comport2.WriteByte(arr[j]);
                j:=j + 1;
                sleep(15);
            end;
        x:= x + 1;
    until j = 4;

end;
```

```
procedure DireccionRobot(var movi:integer);
```

```
begin
```

```
    arr[1]:=1;
```

```
    case movi of
```

```
        2: arr[2]:=2;
```

```
        4: arr[2]:=1;
```

```
        5: arr[2]:=3;
```

```
        6: arr[2]:=4;
```

```
    end;
```

```
    arr[3]:=arr[1]+arr[2];
```

```
    MovimientoRobot;
```

```
end;
```

```
procedure controlrobot;
```

```
begin
```

```
    if aux = (#13#10 + '*EAAI' + #13#10) then
```

```
        form1.comport1.WriteString('AT*EASM="Control RoB",1,2,6," -----  
-----  ", "      Adelante      ", " -----      ", "      Izquierda      ", "  
Atras    ", "      Derecha      ", '1' + #13);
```

```
    if a = (#13#10 + '*EAMI: 1' + #13#10) then
```

```
        begin
```

```
            form1.comport1.WriteString('AT*EASM="Control RoB",1,2,6," ----  
-----  ", "      Adelante      ", " -----      ", "      Izquierda      ", "  
Atras    ", "      Derecha      ", '1' + #13);  
            form1.edit2.Text:='nada';
```

```
        end;
```

```
    if a = (#13#10 + '*EAMI: 2' + #13#10) then
```

```
        begin
```

```
            form1.comport1.WriteString('AT*EASM="Control RoB",1,2,6," ----  
-----  ", "      Adelante      ", " -----      ", "      Izquierda      ", "  
Atras    ", "      Derecha      ", '1' + #13);  
            i:=Length(aux);  
            a:=copy(aux,i-2,1);  
            form1.edit2.Text:=a;  
            direc:=strtoint(a);  
            DireccionRobot(direc);
```

```

end;

if a = (#13#10 + '*EAMI: 3' + #13#10) then
begin
form1.comport1.WriteString('AT*EASM="Control RoB",1,2,6," ----
----- "," Adelante "," ----- "," Izquierda ","
Atras "," Derecha ",1' + #13);
form1.edit2.Text:='nada';
end;

if a = (#13#10 + '*EAMI: 4' + #13#10) then
begin
form1.comport1.WriteString('AT*EASM="Control RoB",1,4,6," ----
----- "," Adelante "," ----- "," Izquierda ","
Atras "," Derecha ",1' + #13);
i:=Length(aux);
a:=copy(aux,i-2,1);
form1.edit2.Text:=a;
direc:=strtoint(a);
DireccionRobot(direc);
end;

if a = (#13#10 + '*EAMI: 5' + #13#10) then
begin
form1.comport1.WriteString('AT*EASM="Control RoB",1,5,6," ----
----- "," Adelante "," ----- "," Izquierda ","
Atras "," Derecha ",1' + #13);
i:=Length(aux);
a:=copy(aux,i-2,1);
form1.edit2.Text:=a;
direc:=strtoint(a);
DireccionRobot(direc);
end;

if a = (#13#10 + '*EAMI: 6' + #13#10) then
begin
form1.comport1.WriteString('AT*EASM="Control RoB",1,6,6," ----
----- "," Adelante "," ----- "," Izquierda ","
Atras "," Derecha ",1' + #13);
i:=Length(aux);
a:=copy(aux,i-2,1);
form1.edit2.Text:=a;
direc:=strtoint(a);

```

```

        DireccionRobot(direc);
    end;
end;

procedure basededatos;
begin

    if aux = (#13#10 + '*EAAI' + #13#10) then
        form1.comport1.WriteString('AT*EASM="Base de
DatoS",1,1,1,"Consulta",1' + #13);

    if a = (#13#10 + '*EAMI: 1' + #13#10) then
        begin
            form1.comport1.WriteString('AT*EAID=11,0,"Consulta","Ingrese
RUT",12,"" + #13);
        end;

    if (c = (#13#10 + '*EAI: 11,') and (aux <> (#13#10 + '*EAI: 11,0')) then
        begin
            i:=length(aux);
            Valor:=copy(aux,14,i-16);
            form1.edit2.Text:=Valor;
            form1.Query1.Close;
            form1.Query1.SQL.Clear;
            form1.Query1.SQL.Add ('Select rut, nombre, deuda, telefono');
            form1.Query1.SQL.Add ('FROM nueva.dbf ');
            form1.Query1.SQL.Add ('WHERE rut LIKE "' + Valor + '%"');
            form1.Query1.Open;
            if form1.DBText1.Caption = " then
                form1.comport1.WriteString('AT*EAID=1,0,"No existe Registro
con ese RUT",20' + #13)
            else
                form1.comport1.WriteString('AT*EAID=15,1,"Base de
DatoS","Datos Acualizados",4,"Rut:"" + form1.DBText1.Caption + ","Nom:"" +
form1.DBText2.Caption + ","Deuda:"" + form1.DBText3.Caption + ","Fono:"" +
form1.DBText4.Caption + ""',1' + #13);
            end;

    if c = (#13#10 + '*EAI: 16,') then
        begin
            form1.comport1.WriteString('AT*EASM="Base de
DatoS",1,1,1,"Consulta",1' + #13);

```

```
end;

if a = (#13#10 + '*EAI: 0' + #13#10) then
begin
form1.comport1.WriteString('AT*EASM="Base
DatoS",1,1,1,"Consulta",1' + #13);
end;
end;
```

```
procedure TForm1.ComPort1RxFlag(Sender: TObject);
begin
```

```
  a:="";
  b:="";
  c:="";
  Valor:="";
  i:=0;
  aux:=comport1.ReadString;
  a:=copy(aux,1,13);
  b:=copy(aux,1,11);
  c:=copy(aux,1,12);
  edit1.Text:=aux;
  if checkbox1.Checked = true then
    controlrobot
  else
    if checkbox1.Checked = false then
      basededatos;
```

```
end;
```

```
end.
```

9. Conclusión

Una vez terminado el periodo de investigación, se puede concluir que en nuestro país el tema de los microcontroladores se esta recién conociendo y por ende hay miles de aplicaciones para estos que aún no se han desarrollado. Por otro lado la comunicación inalámbrica esta llegando de a poco en lo que son las redes 802.11b y 802.11g. Cuando hablamos sobre la comunicación inalámbrica entre microcontroladores podemos decir que al parecer aún hay pocas instituciones o universidades que están investigando sobre ello, es por eso que durante el periodo de investigación se fue dificultoso conseguir información ya que era muy escasa, dado aquello salta a la vista de que es una tecnología poco explorada y de seguro sera muy valorada cuando se logre crear un producto basado en ello.

En cuanto a la tecnología de los celulares, tiene una vida de cerca de una década en nuestro país y ha tomado gran terreno en el ámbito económico, por lo cual el desarrollar aplicaciones para equipos celulares es algo que se debe tomar en serio por el enorme negocio que esto puede significar.

La tecnología bluetooth es algo que tiene un buen tiempo en el extranjero pero que a Chile recién empieza a entrar, esta tecnología es extremadamente útil y confiable, la comunicación entre dispositivos móviles y PCS ya no es la misma después de la incorporación de el Standard bluetooth, lo que hace muy atractivo para los desarrolladores el investigar en posibles aplicaciones para dispositivos móviles ya que integra la conectividad local a otro nivel. Con la incorporación de la nueva clase 1 en el Standard bluetooth que nos permitirá un alcance de 100 metros lo cual nos brinda un rango de trabajo realmente increíble.

Se ha logrado un nivel básico de comunicación entre distintos sistemas, lo que demuestra que es posible la comunicación entre microcontroladores y

celulares usando un equipo que funciones como nexo entre ellos, además de tener una comunicación completamente sin cables.

10. Bibliografía

- Spanbauer, Scott. “Los teléfonos inalámbricos”, Revista PC World, octubre del 2000.
- "<http://www.noticias3d.com/articulos/>", Tecnologías de semiconductores.
- “PDA’S en alza”, Revista PC World, abril del 2002.
- Argüello, Francisco. “http://web.usc.es/~elusive/index_e.html”, “Simulación de señales digitales en banda base”, año 2001.
- “<http://tecnologia.123.cl/otrastecno/telefoniamovil/>”, Celulares en el mundo.
- “La evolución de los sistemas inalámbricos”, Inédito.
- López, José Luis. “La próxima frontera: Virus en teléfonos celulares”, “<http://zonavirus.com/articulos.asp>”.
- “<http://www.yucatan.com/especiales/celular/>”, Historia del teléfono celular.
- Martínez, Evelio. “Evolución de la tecnología celular”, Revista Net, Julio 2001.
- Gómez Cárdenas, Roberto. “Los sistemas distribuidos inalámbricos y móviles”, <http://webdia.cem.itesm.mx/dia/ac/rogomez>.
- Martínez, Evelio. “Telecomunicaciones Infrarrojo/Láser”. <http://www.eveliux.com/>.
- Polo, Jesús. “Arquitectura de microcontroladores”, Documento inédito, Diciembre 1997.
- “<http://www.pictronic.com>”, Placa de aprendizaje para microcontroladores PIC.
- Vega, Alejandro. “Manual microcontrolador 8051”, Documento inédito. 1999.
- "<http://www.geocities.com/CapeCanaveral/Lab/9827/contenido.htm>", Microcontroladores PIC16CXX.
- “<http://www.microcontroladores.com/parallax.asp>”, Stamp Basic II.