

UNAB



BT00003352

DISEÑO Y DESARROLLO DE UN SISTEMA DOMÓTICO DIDACTICO PARA  
EL CONTROL DE ILUMINACIÓN Y PRESENCIA EN UN RECINTO  
MEDIANTE TECNOLOGÍA INALAMBRICA BLUETOOTH

JOSE FERNANDO ELIZALDE DIAZ  
LUIS CARLOS ORTIZ NIÑO

UNAB BIBLIOTECA		
VENDEDOR	FECHA	No CLASIFICACION
	22 NOV 2005	IMT
PRECIO	No. INVENTARIO	2104
	662577	

BUCARAMANGA  
UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
FACULTAD DE INGENIERÍA MECATRONICA  
2005

NOTA DE ACEPTACION

---

---

---

---

---

Presidente del Jurado

Jurado

---

Jurado

---



## **AGRADECIMIENTOS**

A Dios por regalarnos la oportunidad de ir paso a paso cumpliendo con nuestros sueños.

A nuestras familias por su apoyo incondicional y por su preocupación constante para la consecución de nuestras metas.

A nuestro director de tesis Ingeniero Jhon Faber Archila, por su constante apoyo y dedicación en la realización de este proyecto.

A todas las personas que de una u otra forma colaboraron en la realización de este proyecto.

062577

A mis Padres, motor y horizonte de mi vida, y a quienes debo todo. A Dios por darme la vida en la cual he podido conocer tantas cosas maravillosas, y a mi Madrecita del Cielo, que siempre esta con migo y nunca me deja solo.

**JOSE FERNANDO**

A Dios, a mis padres y a esa estrellita que sin necesidad de estar en el cielo  
ilumina todos los pasos de mi existir.

**LUIS CARLOS**

## TABLA DE CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN	1
OBJETIVO GENERAL	4
OBJETIVOS ESPECÍFICOS	4
1. MARCO TEÓRICO	5
1.1. DOMÓTICA	5
1.2. ESQUEMA GENERAL	7
1.3. CARACTERÍSTICAS	7
1.4. ELEMENTOS DE LA DOMÓTICA	8
1.4.1. Energía	9
1.4.2. Seguridad	9
1.4.2.1. Seguridad de los bienes	9
1.4.2.2. Seguridad de personas	10
1.4.3. Incidentes, Averías y confort	10
1.4.4. Comunicaciones	10
1.5. MEDIOS DE TRANSMISIÓN	11
1.5.1. Líneas de distribución de energía eléctrica	11
1.5.2. Cables de cobre	11
1.5.3. Fibra óptica	11
1.5.4. Infrarrojos	12
1.5.5. Radiofrecuencias	12
1.6. BLUETOOTH	13
1.7. APLICACIONES DE BLUETOOTH	14
1.8. FUNCIONAMIENTO DE LA TECNOLOGÍA BLUETOOTH	15
1.9. ARQUITECTURA DE HARDWARE	16
1.10. ARQUITECTURA DE SOFTWARE	19

@62577

	<b>Pág.</b>
1.11. REDES BLUETOOTH	21
1.12 TRANSMISIÓN	22
1.13 PROTOCOLO DE CONEXIÓN	23
1.14 SEGURIDAD Y CORRECIÓN DE ERRORES	24
1.15. USB(UNIVERSAL SERIAL BUS)	25
1.16. MICROCONTROLADORES	28
1.17 CONTROLADOR Y MICROCONTROLADOR	28
1.18. EL MERCADO DE LOS MICROCONTROLADORES	30
2. DISEÑO	32
2.1. METODOLOGÍA DEL DISEÑO	34
2.2. DISEÑO ELECTRONICO	39
2.3. COMUNICACIÓN ENTRE EL MICROCONTROLADOR Y EL PC POR MEDIO DEL PUERTO USB	40
2.4. SELECCIÓN DEL MICROCONTROLADOR	40
2.4.1. Microcontrolador PIC18F2550	43
2.5. COMUNICACIÓN VIA PUERTO USB	45
2.5.1. Conexión USB	46
2.5.2. Diagrama de capas del USB	49
2.5.3. Cables y Conectores	51
2.6. SENSORICA	52
2.6.1. Control de presencia e Iluminación	52
2.6.2. Medición de Presencia	53
2.6.3 Elección del Sensor de Movimiento	53
2.6.4. Sensor de movimiento pasivo (PIR)	54
2.6.5. Sensor de presencia DS940PT	55
2.6.5.1. Procesamiento de Señales	56
2.6.5.2. Cobertura	57
2.7. MEDICIÓN DE LUMINOSIDAD	58
2.7.1. Fotorresistencia (LDR)	59
2.7.2. Funcionamiento de una LDR	60

	<b>Pág.</b>
2.8 DISEÑO DE SOFTWARE	64
2.9. SOFTWARE DE COMUNICACIÓN ENTRE EL PC Y LA PLATAFORMA DE CONTROL	65
2.9.1. Selección de Software	65
2.9.1.1. Labview	65
2.10. INTERACCION MICROCONTROLADOR – LABVIEW	68
2.10.1. CIN (Nodo de Interfaz de Código)	68
2.11. COMUNICACIÓN BLUETOOTH	69
2.11.1. Conexión entre el PC y otros dispositivos con tecnología Bluetooth	69
2.11.2. Adaptador USB-Bluetooth (TBW-101UB)	69
2.12. COMUNICACION DEL SISTEMA DENTRO DE UNA LAN	71
2.13. PROTOCOLO DE COMUNICACIÓN TCP/IP	71
3. DESARROLLO	72
3.1. CIRCUITO ELECTRONICO (INTERFACE SISTEMA DE CONTROL)	72
3.2. DISEÑO PCB	78
3.3. DESARROLLO DEL SOFTWARE	81
3.4. SOFTWARE PARA EL MICROCONTROLADOR PIC18F2550	81
3.5. USANDO LA PICDEM FS USB DEMONSTRATION BOARD PARA ACONDICIONAR OTROS DISPOSITIVOS	83
3.6. SOFTWARE PARA COMUNICACIÓN ENTRE LA PLATAFORMA DE CONTROL Y EL SISTEMA DE MONITOREO	84
3.6.1. Conexión entre dos estaciones usando Labview 7	87
4. PRACTICAS	94
5. CONCLUSIONES	99
BIBLIOGRAFÍA	103
ANEXOS	104



## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1 Campos de la Domótica	1
Figura 2 Domótica, Integración de sistemas	7
Figura 3 Elementos de la Domótica	9
Figura 4 Logotipo de la Tecnología Bluetooth	13
Figura 5 Relación de la Frecuencia utilizada por Bluetooth y otras frecuencias	15
Figura 6 Conformación de un Dispositivo Bluetooth	18
Figura 7 Arquitectura de Software de un dispositivo Bluetooth	20
Figura 8 Arquitectura de la red basada en el modelo OSI	21
Figura 9 Topología de una red Bluetooth	22
Figura 10 Posibles dispositivos controlador por el puerto USB	26
Figura 11 Diseño del Sistema Domótico	32
Figura 12 Metodología de un Diseño Domótico	34
Figura 13 Integración de la Mecatrónica en un Sistema Domótico	35
Figura 14 Esquema de funcionamiento del Sistema Domótico	36
Figura 15 Pasos a seguir para el diseño del sistema domótico	37
Figura 16 Estructura de capas del bus USB	46
Figura 17 Esquema de un concentrador	48
Figura 18 Posible esquema de conexiones del bus USB	49
Figura 19 Capas del sistema de comunicación USB	50
Figura 20 Esquema del cable para USB	51
Figura 21 Vista de un sensor de presencia REF: DS940PT	56
Figura 22 Cobertura del Sensor DS940PT	57
Figura 23 Representación gráfica de una LDR	59
Figura 24 Característica resistencia-iluminación de una LDR	62
Figura 25 Respuesta espectral de distintos fotoconductores intrínsecos	63
Figura 26 Diagrama de Bloques de un programa en Labview	66

	<b>Pág.</b>
Figura 27 Adaptador USB Bluetooth REF: TBW 101UB	69
Figura 28 Circuito electrónico del sistema de control	73
Figura 29 Conexión del sistema de oscilación	75
Figura 30 Tipos de conectores USB	78
Figura 31 Diseño del circuito PCB, ingreso partes del circuito	79
Figura 32 Diseño del circuito PCB	79
Figura 33 Borneras instaladas en el circuito	80
Figura 34 Circuito PCB con los nombres de los elementos	80
Figura 35 Kit PICDEM FS USB DEMONSTRATION BOARD	82
Figura 36 Circuito electrónico del Kit PICDEM FS USB DEMONSTRATION BOARD	83
Figura 37 Captura del Sistema de Monitoreo	85
Figura 38 Asistente para la creación de un CIN	86
Figura 39 Diagrama de flujo del establecimiento de las estaciones activa y pasiva	92
Figura 40 Operación de lectura o recepción de datos de una conexión TCP/IP	93



## LISTA DE TABLAS

	<b>Pág.</b>
TABLA 1 Ventajas y desventajas de la conexión por la línea de distribución eléctrica	11
TABLA 2 Ventajas y desventajas de la conexión por fibra óptica	12
TABLA 3 Diseño del sistema Domótico	35
TABLA 4 Principales Características del PIC18F2550	45
TABLA 5 Anchura de banda prohibida, en electrón-voltios, y longitud de onda máxima, en micras, para diversos semiconductores intrínsecos	61
TABLA 6 Elementos circuito electrónico sistema de control	74
TABLA 7 Valores para cristal y condensadores (Microchip)	75
TABLA 8 Características eléctricas de un conector USB	77
TABLA 9 Características de un cable USB	77
TABLA 10 Configuración de una estación pasiva en Labview	88
TABLA 11 Configuración de una estación activa en Labview	89
TABLA 12 Configuración de bloques para leer y escribir en Labview	89
TABLA 13 Configuración para recibir datos con Labview	90

## INTRODUCCIÓN

Son las 6:30 de la mañana, suena el despertador, se levantan las persianas, las luces de la casa se apagan, el calentador del agua se enciende y la cafetera comienza a calentarse. En el exterior de la casa el sistema automático de riego, humedece el césped, se apagan las luces exteriores de la casa, todo esta listo para el abandono de la familia. Cuando todos se disponen a dejar la casa, el sistema avisa si hay alguna puerta o ventana abierta, y se procede a activar el sistema de seguridad. Al llegar a su oficina el dueño de casa, podrá monitorear desde su PC el interior y el exterior de la casa, y podrá ser prevenido sobre cualquier eventualidad que ocurra en ella. En la noche, faltando media hora para que la familia retorne nuevamente, el sistema enciende las luces exteriores e interiores, y procede a ejecutar algunas órdenes impartidas por el dueño de casa, y así se prepara para el fin de un día, y el comienzo nuevamente de otro.

Con el pasar del tiempo el ser humano se ha concentrado tanto en sus labores cotidianas (trabajo, estudio, deportes, etc.), en fin en todo lo que hoy mantiene al hombre ocupado, y lo vuelve reacio de realizar algunas labores domesticas que hasta hace pocos años solo eran realizadas por los integrantes de la casa. Las personas se levantaban más temprano y regaban el jardín, apagaban las luces del exterior de la casa, y alistaban lo necesario para hacer el desayuno y empezar un nuevo día de trabajo.

Para cubrir estas necesidades, surge la domótica, que esta definida como el uso simultáneo de la electricidad, la electrónica, y la informática, aplicadas a la gestión de viviendas. Esta nueva generación de viviendas inteligentes nos permite controlar nuestras casas, inclusive cuando estemos a miles de kilómetros de ellas,

solo con conectarnos a Internet, o simplemente enviando algunos datos por nuestro teléfono celular.

En nuestro país este concepto de tecnología esta empezando a tomar mucha importancia, como ejemplo tenemos el Edificio Inteligente de las Empresas Publicas de Medellín, el cual esta controlado por un sistema que se encarga de mantener constante y agradable la temperatura de las diferentes oficinas del edificio, también consta de una serie de sensores que controlan las persianas las cuales son graduadas dependiendo de la intensidad de los rayos de sol, el mismo sistema es el encargado de controlar la luz de los pasillos y oficinas, las cuales se encienden ante la presencia de alguna persona y cuando las personas abandonan el lugar, el sistema esta encargado de apagar las luces. Pasando a un plano mas sencillo, las personas se están interesando en estos sistemas por la posibilidad de ofrecer seguridad, confort, a los habitantes de sus hogares, y por la posibilidad de obtener un ahorro en los servicios públicos. Por estas y muchas características mas, los sistemas domóticos se están convirtiendo en una necesidad para el buen vivir de las personas.

En estos sistemas, la comunicación es una pieza fundamental, y dependiendo del sistema usado, podremos obtener buenos o malos resultados. El Bluetooth es un nuevo sistema de comunicación que esta siendo empleado a nivel mundial para comunicar Computadores, Celulares, Impresoras, Cámaras Digitales, etc., en fin sus aplicaciones son diversas. Esto fue lo que nos motivo a implementar este tipo de comunicación en nuestro proyecto.

Con este proyecto se quiere resaltar la aplicación que tiene la domótica en el mejoramiento de nuestras vidas. En este libro se plantea un proyecto que sirve para el control de la iluminación y la presencia en un recinto. Para el desarrollo de



este proyecto se implementó el uso de diferentes tecnologías, entre ellas están, La Comunicación Inalámbrica Bluetooth, la adquisición de datos a través del puerto de comunicación USB, entre otras. Para lograr la interacción correcta de nuestro sistema domótico, se emplean Software como Labview y C++.

En la primera parte de este libro se puede encontrar todo lo referente al fundamento teórico del proyecto, seguido por el diseño y el desarrollo de este, empleando las tecnologías anteriormente mencionadas. El proyecto también cuenta con un capítulo de tipo práctico, que tiene como fin servir para el Semillero de Investigación de la Facultad de Ingeniería Mecatronica de la Universidad Autónoma de Bucaramanga.

Esperamos que con este proyecto se abran nuevos horizontes en el conocimiento de nuevas tecnologías y también se fomente el uso de las mismas en nuestra vida cotidiana.

## **OBJETIVO GENERAL**

Diseñar, desarrollar y construir un sistema Domótico didáctico para el control de iluminación y presencia en un recinto, mediante tecnología inalámbrica (BlueTooth).

## **OBJETIVOS ESPECIFICOS**

- Diseñar y seleccionar los componentes del sistema domótico de iluminación y presencia
- Usar un microcontrolador, como plataforma con la cual se desea controlar el sistema domótico.
- Lograr una interacción remota con el sistema a controlar, de manera inalámbrica por medio de tecnología Bluetooth
- Desarrollar un algoritmo que permita la interacción del usuario y el sistema domótico y sea capaz de monitorear y vigilar las variables de iluminación y presencia que se presente en un recinto.
- Diseñar e implementar 2 prácticas académicas para el sistema domótico.

# 1. MARCO TEORICO

## 1.1. DOMÓTICA<sup>1</sup>

En 1988 el termino domótica se definía como “El concepto de vivienda que integra todos los automatismos en materia de seguridad, gestión de la energía, comunicaciones, etc”. En conclusión su principal objetivo es el de asegurar al usuario de la vivienda un aumento del confort, la seguridad, ahorro energético, y las facilidades de comunicación. Los primeros en usar el termino “Domotique”, fueron los franceses. En la figura 1, se muestra los campos de acción de la domótica



Los autores

---

<sup>1</sup> Tomado de [www.domointel.com](http://www.domointel.com)

Una definición mas técnica sería "Conjunto de servicios de la vivienda garantizado por sistemas que realizan varias funciones, los cuales pueden estar conectados entre si y a redes interiores y exteriores de comunicación". Gracias a todo esto podemos obtener un considerable ahorro de energía, una eficaz gestión técnica de la vivienda, una buena comunicación con el exterior y un alto nivel de seguridad. Para que un sistema sea considerado "inteligente" tiene que estar basado en las nuevas tecnologías de la información (NTI).

Dependiendo de la localización geográfica de la vivienda puede adoptar diferentes términos como: Casa Inteligente (Smart House), Automatización de viviendas (**Home Automation**), Domótica (**Domotique**), Sistemas Domésticos (**Home Systems**), etc.

La domótica no es solo para oficinas, hoteles. Colegios y viviendas particulares también pueden requerir los servicios que proporciona la domótica. Un sistema domótico es flexible, versátil, y adaptable a cualquier tipo de edificio y a cualquier actividad en él que se vaya a desarrollar.

Un sistema domótico proporciona un sinnúmero de beneficios y ventajas inalcanzables mediante una instalación tradicional.

Resumiendo, las principales razones para instalar un sistema domótico son: Comodidad, Seguridad, Confort, Información, Ahorro Energético. Pero sin duda alguna estas razones se resumen en una sola: AUMENTO DE LA CALIDAD DE VIDA.



## 1.2. ESQUEMA GENERAL

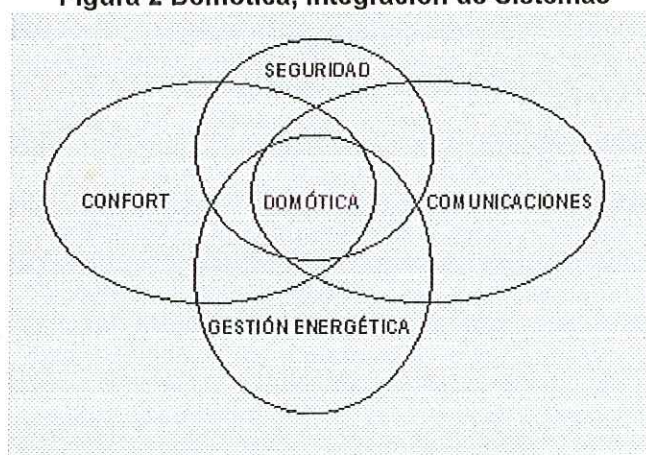
Un sistema domótico dispone de un sistema de comunicación que permite la interconexión de una serie de equipos, los cuales se encargan de obtener la información sobre el entorno doméstico y basándose en ésta, realizar unas determinadas acciones sobre dicho entorno.

Los elementos como (detectores, sensores, actuadores, etc.), transmiten las señales a una unidad central inteligente que tratará la información recibida. En función de dicha información y de una determinada programación, la unidad central actuará en consecuencia.

## 1.3. CARACTERÍSTICAS

Las características con las cuales debe contar una vivienda inteligente como lo muestra la figura, son las siguientes:

**Figura 2 Domótica, Integración de Sistemas**



Los autores

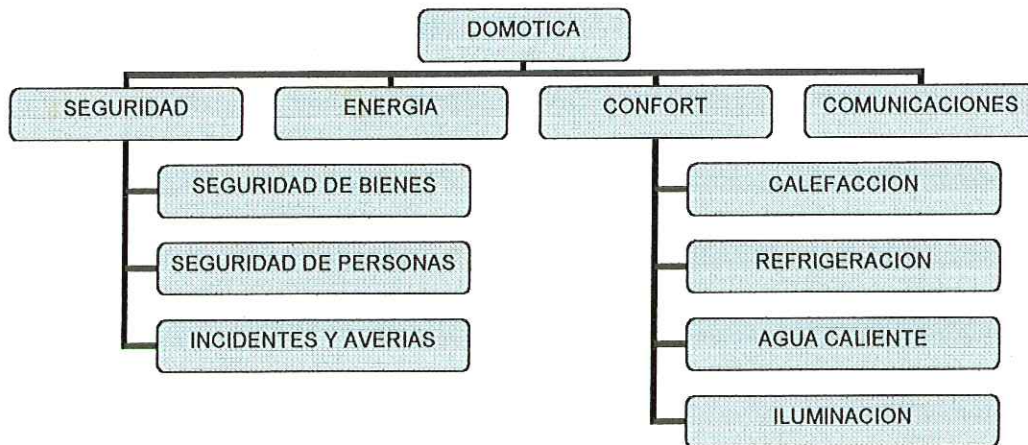


- Control remoto en el interior de la vivienda: A través de un esquema de comunicación con los distintos equipos. Reduce la necesidad de moverse dentro de la vivienda, esto puede ser particularmente importante en el caso de personas de la tercera edad o personas discapacitados.
  
- Control remoto en el exterior de la vivienda: Crea la posibilidad de realizar cambios en las tareas previamente predefinidas, (por ejemplo: la posibilidad de que el usuario encienda el sistema de calefacción desde el exterior de la vivienda), en consecuencia de esto el usuario tendrá un mejor aprovechamiento de el tiempo.
  
- Programabilidad: Una de las principales características de estos sistemas es la posibilidad de programar ciertas funciones, las cuales se lleven a cabo dependiendo de varios factores como (hora, temperatura interior o exterior, etc.), esto da un aumento en el confort y un ahorro considerable de tiempo.
  
- Acceso a servicios externos: Servicios de información, telecompra, telebanco, etc.

#### **1.4. ELEMENTOS DE LA DOMÓTICA**

La domótica se encarga de gestionar cuatro aspectos principalmente, para incrementar nuestra calidad de vida. En la figura 3 se muestran los cuatro elementos principales de la domótica.

Figura 3 Elementos de la Domótica



Los autores

1.4.1. Energía. En este campo, la domótica se encarga de gestionar el consumo de energía, para esto se usan temporizadores, relojes, programadores, termostatos, etc. En algunos sistemas se aprovechan las tarifas energéticas más económicas para usar acumuladores de carga.

1.4.2. Seguridad. La seguridad que nos proporciona un sistema domótico es mucho mayor que la que nos puede ofrecer cualquier otro sistema, pues se integran tres campos de la seguridad que normalmente están controlados por sistemas distintos.

1.4.2.1. Seguridad de los bienes. Gestión del control de acceso y control de presencia, así como la simulación de presencia.

1.4.2.2. Seguridad de personas. Especialmente, para las personas mayores y las personas enfermas. Mediante un nodo telefónico, se puede tener acceso a los servicios de Ambulancia, Medico, Policía, etc.

1.4.3. Incidentes, averías y confort. Mediante una serie de sensores, se pueden detectar los incendios y las fugas de gas y agua, y mediante el sistema telefónico avisar de manera inmediata a los bomberos o la policía según sea el caso de la emergencia.

La domótica nos proporciona una serie de comodidades, como pueden ser el control automático de los servicios de:

- Calefacción
- Refrigeración
- Agua Caliente
- Iluminación

1.4.4. Comunicaciones. La domótica tiene una característica fundamental, que es la integración de sistemas, por eso hay nodos que interconectan la red domótica con diferentes dispositivos, como la red telefónica, las videocámaras, los sensores, etc.

## **1.5. MEDIOS DE TRANSMISIÓN**

En todo sistema domótico con arquitectura distribuida, los diferentes elementos de control deben intercambiar información unos con otros a través de un soporte físico (par trenzado, red eléctrica, radio, infrarrojos, etc.).

1.5.1. Líneas de distribución de energía eléctrica. Si bien no es el medio más adecuado para la transmisión de datos, si es una alternativa a tener en cuenta para las comunicaciones domesticas dado el bajo costo que implica su uso, dado que se trata de una instalación existente.

TABLA 1 Ventajas y desventajas de la conexión por la línea de distribución eléctrica

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
<ul style="list-style-type: none"> <li>▪ Bajo costo en la instalación</li> <li>▪ Facilidad de Conexión</li> </ul>	<ul style="list-style-type: none"> <li>▪ Poca fiabilidad en la transmisión de los datos.</li> <li>▪ Baja velocidad de transmisión.</li> </ul>

1.5.2. Cables de cobre. Los cables de cobre como soporte para la transmisión de datos son un buen medio. Puesto que, la infraestructura de las redes de comunicación actuales, tanto públicas como privadas, tiene en un porcentaje muy elevado de cables de cobre como soporte de transmisión de las señales eléctricas.

1.5.3. Fibra óptica. La fibra óptica esta constituida por un material dieléctrico transparente, conductor de luz, compuesto por un núcleo con un índice de refracción menor que el del revestimiento, que envuelve ha dicho núcleo. Estos dos elementos forman una guía para que la luz se desplace por la fibra.



TABLA 2 Ventajas y desventajas de la conexión por fibra óptica

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> <li>▪ Fiabilidad en la transferencia de los datos.</li> <li>▪ Inmunidad frente a interferencias electromagnéticas y de radiofrecuencias.</li> <li>▪ Distancia ilimitada entre los puntos de instalación.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Elevados costos de instalación.</li> <li>▪ Ideal solo en casos donde se necesitan transmisión de grandes cantidades de datos.</li> </ul>

1.5.4. Infrarrojos. El uso de mandos a distancia basados en transmisión por infrarrojos esta ampliamente extendida en la casa para manejar equipos de audio y vídeo. Dentro de sus ventajas se encuentran la comodidad, la flexibilidad y la posibilidad de agrupar un gran numero de aplicaciones. Al tratarse de un medio de transmisión óptico es inmune a las radiaciones electromagnéticas producidas por los equipos domésticos o por los demás medios de transmisión (coaxial, cables pares, red de distribución de energía eléctrica, etc.).

1.5.5. Radiofrecuencias. Este medio de transmisión puede parecer, en principio, idóneo para el control a distancia de los sistemas domóticos, dada la gran flexibilidad que supone su uso. Sin embargo resulta particularmente sensible a las perturbaciones electromagnéticas.

## 1.6. BLUETOOTH<sup>2</sup>

Figura 4 Logotipo de la Tecnología Bluetooth



Tomado de: [www.bluetooth.org](http://www.bluetooth.org)

El nombre de esta tecnología procede del rey vikingo Harald Blatand, conocido por los ingleses como BLUETOOTH, quien reino en Dinamarca y Noruega en la segunda mitad del siglo X y fue considerado como uno de los hombres mas importantes de Europa.

Durante el año 1994, surge la idea de investigar la posibilidad de crear un dispositivo de bajo costo que sirviera para comunicar diversos dispositivos, la idea era hacerlo basado en un estándar estricto para que su uso se popularizara y diversos fabricantes pudieran desarrollar dispositivos que lo utilizaran.

En 1998, un grupo conformado por las empresas lideres en telecomunicaciones y computadoras, entre las que se encontraban Intel, IBM, Toshiba, Ericsson, y Nokia, estuvieron trabajando en el desarrollo de dicho dispositivo. A comienzos del año 1997, ante el avance del proyecto, se fue despertando el interés de otros fabricantes de equipos portátiles y empresas de telecomunicaciones. En seguida se vio claramente que para que el sistema tuviera éxito, un gran número de equipos deberían estar equipados con esta tecnología. Esto origino en 1998 la creación del SIG<sup>3</sup> el cual estaba conformado por 5 de los promotores del sistema.

---

<sup>2</sup> Tomado de BRAY, Jennifer. Bluetooth: Connect without cables. Upper Saddle River: Prentice Hall, 2001, 495 p.

<sup>3</sup> SIG: Special Interest Group

La idea era lograr un conjunto adecuado de áreas de negocios, dos líderes del mercado de las telecomunicaciones (Ericsson y Nokia), dos líderes en el mercado de los computadores portátiles (IBM y Toshiba), y un líder en la fabricación de los chips (Intel). El propósito principal del consorcio fue y es, el establecer un estándar para la interfase aérea junto con su software de control, con el fin de asegurar la interoperabilidad de los equipos entre los diversos fabricantes.

## **1.7. APLICACIONES DE BLUETOOTH**

El futuro de Bluetooth está fuertemente influenciado por las expectativas que tienen las diferentes compañías pertenecientes al Bluetooth SIG. Se espera que Bluetooth conforme a la norma 802.15.2 de coexistencia de las redes WLAN y que surjan versiones de alta y baja velocidad, para aplicaciones de multimedia y de dispositivos de baja complejidad respectivamente. Al crearse estos estándares, se ampliarán aún más las posibilidades para el uso de Bluetooth, por ejemplo, para el modelo de baja velocidad y baja complejidad (802.15 TG 4) se esperan las siguientes aplicaciones:

- Sensores
- Juguetes interactivos
- Carnets Inteligentes
- Controles Remotos
- Dispositivos para automatiza el hogar
- Etc.

En general, estos dispositivos deben tener un alto grado de simplicidad, bajo costo, bajos requerimientos de tasa de transferencia, y deben mantener una vida de batería de varios meses o varios años.



## 1.8. FUNCIONAMIENTO DE LA TECNOLOGÍA BLUETOOTH

La tecnología Bluetooth, proporciona conectividad entre cualquier tipo de dispositivo que cumpla con las especificaciones inalámbricas bluetooth. Además éste es un estándar libre lo que simplifica su uso para diseñar nuevos productos innovadores que se beneficien de la conectividad inalámbrica.

A diferencia de otros estándares inalámbricos, la especificación Bluetooth incluye dos capas, la capa de enlace y la de aplicación para los desarrolladores de productos que soportan voz y datos y aplicaciones de contenido centralizado.

La especificación de Bluetooth define un canal de comunicación de máximo 720Kb/seg con rango óptimo de 10m (opcionalmente 100m).

La frecuencia de radio con la cual se trabaja esta en un rango de 2.4 a 2.48Ghz con amplio espectro y saltos de frecuencia con posibilidades de transmitir en full duplex con un máximo de 1600 saltos/seg. Los saltos de frecuencia dan entre un total de 79 frecuencias con intervalos de 1Mhz, esto permite brindar seguridad y robustez.

Figura 5 Relación de la frecuencia utilizada por Bluetooth y otras frecuencias.



Tomado de: <http://www.monografias.com/trabajos12/tecninal/tecninal.shtml#intro>



La potencia de salida para transmitir a una distancia máxima de 10m es de 0dBm (1mW), mientras que para la versión largo alcance transmite entre -30 y 20dBm (100mW).

Para lograr un bajo consumo y bajo costo, se ideó una solución que se puede implementar en un solo chip utilizando circuitos CMOS<sup>4</sup>. De esta manera se consiguió una solución de 9x9mm y que consume aproximadamente 97% menos energía que un teléfono celular normal.

El protocolo de Banda Base (canales simples por línea) combina switching, circuitos y paquetes. Para asegurar que los paquetes no lleguen fuera de orden, los slots pueden ser reservados por paquetes sincrónicos, un salto diferente de señal es usado para cada paquete. Tres canales de datos sincrónicos (Voz), o un canal de datos sincrónico y uno asíncrono, pueden ser soportados en un solo canal. Cada canal de voz puede soportar una transferencia de 64Kb/s en cada sentido, la cual es suficientemente adecuada para la transmisión de voz. Un canal asíncrono puede transmitir como mucho 721Kb/s en una dirección y 56 Kb/s en la dirección opuesta, sin embargo, para una conexión asíncrona es posible soportar 432.6Kb/s en ambas direcciones si el enlace es simétrico.

## 1.9. ARQUITECTURA DE HARDWARE

El hardware que compone un dispositivo Bluetooth está compuesto por dos partes. Un dispositivo de radio, encargado de modular y transmitir la señal, y un

---

<sup>4</sup> **CMOS:** Complementary Metal Oxide Semiconductor. Es el tipo de tecnología de semiconductores más utilizada en la actualidad. Su principal ventaja es su bajo consumo de potencia.

controlador digital. El controlador digital esta compuesto por una CPU, por un procesador de señales digitales (DSP Digital Signal Processor) llamado Link Controller (o controlador de enlace) y de los interfaces con el dispositivo anfitrión. El LC o Link Controller está encargado de hacer el procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de capa física.

- **Protocolo ARQ:** (Automatic Repeat-reQuest) es un protocolo utilizado para el control de errores en la transmisión de datos, garantizando la integridad de los mismos. Éste suele utilizarse en sistemas que no actúan en tiempo real ya que el tiempo que se pierde en el reenvío puede ser considerable y ser más útil emitir mal en el momento que correctamente un tiempo después. Esta técnica de control de errores se basa en el reenvío de los paquetes de información que se detecten como erróneos (Esto quiere decir que no todos los paquetes de información se detectan como erróneos).

Para controlar la correcta recepción de un paquete se utilizan ACK's (acknowledge) y NACK's de forma que cuando el receptor recibe un paquete correctamente el receptor asiente con un ACK y si no es correcto responde con un NACK. Durante el protocolo que controla recepción de paquetes pueden surgir múltiples problemas (perdida de ACK, recibir un ACK incorrecto, etc,) complicándose así el contenido del ACK y surgiendo nuevos conceptos como el de timeout.

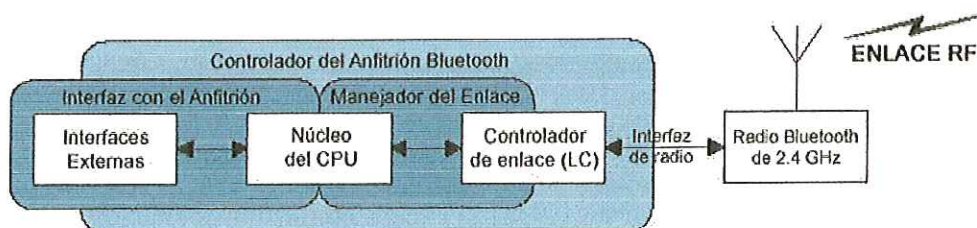
Si el emisor no recibe información sobre la recepción del paquete durante un tiempo fijado (timeout) éste se reenvía automáticamente.

- **Protocolo FEC: (Forward Error Correction)** es un protocolo utilizado para corregir errores en una transmisión. Este se utiliza en sistemas sin retorno o sistemas en tiempo real donde no se puede esperar a la retransmisión para mostrar los datos. El funcionamiento consiste en evaluar el síndrome del vector recibido (si no existe error el síndrome es el vector cero). Asociado a éste síndrome tenemos varios errores (para un código  $C(n,k)$   $2^k$  diferentes) de entre todos ellos tomaremos el más probable según las características del canal (en general el de menor peso). Para recuperar el vector original sumamos el error calculado al vector recibido de forma que si hemos acertado al suponer el error concreto de todos los asociados al síndrome cancelaremos su efecto. Esta técnica tiene sus limitaciones ya que la interpretación del vector error asociado al síndrome calculado no es exacta.

El LC también está encargado de las funciones de transferencia (tanto asíncrona como síncrona), codificación de audio y encriptación de datos.

El CPU del dispositivo se encarga de atender las instrucciones relacionadas con Bluetooth del dispositivo anfitrión, esto con el fin de simplificar su operación. Para ello, sobre el CPU corre un software denominado Link Manager que tiene la función de comunicarse con otros dispositivos por medio del protocolo LMP.

**Figura 6 Conformación de un Dispositivo Bluetooth**



Tomado de: [www.atenea.unicauca.edu.co/~dabravo/bluetooth/index.htm](http://www.atenea.unicauca.edu.co/~dabravo/bluetooth/index.htm)



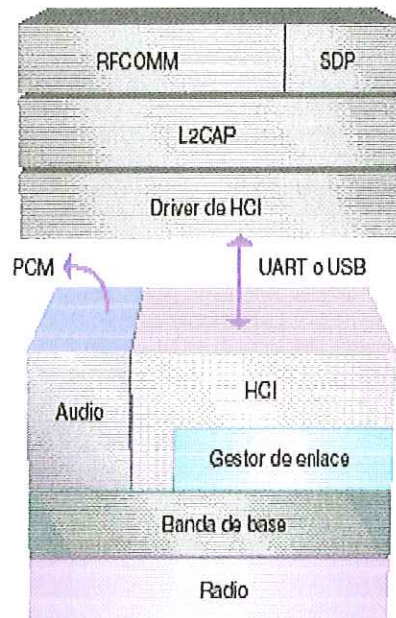
Entre las tareas realizadas por el LC y el Link Manager, están:

- Determinación de conexiones
- Envío y Recepción de Datos
- Empaginamiento y peticiones
- Autenticación
- Negociaciones y determinación de tipos de enlace
- Determinación del tipo de cuerpo de cada paquete
- Ubicación del dispositivo en modo sniff o hold.

## **1.10. ARQUITECTURA DE SOFTWARE**

Con el ánimo de ampliar la compatibilidad de los dispositivos Bluetooth, los dispositivos que se unan al estándar utilizan como interfaz entre el dispositivo anfitrión (laptop, teléfono celular, etc) y el dispositivo Bluetooth (Chip Bluetooth) una interfaz denominada HCI (Host Controller Interface).

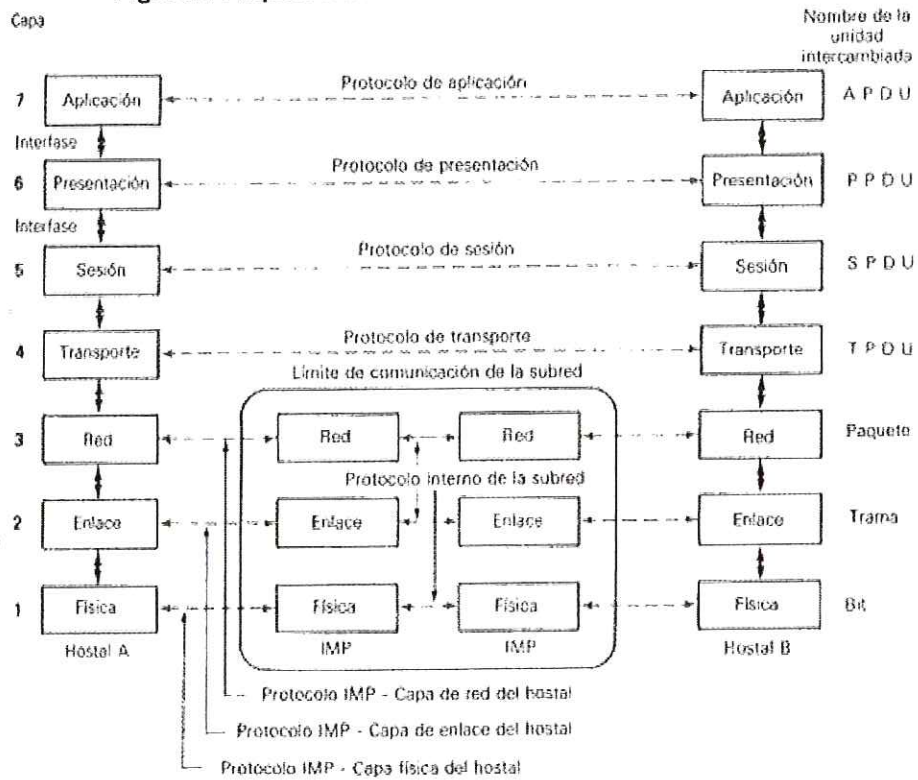
Figura 7 Arquitectura de Software de un Dispositivo Bluetooth



Tomado de: [www.bluetooth.org](http://www.bluetooth.org)

Los protocolos de alto nivel como el SDP (Utilizado para encontrar otros dispositivos Bluetooth dentro del rango de comunicación, encargado también, de detectar la función de los dispositivos en rango), RFCOMM (Protocolo utilizado para simular conexiones de puerto serial) y TCS (Protocolo de control de telefonía) interactúan con el controlador de banda base a través del Protocolo L2CAP(Logical Link Control and Adaptation Protocol). El protocolo L2CAP se encarga de la segmentación y reensamblaje de los paquetes para poder enviar paquetes de mayor tamaño a través de la conexión Bluetooth.

**Figura 8 Arquitectura de la red basada en el modelo OSI**

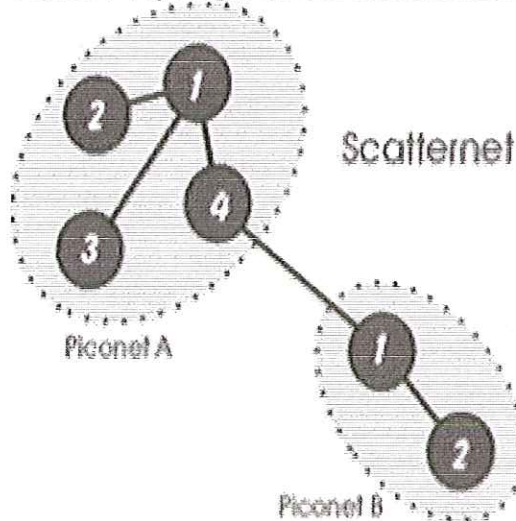


Tomado de: [http://web.frm.utn.edu.ar/comunicaciones/modelo\\_osi.html#1](http://web.frm.utn.edu.ar/comunicaciones/modelo_osi.html#1)

### 1.11. REDES BLUETOOTH

Las topologías de las redes Bluetooth puede ser punto-a-punto o punto-multipunto.

Figura 9 Topología de una Red Bluetooth



Tomado de: <http://atenea.unicauca.edu.co/~dabravo/bluetooth/quees.htm>

Los dispositivos, se comunican en redes denominadas piconets. Estas redes tienen la posibilidad de crecer hasta tener 8 conexiones punto a punto. Además, se puede extender la red mediante la formación de scatternet. Un scatternet es la red producida cuando dos dispositivos pertenecientes a dos piconets diferentes, se conectan.

En una piconet, un dispositivo debe actuar como master, enviando la información del reloj (para sincronizarse) y la información de los saltos de frecuencia. El resto de los dispositivos actúan como slaves.

## 1.12. TRANSMISIÓN

Bluetooth está diseñado para usar acuses de recibos y saltos de frecuencias, lo cual hará conexiones robustas. Este sistema está basado en el envío de paquetes, y saltarán a una nueva frecuencia después de que cada paquete es recibido, lo



cual no solo sirve para evitar problemas de interferencia, sino que aumenta el nivel de seguridad. La tasa de datos es un megabyte/segundo. Una transmisión "full duplex" (ambas direcciones al mismo tiempo) es realizado por multiplexaje de división de tiempo.

La transmisión puede ser realizada de manera síncrona o asíncrona. El método sincrónico Orientado a Conexiones (SCO) es usado principalmente para voz, y el Asíncrono No Orientado a Conexión (ACL) es principalmente usado para transmitir datos. Dentro de un "piconet" cada master-slave puede usar un modo de transmisión distinto, y los modos pueden ser cambiados en algún momento. La división de tiempo "Duplex", es usado para SCO y ACL, y ambos soportan 16 tipos de paquetes, cuatro de los cuales son paquetes de control, que son los mismos de cada tipo.

Debido a la necesidad de una seguridad en la transmisión de los datos, los paquetes SCO son entregados en intervalos reservados, es decir, los paquetes son enviados en grupos sin permitir la interrupción de otras transmisiones. Los enlaces ACL soportan tanto transmisión simétrica como transmisión asimétrica.

### 1.13. PROTOCOLO DE CONEXIÓN

Las conexiones Bluetooth, son establecidas a través de la siguiente técnica:

- **Standby:** Los dispositivos en una red o piconet, que no se encuentran conectados, se encuentran en modo standby. Estos escuchan mensajes cada 1.28 segundos, sobre 32 saltos de frecuencia.



- **Page/Inquiry:** Si un dispositivo desea hacer una conexión con otro dispositivo, éste le envía un mensaje de tipo page, si la dirección es conocida: o una petición a través de un mensaje de page, si éste no es conocido. La unidad "master" envía 16 page mensaje idénticos, en 16 saltos de frecuencia, a la unidad "slave". Si no hay respuesta, el "master" retransmite en los otros 16 saltos de frecuencia. El método de petición (inquiry) requiere una respuesta extra por parte de la unidad "slave", desde la dirección MAC, que no es conocida por la unidad "master".
  
- **Active:** Ocurre la transmisión de datos.
  
- **Hold:** Cuando el "master" o el "slave" desean, puede ser establecido un modo en el cual no son transmitidos datos. El objetivo de esto es conservar el poder.
  
- **Sniff:** El modo sniff, es aplicable solo para las unidades "slaves", es para conservar el poder. Durante este modo, el "slave", no toma un rol activo, en la red, pero escucha a un reducido nivel.
  
- **Park:** El modo park es un nivel más reducido, que el modo hold. Durante este, el "slave" es sincronizado a la red, por eso no requiere una reactivación completa, pero no es parte del tráfico. En este estado, ellos no tienen direcciones MAC y solo escuchan para mantener su sincronización con el "master" y chequear los mensajes de broadcast.

#### 1.14. SEGURIDAD Y CORRECCIÓN DE ERRORES

Para corregir los errores se usan las siguientes tres técnicas:

- 1/3 rate forward error correction code (FEC), este método es diseñado para reducir el número de retransmisión.
- 2/3 rate forward error correction code FEC
- Automatic Repeat Request (ARQ)

En cuanto a la Seguridad, se usan las siguientes técnicas:

- A través de saltos de frecuencia pseudos-aleatorios que dificultan que dispositivos ajenos a la red pueden interceptar o ver el tráfico de información.
- Autenticación, permite a un usuario controlar la conectividad para solo dispositivos especificados.
- Encriptación, se usan claves secretas con longitudes de 1 o 6 bits.

### **1.15. USB (UNIVERSAL SERIAL BUS)<sup>5</sup>**

En un principio teníamos la interfase serie – paralelo, pero con el avance de las computadores y de los dispositivos se hizo necesario la unificación de todos los conectores, para la cual se creo uno mas sencillo y de mejor rendimiento.

Si bien es cierto que estos conectores todavía cumplen su función correctamente en casos como la conexión del teclado, el mouse o un módem, en casos como la

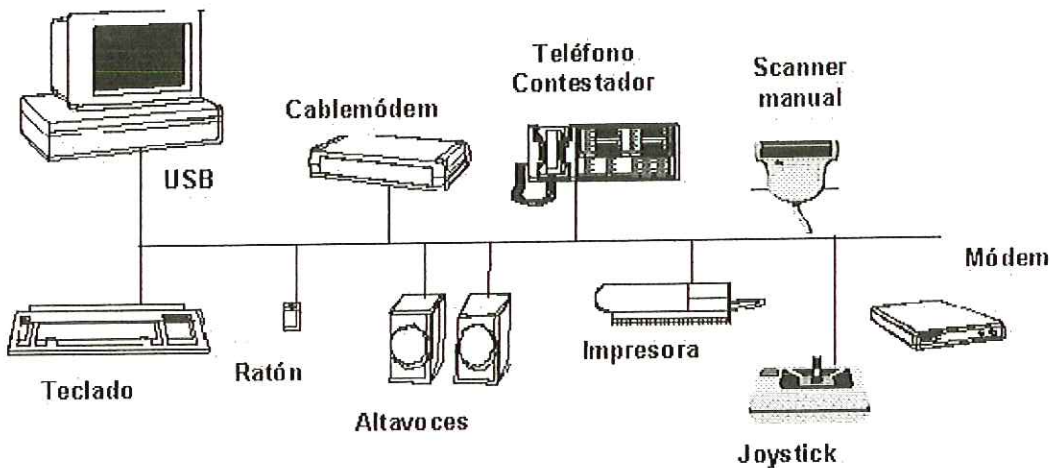
---

<sup>5</sup> Tomado de <http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

conexión de dispositivos rápidos como por ejemplo cámaras digitales, entre otros, estos conectores no son tan eficientes.

El USB nace para crear un estándar de entrada/salida de velocidad media-alta que permita conectar dispositivos que hasta ahora requerían de una tarjeta especial para sacarle el mayor rendimiento a cada dispositivo, lo que implicaba un sobre costo en cada producto.

**Figura 10 Posibles dispositivos controlados por el puerto USB**



Tomado de: <http://www.monografias.com/trabajos11/usbmemb/usbmemb.shtml>

El USB es una interfaz de transmisión de datos y distribución de energía. Esta interfaz de 4 hilos, 12 Mbps y plug and play<sup>6</sup> distribuye 5V de alimentación, está siendo ampliamente investigada y a su vez desarrollada por la industria informática.

El Universal Serial Bus, es un bus serie que hace posible la conexión de hasta 127 periféricos a una única puerta de un PC, con detección y configuración automática, siendo esto posible con el computador en funcionamiento, sin tener que instalar ninguna clase de hardware ni software y lo mas importante sin tener que reiniciar el computador.



Hasta el momento ha sido ampliamente desarrollado por empresas como Compaq, IBM, Intel, Microsoft, entre otras empresas. Actualmente se pueden conseguir periféricos como teclados, impresoras, pantallas, scanner, cámaras digitales, etc., que disponen de conexión de tipo USB.

Algunas características importantes del USB son:

- Posee dos velocidades de acceso, la velocidad baja es de 1.5Mbps y es usada para dispositivos que no necesitan un ancho de banda muy grande, y para dispositivos de gama alta (rápidos) como cámaras digitales, el USB tiene la posibilidad de trabajar a 12Mbps.
- Topología en estrella, lo que implica la necesidad de un dispositivo tipo hub que centralice las conexiones aunque en algunos dispositivos como teclados y monitores ya se implementa esta característica, lo que permite tener un solo conector al PC, y desde este dispositivo sacar conexiones adicionales.
- Tiene la posibilidad de suministrar energía eléctrica a dispositivos que no tengan un alto consumo y que no estén a más de 5 metros, lo que elimina la necesidad de conectar dichos periféricos a la red eléctrica.

El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG, mientras que el de los conductores de señal es de 28 AWG. La longitud máxima de los cables es de 5 metros.

---

<sup>6</sup> **Plug and Play:** Traduce enchufar y listo, este tipo de tecnología permite conectarse un dispositivo a un computador sin tener que reiniciar el sistema.



## **1.16. MICROCONTROLADORES<sup>7</sup>**

Los microcontroladores, están conquistando actualmente el mundo. Están en nuestro trabajo, en nuestra casa, en general, en nuestra vida. Si miramos detenidamente, los microcontroladores están encargados de controlar nuestros hornos microondas, los televisores, los teléfonos, en general muchos de los electrodomésticos que usamos a diarios, son gobernados por un microcontrolador.

## **1.17. CONTROLADOR Y MICROCONTROLADOR**

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Tomando un ejemplo en general, el controlador que regula un horno microondas dispone de un sensor que mide constantemente la temperatura y cuando esta supera los niveles establecidos, ejecuta una acción que previamente fue predefinida para poder corregir este problema. 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.

---

<sup>7</sup> Tomado de CUENCA, Martín. Microcontroladores PIC: La solución en un chip, Quinta Edición. Madrid: Paraninfo, 2001, 489 p.

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

Un microcontrolador dispone normalmente de las siguientes partes:

- Procesador o UCP(Unidad Central de Proceso)
- Memoria RAM
- Memoria para el programa de tipo ROM/PROM/EPROM
- Líneas de E/S para comunicación con otros dispositivos
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Convertidor Análogo/Digital, CDA: Convertidor Digital/Análogo, etc.)

Algunas de las ventajas que tienen los productos que en su interior contienen un microcontrolador son:

- Aumento de prestaciones: Un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: Al reemplazar un elevado número de elementos disminuye el riesgo de averías.
- Reducción del tamaño en el proceso de acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.

- Mayor flexibilidad: Las características de control están programadas por lo que su modificación sólo necesita cambio en el programa de instrucciones.

En resumen, un microcontrolador, es 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.

### **1.18. EL MERCADO DE LOS MICROCONTROLADORES**

Existe una gran diversidad de fabricantes de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4,8,16 ó 32 bits. Aunque las ventajas de los microcontroladores de 16 y 32 bits son superiores que los de 4 y 8 bits, la realidad es que los de 8 bits dominan el mercado y los de 4 bits se niegan a abandonar el mercado. La razón por la cual los microcontroladores de 4 y 8 bits son tan apetecidos esta en que son perfectos para la mayoría de las aplicaciones, que no necesitan microcontroladores tan sofisticados y costosos. Uno de los sectores que más impulsa el mercado de los microcontroladores es el sector automovilístico, siendo este sector uno de los más exigentes, ya que los microcontroladores tienen que trabajar en situaciones adversas como vibraciones, choques, ruido, etc. y seguir funcionando correctamente, porque el mal funcionamiento de uno de ellos, podría ocasionar un grave accidente.

En cuanto a la fabricación de los actuales microcontroladores, cabe decir que la totalidad de los microcontroladores existentes en el mercado son fabricados con tecnología CMOS (Complentary Metal Oxide Semiconductor). Esta tecnología es usada, dado que los microcontroladores fabricados bajo esta tecnología poseen una alta inmunidad al ruido y un bajo consumo.



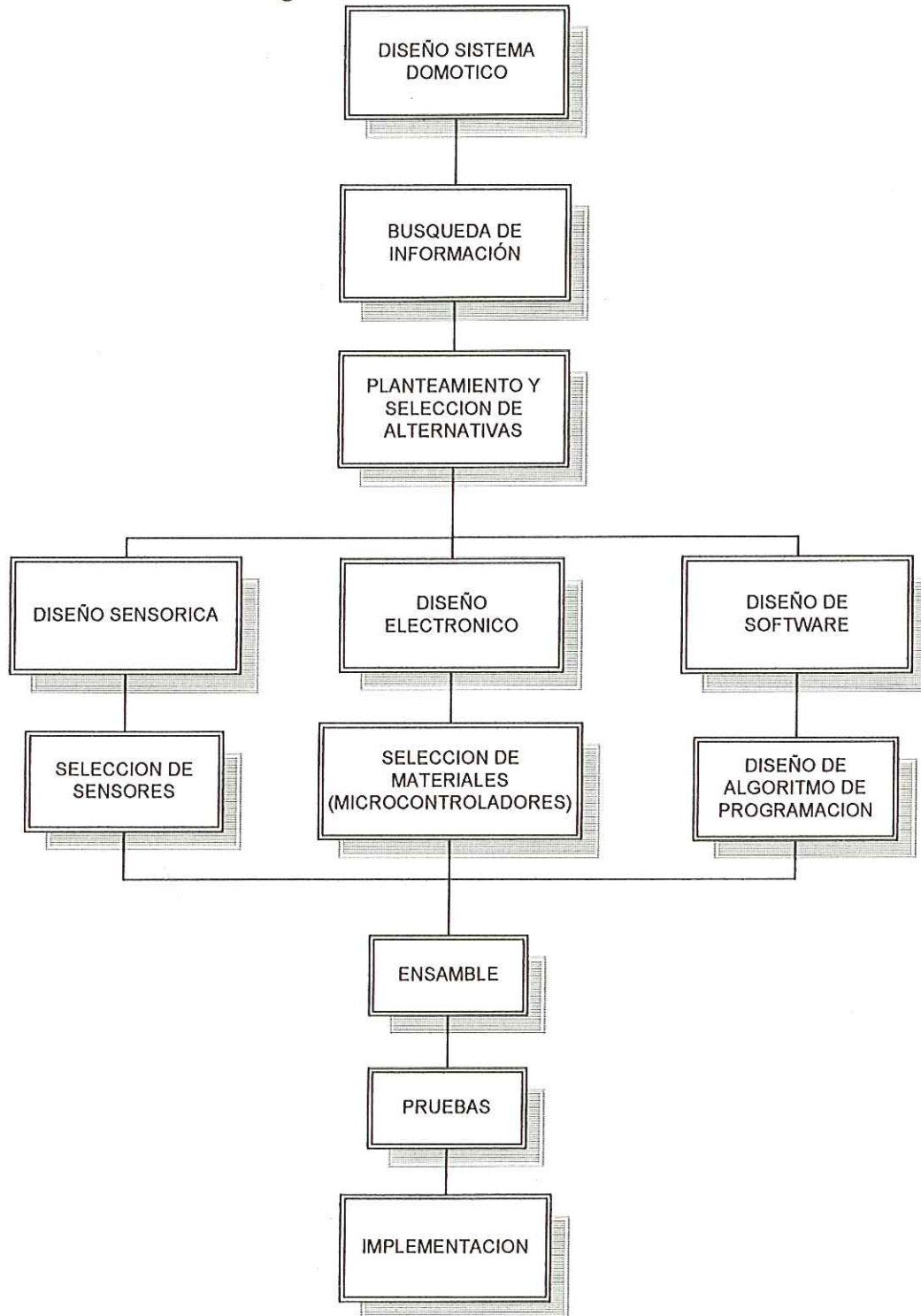
Los microcontroladores más populares son:

- 8048(Intel): Es el padre de los microcontroladores actuales. Su precio, disponibilidad y herramientas de desarrollo hace que todavía sea muy popular.
- 8051 (Intel y otros): Es sin duda el microcontrolador mas popular. Fácil de programar, pero potente, en el momento hay mucha documentación y posee cientos de variantes y herramientas de desarrollo.
- 80186, 80188 y 80386 EX (Intel): Versiones en microcontroladores de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.
- 68HC11 (Motorola y Toshiba): Es un microcontrolador de 8 bits potente y popular co gran cantidad de variantes.
- 683XX (Motorola): Surgió a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de altísimas aplicaciones.
- PIC (Microchip): Familia de microcontroladores que esta ganando popularidad día a día. Fueron los primeros microcontroladores RISC.



## 2. DISEÑO

Figura 11 Diseño del sistema domótico



Los autores

Para el diseño de este sistema, se inicio con la búsqueda de toda la información concerniente al proyecto. Después de recopilar toda esta información y de tener claro el camino a tomar para el diseño y construcción del proyecto, se procedió a diseñar la parte electrónica y el software encargado de controlar todo el sistema.

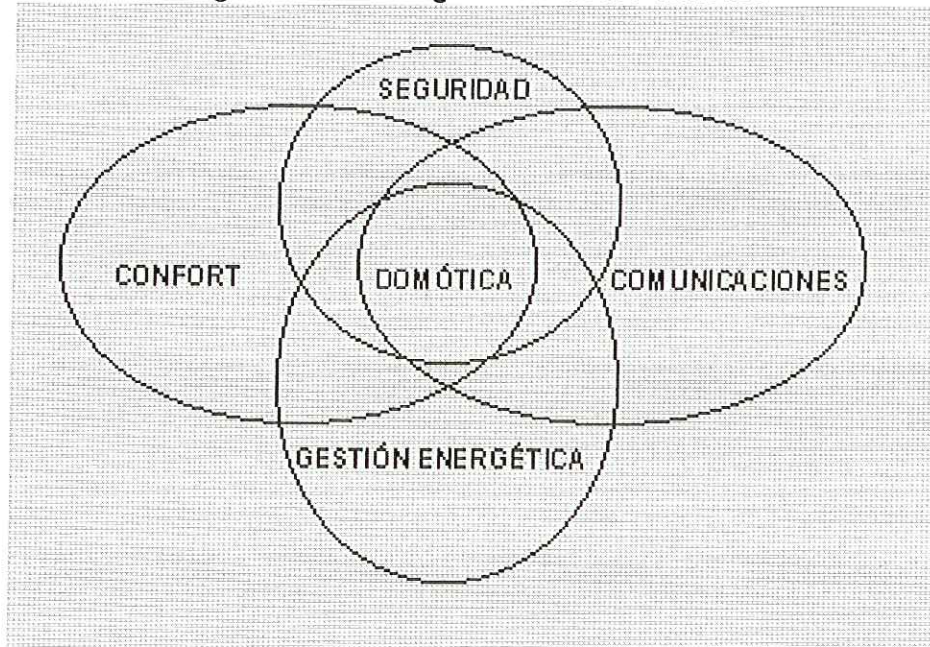
Para el diseño electrónico del proyecto, se miraron las diferentes alternativas que se presentaban. Luego de investigar a fondo sobre cada una, se pudo ver que algunas de ellas no eran viables, ya fuera por su costo, o por su imposibilidad de conseguirlas en el mercado en la cantidad necesitada. Por estas razones se escogió el diseño que se plantea en el presente capítulo. En lo correspondiente a la construcción del sistema domótico, en el parte correspondiente al diseño de software se eligió el programa que mejores alternativas y beneficios nos pueda brindar al diseñar el algoritmo de control. Para esta selección se tuvo en cuenta las características que podía presentar el programa, sus ventajas, desventajas y su facilidad de programación. Esta selección se puede observar más detalladamente en la sección de selección de software del presente capítulo.

Una vez hecho el diseño de la parte electrónica y de software se procede a ensamblar todo el sistema, posteriormente, se realizan las pruebas al sistema, con el fin de lograr un optimo rendimiento del mismo.

Después de realizadas las pruebas, el sistema se implementará en el recinto y será puesto a operar.

## 2.1. METODOLOGIA DEL DISEÑO

Figura 12 Metodología de un diseño domótico



Los autores

Para la realización de este proyecto domótico se necesita la integración tanto de la seguridad, las comunicaciones, la gestión energética y el confort. A su vez cada uno de estos componentes domóticos se integran a las partes mecatronicas involucradas en este proyecto como los son: Software, Electrónica, y Sensorica. La unión de estos tres componentes mecatronicos más la utilización de la comunicación inalámbrica Bluetooth, nos permite abarcar en su totalidad las ramas o componentes que involucra la domótica.

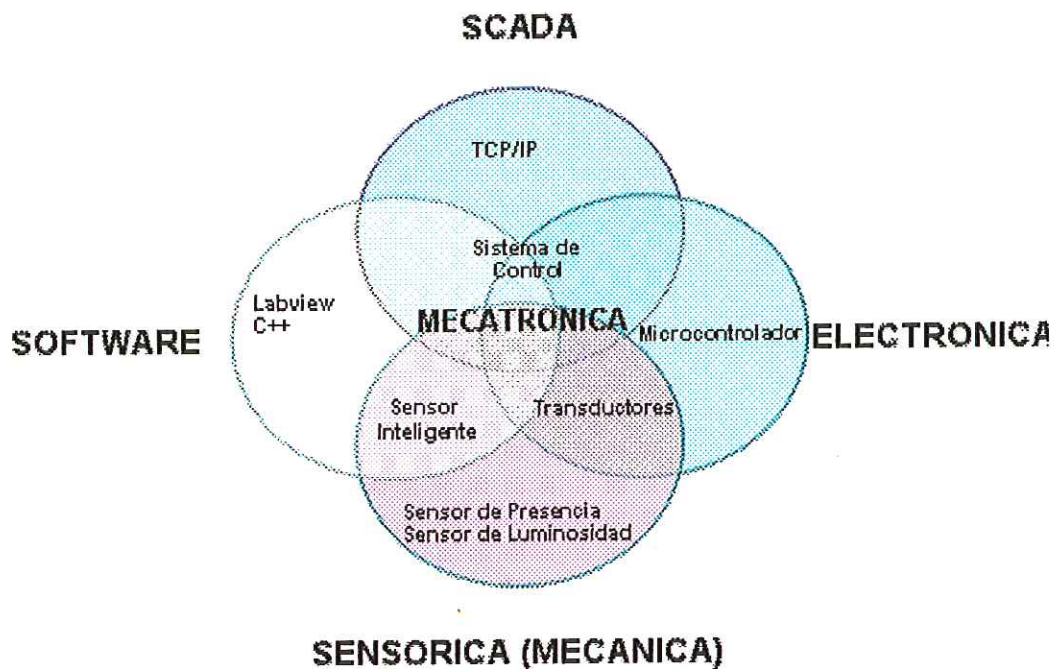
La integración tanto del diseño domótico como del diseño mecatronico se observa mejor en la tabla 3.



TABLA 3 Diseño del sistema domotico

DISEÑO MECATRONICO	DISEÑO DOMOTICO	COMPONENTES
ELECTRONICA	SEGURIDAD	<ul style="list-style-type: none"> <li>Clave de acceso</li> </ul> Plataforma de control (microcontrolador )
SENSORICA	CONFORT  GESTION ENERGETICA	<ul style="list-style-type: none"> <li>Sensor de presencia</li> <li>Sensor de iluminación</li> </ul>
SOFTWARE	COMUNICACIONES	<ul style="list-style-type: none"> <li>Bluetooth</li> <li>Programas: Labview, C++</li> </ul>

Figura 13 Integración de la Mecatronica en el Sistema Domotico

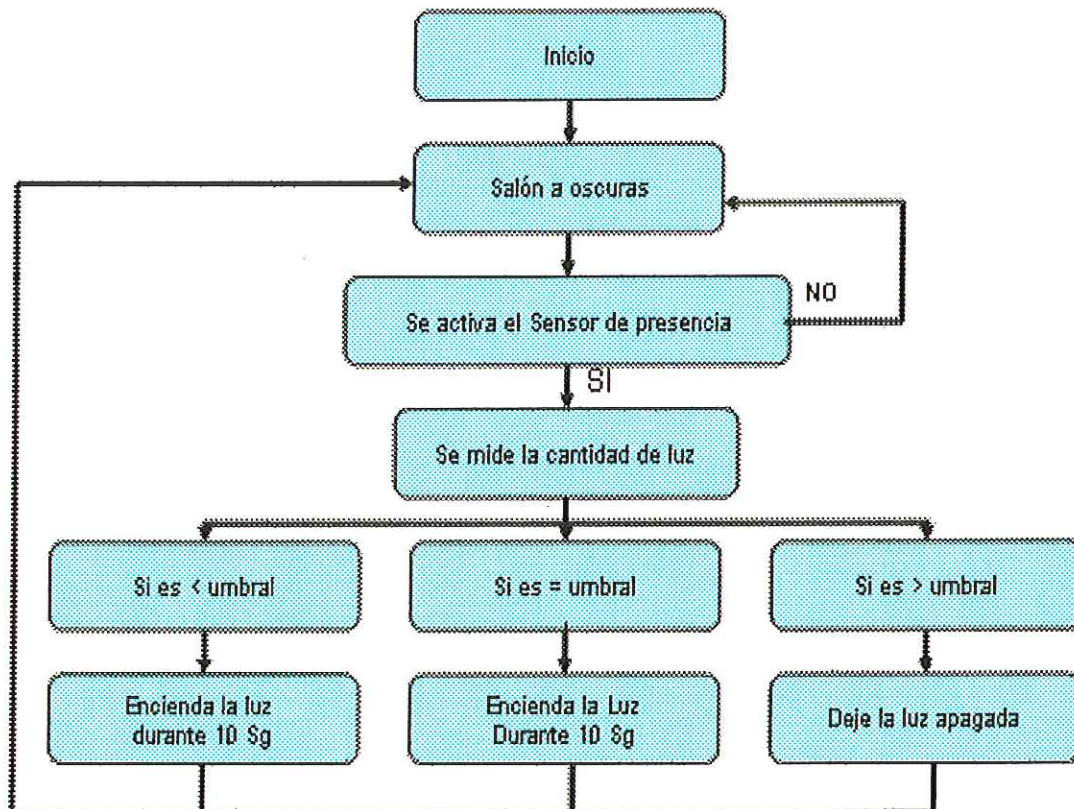
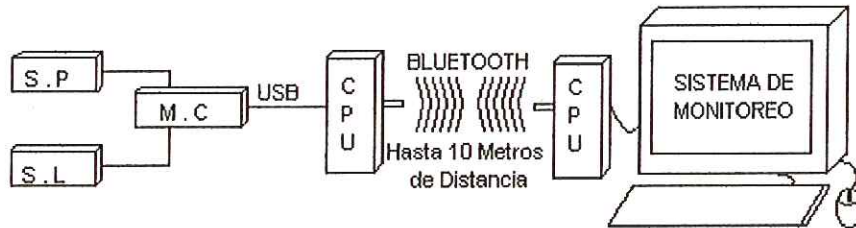


Los autores



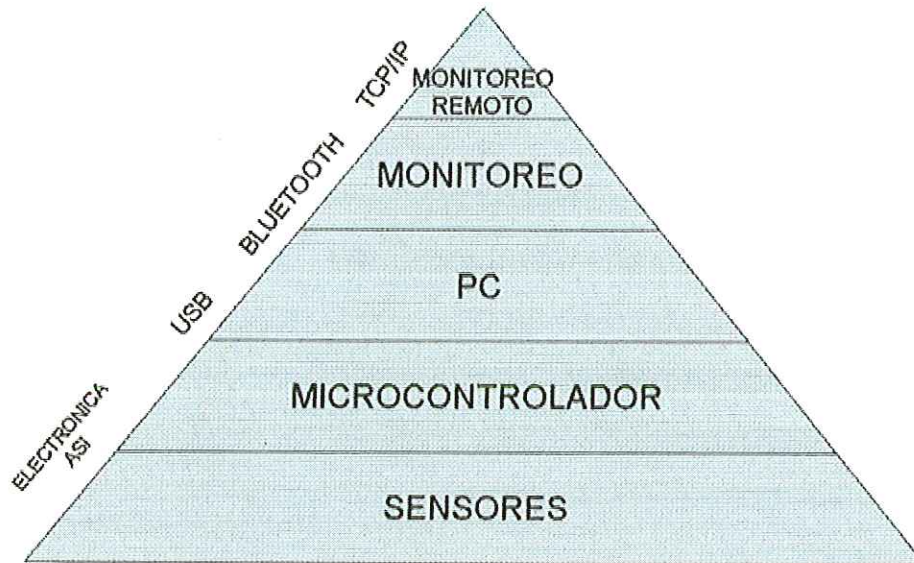
El Principal objetivo del proyecto es diseñar y construir un sistema domótico inteligente que se comunique por medio de tecnología inalámbrica Bluetooth.

Figura 14 Esquema de funcionamiento del sistema domótico



Los autores

Figura 15 Pasos a seguir para el diseño del sistema domótico



Los autores

Inicialmente el recinto a controlar estará protegido por un sistema de control de ingreso, el cual impedirá que personal no autorizado haga ingreso al recinto. Para realizar este control el sistema verificara la autorización de la persona para entrar por medio de una clave secreta, la cual le permitirá al usuario al ingresarla correctamente la apertura de la puerta de entrada, y si la persona no conoce la clave el sistema dará tres oportunidades para corregir, ante la tercera clave incorrecta el sistema deberá activar una alarma para avisar sobre la presencia de una persona no autorizada.

Ante el ingreso de una persona en el recinto, el sistema debe controlar los siguientes parámetros:

- **Control de Presencia:** El sistema deberá por medio del sensor de presencia instalado en el recinto, revisar si en el recinto hay o no personas. Si en el

recinto no hay personas, el sistema deberá seguir supervisando hasta el momento en que alguna persona ingrese.

- **Control de Iluminación:** Ante la entrada de alguna persona al recinto, el sistema deberá por medio del sensor de luminosidad medir la intensidad de luz presente en el recinto, enviar la información relacionada con la cantidad de luz presente en el recinto y de acuerdo a unos parámetros fijados en el sistema de control, graduar una intensidad de luz agradable y adecuada para el lugar.

Todos los datos provenientes de los sensores son recogidos por un microcontrolador el cual se comunica por medio de el puerto USB con un computador en donde se encuentra el programa principal de monitoreo y control del sistema. El operario o en su defecto ante la ausencia del operador el sistema estará en la capacidad de tomar las decisiones para el control del recinto.

El sistema tiene la ventaja de poder ser monitoreado por alguna persona desde otro PC o de un dispositivo que tenga comunicación Bluetooth, lo que le permitirá al encargado de controlar el recinto no estar dentro de el, para poder estar monitoreando el sistema. Dentro de la gama de productos desde lo que puede ser monitoreado el sistema están:

- Computador
- Palms
- Celular

Todos los anteriores con tecnología Bluetooth.



## 2.2. DISEÑO ELECTRONICO

En esta parte del proyecto se planteo el diseño de los diferentes circuitos que estarán encargados de recoger las señales de los sensores, los circuitos de potencia que estarán encargados de gobernar las luces del recinto, así como el circuito de control el cual será donde este insertado el microcontrolador, quien estará encargado de recoger todas las señales provenientes de los sensores, y enviarlas al sistema de control y monitoreo.

En el diseño de la plataforma de control, la pieza clave y fundamental es la correcta selección del microcontrolador, como se puede apreciar con mayor detenimiento en la sección de Selección del Microcontrolador. Debido a que el sistema va a ser para uso estudiantil y educativo, el sistema debe tener la posibilidad de poderse reprogramar tantas veces como sea posible.

El siguiente paso es seleccionar entre las diferentes opciones los sensores (Presencia e luminosidad). En el caso del sensor de presencia debemos mirar sus características, funcionamiento. Para la elección de este sensor se debe implementar un sensor que tenga una cobertura adecuada para el recinto a controlar, así como inmunidad a interferencias electromagnéticas, e inmunidad a corrientes de aire. En el caso del sensor de luminosidad, la principal característica será la de poder conectar el sensor al sistema de control.

Para el diseño del control de potencia de las luces se miraron las especificaciones de las luces a controlar, así como la señal de salida de la plataforma de control para realizar una correcta separación entre ambas señales y que a su vez presente un excelente control de potencia.



Después de concluir el diseño electrónico, se procederá a realizar el diseño del Software encargado de la programación de los microcontroladores, así como de realizar el software encargado del sistema de monitoreo del sistema domótico, y del software encargado de conectar todo el sistema con la tecnología Bluetooth.

### **2.3. COMUNICACION ENTRE EL MICROCONTROLADOR Y EL PC POR MEDIO DEL PUERTO USB**

En esta etapa de diseño, se tomará un microcontrolador, el cual será la parte más importante de la plataforma de control de nuestro sistema.

Las funciones específicas del microcontrolador serán las de recoger las diferentes señales de los sensores de presencia y de luminosidad, dichas señales deberán ser enviadas a través del puerto USB al PC.

Todo lo anterior tiene como fin, que la comunicación entre el microcontrolador y el PC se haga por medio del puerto USB y este a su vez se comunique por medio de comunicación Bluetooth a otros dispositivos.

### **2.4. SELECCIÓN DEL MICROCONTROLADOR**

En el momento de escoger un microcontrolador para nuestro proyecto se deben tener en cuenta diferentes factores que pueden repercutir en el buen o mal funcionamiento del sistema. Los factores que debemos tener en cuenta son:

- Documentación Existente
- Herramientas de desarrollo disponibles
- Costo
- Características Técnicas (Tipo de Memoria, Interrupciones, etc.)
- Fabricantes

Antes de seleccionar el microcontrolador es importante analizar los requisitos de la aplicación:

Procesamiento de datos: Debemos tener en cuenta que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso debemos asegurarnos de seleccionar un dispositivo suficientemente rápido para ello. Por otro lado, se debe tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits•

Entrada Salida: Para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.

Consumo: Algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la

activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.

- Memoria: Para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

Ancho de palabra: el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte. Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

Diseño de la placa: la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

Como anteriormente lo explicamos el sistema debe comunicarse por puerto USB, ya que este permite transferencia de datos mas seguras, rápidas y de fácil instalación. Al evaluar las diferentes alternativas sobre microcontroladores existentes en el mercado, se decide por escoger la rama de microcontroladores de Microchip, todo esto respaldado por el bajo costo, alta calidad, flexibilidad, y una razón muy importante es la documentación e información existente sobre este tipo



de microcontroladores. Dentro de la familia de microcontroladores de Microchip, hay varios que tienen las especificaciones necesarias tanto de hardware como de software para el desarrollo del proyecto, entre ellas estaban el pic16c745 y la familia de microcontroladores pic18fxxxx. Como una de las principales finalidades de este proyecto es convertirse en un sistema didáctico, se escoge el PIC18F2550 ya que este tiene la posibilidad de ser reprogramado alrededor de hasta 100.000 veces y cuenta con una retención de datos de 40 años.

#### **2.4.1. MICROCONTROLADOR PIC18F2550<sup>8</sup>**

Este es un microcontrolador PIC flash con interfase USB 2.0 'Full Speed' y 48 MHz de frecuencia de oscilación para un rango de transferencia de datos de 12 Mbps. Estas características, combinadas con una gran variedad de periféricos y un avanzado control de alimentación con tecnología nanoWatt, proporcionan una solución completa para trabajos con USB.

Las características de la interfase USB 2.0 'Full Speed' incluyen un transmisor/receptor en el dispositivo y un puerto paralelo para transferir datos directamente hacia periféricos externos con una mínima carga para la CPU. Otras características para la familia son sus 24 o 32 kbytes de memoria flash mejorada autoprogramable, soportando actualizaciones vía puerto USB. Con la tecnología flash, de Microchip, PEEC, celdas PMOS eléctricamente borrables, los dispositivos proporcionan un elevado número de ciclos de escritura/borrado, 100.000, y más de 40 años de retención de datos.

---

<sup>8</sup> Tomado de MICROCHIP, PIC 18F2455/2550/4550. Data Sheet



Este microcontrolador también incluye otras características como son: 2 kbytes de RAM, de los cuales 1 kbyte puede ser dedicado al buffer USB, 256 bytes de memoria de datos EEPROM, módulo EUSART que soporta interfase serie RS232, RS485 y LIN, puerto SSP de comunicaciones serie I2C™ y SPI™, convertidor A/D de 10 bit de resolución con más de 12 canales, 2 comparadores analógicos, módulo Captura/Comparación/PWM con 16 bit de resolución, 4 temporizadores (3 de 16 bit y uno de 8 bit), circuito detector de bajo voltaje, reset por caída de tensión programable y mejoras en la depuración en circuito con tres puntos de ruptura.

Una de las consideraciones claves para aplicaciones USB es la disponibilidad de soporte firmware. Este microcontrolador ofrece un amplio conjunto de librerías para las aplicaciones con las "clases" más comunes, incluyendo la interfase humano (HID), dispositivo de comunicación (CDC) y que se pueden personalizar para el cliente. También está publicando notas de aplicación que ilustran como reemplazar un puerto serie con una conexión USB 'Full Speed' en pocos y fáciles pasos. En la tabla 4 podemos ver las principales características del PIC18F2550.

TABLA 4 Principales características del PIC18F2550

<b>PIC18F2550</b>	
<b>Frecuencia de Operación</b>	48 Mhz
<b>Memoria de Programa</b>	32.768(Bytes)
<b>Memoria de Programa</b>	16.384 Instrucciones
<b>Memoria de Datos</b>	2.048 (Bytes)
<b>Interrupciones</b>	19
<b>I/O Puertos</b>	A,B,C,€
<b>Temporizadores</b>	4
<b>Comunicación Serie</b>	MSSP, EUSART
<b>USB</b>	1
<b>10 Bit Conversor Analogo-Digital</b>	10 Canales
<b>Comparadores</b>	2
<b>Resets</b>	POR, BOR, Reset Instructions, Pila Llena, Pila Vacía
<b>Programable Low Voltaje Detección</b>	Si
<b>Instrucciones</b>	75 Instrucciones 83 Extendidas
<b>Encapsulados</b>	28 pin PDIP 28 pin SOIC

## 2.5. COMUNICACIÓN VIA PUERTO USB

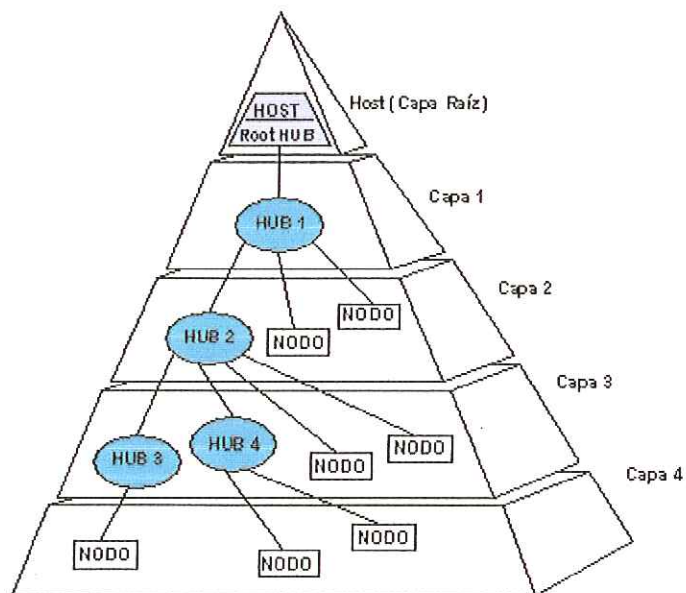
Como anteriormente mencionamos, el microcontrolador deberá comunicarse con el PC por medio del puerto USB, para esto es importante conocer el funcionamiento, manejo y en general todo lo relacionado con el puerto USB.

### 2.5.1. CONEXIÓN USB<sup>9</sup>

El puerto USB es un bus basado en el paso de un testigo, semejante a otros buses como los de las redes locales en anillo con paso de testigo y la red FDDI (Fiber Distributed Data Interface). El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo, responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.

Como se muestra en la figura 15 USB emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran más abajo en la pirámide sufran retardo.

Figura 16 Estructura de capas del bus USB



Tomado de: <http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

<sup>9</sup> Tomado de: <http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no hay retardo en el envío de un paquete de datos hacia capas inferiores.

El sistema de bus serie universal USB consta de tres componentes:

- Un controlador
- Hubs o concentradores
- Periféricos.

**El controlador.** Reside dentro del PC y es responsable de las comunicaciones entre los periféricos USB y la CPU del PC. El controlador es también responsable de la admisión de los periféricos dentro del bus, ya sea una conexión o una desconexión. Para cada periférico añadido, el controlador determina su tipo y le asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo. Si se producen errores durante la conexión, el controlador lo comunica a la CPU, la cual se lo transmite al usuario. Una vez se ha producido la conexión correctamente, el controlador asigna al periférico los recursos del sistema que éste precise para su funcionamiento.

El controlador es también responsable del control de flujo de datos entre el periférico y la CPU. Los concentradores o hubs son distribuidores inteligentes de datos y alimentación, y hacen posible la conexión a un único puerto USB de 127 dispositivos. De una forma selectiva reparten datos y alimentación hacia sus puertas descendentes y permiten la comunicación hacia su puerta de retorno o ascendente. Un hub de 4 puertos, por ejemplo, acepta datos del PC para un periférico por su puerta de retorno o ascendente y los distribuye a

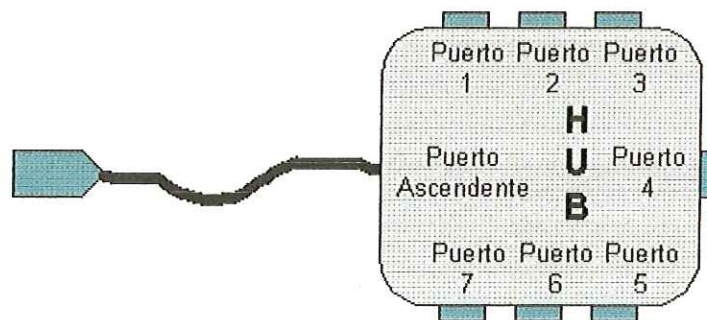


las 4 puert as descendentes si fuera necesario.

Los concentradores también permiten las comunicaciones desde el periférico hacia el PC, aceptando datos en las 4 puert as descendentes y enviándolos hacia el PC por la puerta de retorno.

Además del controlador, el PC también contiene el concentrador raíz. Este es el primer concentrador de toda la cadena que permite a los datos y a la energía pasar a uno ó dos conectores USB del PC, y de allí a los 127 periféricos que, como máximo, puede soportar el sistema. Esto es posible añadiendo concentradores adicionales. Por ejemplo, si el PC tiene una única puerta USB y a ella le conectamos un hub o concentrador de 4 puert as, el PC se queda sin más puert as disponibles. Sin embargo, el hub de 4 puert as permite realizar 4 conexiones descendentes. Conectando otro hub de 4 puert as a una de las 4 puert as del primero, habremos creado un total de 7 puert as a partir de una puerta del PC. De esta forma, es decir, añadiendo concentradores, el PC puede soportar hasta 127 periféricos USB. La mayoría de los concentradores se encontrarán incorporados en los periféricos. Por ejemplo, un monitor USB puede contener un concentrador de 7 puert as incluido dentro de su chasis. El monitor utilizará una de ellas para sus datos y control y le quedarán 6 para conectar allí otros periféricos.

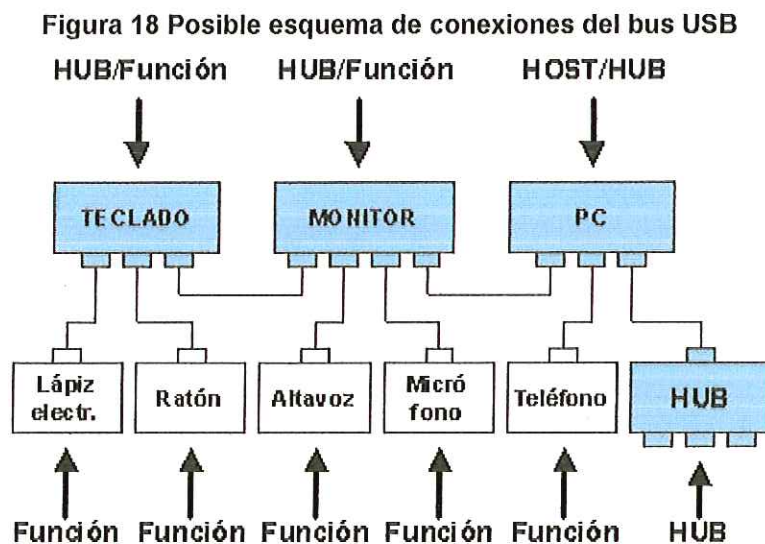
Figura 17 Esquema de un Concentrador



Tomado de: <http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

**Periféricos:** USB soporta periféricos de baja y media velocidad. Empleando dos velocidades para la transmisión de datos de 1.5 y 12 Mbps. Los periféricos de baja velocidad tales como teclados, ratones, joysticks, y otros periféricos para juegos, no requieren 12 Mbps. Empleando para ellos 1.5 Mbps, se puede dedicar más recursos del sistema a periféricos tales como monitores, impresoras, modems, scanners, equipos de audio, etc, que precisan de velocidades más altas para transmitir mayor volumen de datos o datos cuya dependencia temporal es más estricta.

En la figura 18 se puede ver cómo los hubs proporcionan conectividad a toda una serie de dispositivos periféricos.

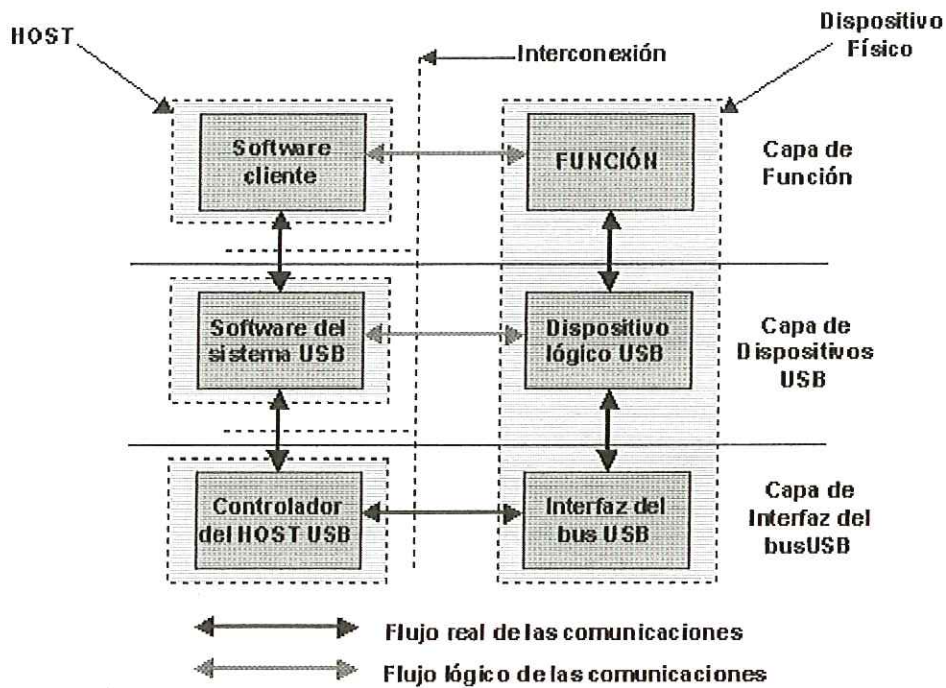


Tomado de: <http://www.monografias.com/trabajos11/usbmemb/usbmemb.shtml>

## 2.5.2. DIAGRAMA DE CAPAS DEL USB

La información entre las diferentes capas a nivel real y a nivel lógico.

Figura 19 Capas del sistema de comunicaciones USB



Tomado de: <http://www.monografias.com/trabajos11/usbmemb/usbmemb.shtml>

En la figura 19 está materializada la conexión entre el controlador anfitrión o host y un dispositivo o periférico. Este está constituido por hardware al final de un cable USB y realiza alguna función útil para el usuario. El software cliente se ejecuta en el host y corresponde a un dispositivo USB. Se suministra con el sistema operativo o con el dispositivo USB. El software del sistema USB, es el que soporta USB en un determinado sistema operativo. Se suministra con el sistema operativo independientemente de los dispositivos USB o del software cliente.

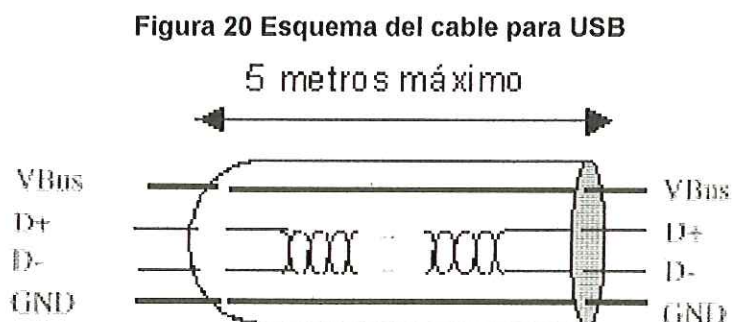
El controlador anfitrión USB está constituido por el hardware y el software que permite a los dispositivos USB ser conectados al anfitrión. Como se muestra en la figura 18, la conexión entre un host y un dispositivo requiere la interacción entre las capas. La capa de interfaz de bus USB proporciona la conexión física entre el host y el dispositivo. La capa de dispositivo USB es la que permite que el software del sistema USB realice operaciones genéricas USB con el dispositivo.



Las capas de función y los dispositivos USB tienen cada una de ellas una visión de la comunicación lógica dentro de su nivel, aunque la comunicación entre ellas se hace realmente por la capa de interfaz de bus USB.

### 2.5.3. CABLES Y CONECTORES

USB transfiere señales y energía a los periféricos utilizando un cable de 4 hilos, apantallado para transmisiones a 12 Mbps y no apantallado para transmisiones a 1.5 Mbps. En la figura 20 se muestra un esquema del cable, con dos conductores para alimentación y los otros dos para señal, debiendo estos últimos ser trenzados o no según la velocidad de transmisión.



Tomado de: <http://www.monografias.com/trabajos11/usbmemb/usbmemb.shtml>

El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG<sup>10</sup>, mientras que el de los conductores de señal es de 28 AWG. La longitud máxima de los cables es de 5 metros.

<sup>10</sup> **AWG**: Medida de los conductores eléctricos, utilizada especialmente por los norteamericanos. Un 1 AWG equivale a 1 mm de diámetro.



## **2.6. SENSÓRICA**

Una de las principales funciones del sistema es la medición de dos diferentes parámetros como son: Detección de presencia y cantidad de luz en el recinto a controlar.

Para poder realizar estas dos labores se necesitan dos diferentes sensores, los cuales independientemente estén monitoreando los diferentes parámetros, y estos datos deben ser enviados a la central de control, en nuestro caso el microcontrolador, el cual esta conectado a la central de monitoreo, el PC.

### **2.6.1. CONTROL DE PRESENCIA E ILUMINACIÓN**

En primer lugar debemos medir constantemente si dentro del recinto a controlar hay alguna persona, para ello usaremos un sensor de movimiento infrarrojo (PIR), el cual será el encargado como anteriormente se menciona de enviar constantemente los datos a la central de control sobre la presencia o no presencia de personal en el recinto.

Acto seguido, ante la presencia de alguna o algunas personas en el lugar, se debe medir la intensidad de luz, para poder hacer un control de la cantidad de luz que se necesita. Para la segunda función la medición de luminosidad en el recinto usaremos una fotorresistencia (LDR).

### **2.6.2. MEDICIÓN DE PRESENCIA**

Los detectores de presencia en la mayoría de los casos disponen de un elemento sensor, el cual tiene como función analizar la alteración de alguna magnitud física. Esta alteración es detectada por un circuito electrónico asociado que opera un contacto normalmente cerrado, que al abrirse envía la información del cambio a la central, la que se encargara de accionar alguna alarma ó seguir un procedimiento previamente programado.

Estas advertencias solo son posible si el sensor ó detector esta correctamente localizado, instalado y mantenido. Por ello es importante instalar detectores en cada recinto que se desee controlar. Los detectores no se deben instalar directamente sobre una cocina, ni en la cercanía de extractores de aire, puertas o ventanas, ni en lugares que tengan temperaturas elevadas. Debe tenerse cierto cuidado en áreas donde haya presencia de animales o corrientes de aire, ya que esto podría dar lugar a falsas alarmas.

### **2.6.3. ELECCIÓN DEL SENSOR DE MOVIMIENTO**

Para la elección de este sensor, se evaluaron las diferentes características que poseen los sensores de presencia, dentro de las mas importantes están:

- Inmунidades a corrientes de aire: Esto es muy importante ya que nuestro sistema estará implementado en lugares donde haya aire acondicionado, o donde puede haber corrientes de aire provenientes de alguna ventana. Esto en algunas ocasiones genera falsas alarmas, lo que en nuestro caso haría que el sistema encendiera las luces del recinto por error.

- Sistema de ajuste automático de la sensibilidad: Es importante que el sensor que sea usado en el proyecto, cuente con un sistema que automáticamente gradúe la sensibilidad del sensor dependiendo del entorno en donde se trabaje.

Las alternativas que encontramos a la hora de seleccionar el sensor de presencia, fueron el sensor de presencia Referencia: AD4080, y el sensor de presencia DS940PT. La primera presenta entre sus características, una detección de cuerpos de máximo 4 metros de distancia, mientras que el segundo sensor cuenta con un sistema de detección de hasta 12 metros, teniendo una mayor área de cobertura, lo cual hace al sistema mucho mas optimo. Otra de las ventajas del sensor de referencia DS940PT es su sistema de ajuste automático, de sensibilidad, facilitando su conexión a cualquier entorno, ya que cada vez que se instale el mismo, gradúa el nivel de sensibilidad adecuado, y aunque el sensor AD4080 tiene un costo menor al otro, el segundo posee muchas más ventajas respecto al otro.

Por las anteriores razones se decidió utilizar el Sensor de Presencia PIR DS940PT de la Marca Bosch, del cual se hará mas adelante una explicación de sus características técnicas.

#### **2.6.4. SENSOR DE MOVIMIENTO PASIVO (PIR).**

Esta clase de sensores trabajan mediante la detección de la radiación infrarroja emitida por los cuerpos vivos ubicados dentro del campo de operación del sensor. El mismo tiene una lente interna especial que concentra los rayos infrarrojos en su foco, donde se instala el sensor, propiamente dicho. Esta lente no enfoca todos los



rayos que inciden en el sensor, presentando zonas o ángulos de sombra que se intercalan con zonas de detección. De esta manera cuando un cuerpo caliente se mueve, se producirá un cambio en la distribución de zonas de sombra y detección de radiación, lo que produce una ligera modificación que es discriminada por el sensor infrarrojo, cuyo circuito asociado envía al control de señal la información correspondiente a la activación del sensor por una persona.

El funcionamiento óptimo se produce cuando el cuerpo caliente se desplaza de forma transversal, atravesando el haz de ángulos de sombra y detección, y el mejor índice de detección ocurre cuando el objeto se desplaza totalmente de frente hacia el detector, ya que de este modo, no se modifica apreciablemente la distribución de haces y la detección se produce de forma más lenta.

#### 2.6.5. SENSOR DE PRESENCIA DS940PT<sup>11</sup>

Figura 21 Vista de un sensor de presencia REF: DS940PT



Tomado de: [www.bosch.com.co](http://www.bosch.com.co)

<sup>11</sup> Tomado de [www.bosch.com.co](http://www.bosch.com.co)



Los detectores DS940P y DS940PT son detectores pasivos infrarrojos para montaje sobre pared. Utilizan lentes de Fresnel facetadas para lograr una capacidad de detección claramente superior. La facilidad de instalación y la flexibilidad de sus opciones de montaje proveen detección de avanzada en todo momento. Los detectores DS940P y DS940PT otorgan alta inmunidad a falsos disparos ocasionados normalmente por mascotas o animales de hasta 13,6 kgs (30 lbs.). El sistema de compensación térmica de estos detectores, permite su utilización en condiciones muy extremas.

2.6.5.1. PROCESAMIENTO DE SEÑALES. El Procesamiento de Inicialización Patentado First Step (FSP) provee un ajuste automático durante la instalación, lográndose así una respuesta instantánea ante cualquier presencia humana sin sacrificar la inmunidad contra falsas alarmas. Su sensibilidad está basada en el ajuste de la amplitud, polaridad, pendiente y período de la señal, por lo que el FSP le elimina al instalador la necesidad de ajustar un nivel de sensibilidad en cada aplicación. Esta nueva tecnología garantiza el óptimo funcionamiento de captura a cualquier altura de instalación entre 2,2 y 2,7 metros.

Dentro de las principales ventajas de este sensor están:

**Características de Prueba:** LED de alarma de alta luminosidad que puede deshabilitarse después de la instalación.

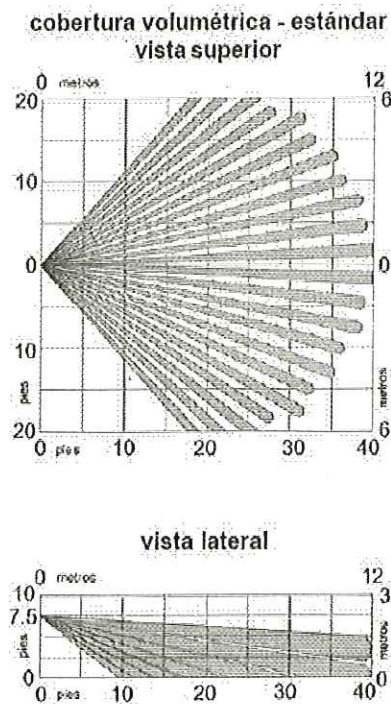
**Inmunidad a Mascotas:** Ofrece inmunidad contra 1 o 2 animales con un peso total de 13,6 Kg.

**Inmunidad contra Corrientes de Aire e Insectos:** Sus ópticas selladas proveen inmunidad contra corrientes de aire e insectos.

**Compensación de Temperatura:** Automáticamente monitorea la temperatura ambiente y ajusta el procesador de señales para mantener una detección estable en rangos de temperatura críticos.

2.6.5.2. **COBERTURA.** La cobertura de un sensor es un dato importante a la hora de realizar el diseño de un sistema domótico. Debemos tener en cuenta el espacio a controlar así como las diferentes entradas que tiene, entre otros parámetros. El sensor nos proporciona una cobertura en la cual su efectividad es confiable, ya con ese parámetro definido podemos realizar el diseño correspondiente a la cantidad de sensores que necesitamos, así como su ubicación. En la figura 22 podemos observar la cobertura del sensor de presencia DSP940PT.

**Figura 22 Cobertura del sensor DS940PT**



Tomado de: [www.bosch.com.co](http://www.bosch.com.co)

## 2.7. MEDICION DE LUMINOSIDAD

Para seleccionar la mejor opción existían varias alternativas, la primera de ellas era el uso de un Luxómetro Digital. El luxómetro sirve para realizar la medición de luminosidad en un recinto. Este elemento toma la medida y da un valor digital. Dentro de sus principales características técnicas están:

- Medida de nivel de luminosidad de 0.01 lux a 50000 lux
- Display LCD 3 ½ dígitos

Sus especificaciones técnicas son:

- Rango de medida: 200, 2000, 20000lux (20000lux lectura rango x10) y 50000lux (50000lux lectura rango x100)
- Precisión:
  - < 10000lux:  $\pm 5\%$  de la lectura + 10 dígitos
  - > 10000lux:  $\pm 10\%$  de la lectura + 10 dígitos
- Repetibilidad:  $\pm 2\%$
- Características de la temperatura:  $\pm 0.1\%/^{\circ}\text{C}$
- Fotodetector: 1 fotodiodo de silicio con filtro
- Alimentación: 1 x pila A23 de 12V (incluida)
- Velocidad de la medición: 1.5 veces por segundo, nominal
- Temperatura de almacenamiento: de  $-10^{\circ}\text{C}$  a  $+60^{\circ}\text{C}$
- Dimensiones fotodetector: 115 x 60 x 27mm
- Dimensiones: 188 x 64.5 x 24.5mm
- Peso: 160g

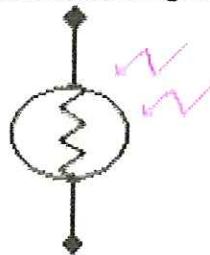
La otra opción existente es el uso de una fotorresistencia. Esta ultima también cumple con las especificaciones técnicas de nuestro proyecto, ya que permite el sensado constante de la luminosidad dependiendo de la cantidad de luz que incida sobre ella.

Teniendo estas dos alternativas, y mirando sus características técnicas, no se uso el luxometro ya que no se encuentra en el mercado colombiano, lo que demoraria mucho su adquisición. El otro inconveniente por el cual se desecho esta alternativa, es su elevado costo, ya que cada uno cuesta alrededor de \$ 650.000 pesos mas gastos de envío. Mientras que la fotorresistencia es un dispositivo fácil de adquirir, y que en nuestro proyecto se acopla de manera correcta, además ante un inesperado daño, es de fácil adquisición y su precio esta alrededor de \$ 2.000 pesos.

### 2.7.1. FOTORESISTENCIA (LDR)

La LDR es una resistencia que varía su valor dependiendo de la cantidad de luz que la ilumine. Es fabricada con materiales de estructura cristalina, los cuales deben ser materiales fotoconductores. Los cristales mas utilizados son el Sulfuro de Cadmio y el Seleniuro de Cadmio.

Figura 23 Representación grafica de un LDR



Tomado del programa Circuit Maker



Los valores de una fotorresistencia (LDR) cuando esta totalmente iluminada y cuando esta totalmente a oscuras varia, puede medir de 50 ohmios a 1000 ohmios (1K) en iluminación total y puede llegar a medir de 50K (50000 ohmios) a varios megaohmios cuando este a oscuras.

### 2.7.2. FUNCIONAMIENTO DE UNA LDR

La conductividad eléctrica en un material depende del número de portadores en la banda de conducción. En un semiconductor, a baja temperatura la mayor parte de sus electrones están en la banda de valencia y se comporta casi como aislante. Pero al aumentar la temperatura, y con ella la agitación de los electrones, dado que las bandas de valencia y de conducción están próximas (a diferencia de lo que sucede en un aislante), aumentando la conductividad. Si el semiconductor está dopado, este salto es aún más fácil.

La energía necesaria para producir el salto puede venir de otras formas externas además del calor, como pueden ser una radiación óptica o una tensión eléctrica. En el caso de la radiación óptica, su energía, E y frecuencia, f, están relacionadas mediante la expresión (1.1):

$$E=hf \quad (1.1)$$

Donde  $h=6.62 \times 10^{-34} \text{ Ws}^2$  es la constante de Planck. Entonces, si la radiación tiene energía suficiente para permitir el salto de los electrones de una a otra banda, pero sin exceder el umbral o fotoconductor, y a mayor iluminación mayor será la conductividad. Si se excediera dicho umbral, se tendría efecto fotoeléctrico externo. En el caso de un conductor, la conductividad es de por sí tan alta que el cambio debido a la incidencia de la radiación apenas se nota.

En la tabla 5 se exponen los valores de la anchura de banda prohibida, diferencia de niveles de energía entre las bandas de conducción y de valencia, para distintos semiconductores y también la longitud de onda máxima de radiación para poder producir efecto fotoeléctrico.

TABLA 5 Anchura de banda prohibida, en electrón-voltios, y longitud de onda máxima, en micras, para diversos semiconductores intrínsecos

<b>MATERIAL</b>	<b>ANCHURA DE LA ANDA PROHIBIDA</b>	<b>LONGITUD DE LA ONDA MAXIMA</b>
SZn	3.60	0.345
SCd	2.40	.52
SeCd	1.80	0.69
TeCd	1.50	0.83
Si	1.12	1.10
Ge	0.67	1.85
SPb	0.37	3.35
Asin	0.35	3.54
Te	0.33	3.75
TePb	0.30	4.13
SePb	0.27	4.58
Sbin	0.18	6.90

La relación entre la energía de los fotones y la longitud de onda de la radiación,  $\lambda$ , es (1.2):

$$\lambda = ch/E \quad (1.2)$$

Donde  $c$  es la velocidad de la luz. Si  $E$  se expresa en electronvoltios, la anterior expresión se reduce a (1.3):

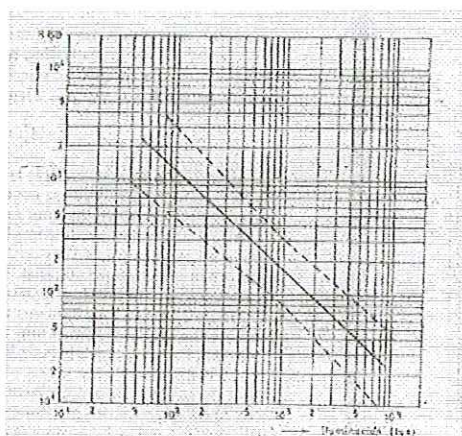
$$\lambda(\mu\text{m})=1.24/(E(\text{eV})) \quad (1.3)$$

La relación entre la resistencia R de un fotoconductor y la iluminación, E (densidad superficial de energía recibida expresada en lux), es fuertemente no lineal. Un modelo simple es (1.4):

$$R=AE^{-\alpha} \quad (1.4)$$

Donde A y  $\alpha$  dependen del material y las condiciones de fabricación. Para el SCd, por ejemplo,  $0.7 < \alpha < 0.9$ .

**Figura 24 Característica resistencia-iluminación de una LDR (Documentación Philips)**



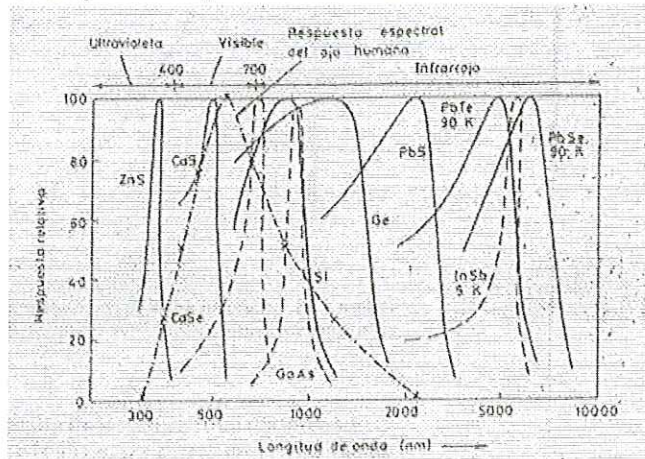
En la figura 24 se presenta gráficamente esta relación para un modelo concreto de SCd. Como se puede observar, además de la no linealidad, que la relación entre la resistencia en presencia de luz y la resistencia en la oscuridad es muy alta (superior a  $10^4$ ).

Otro factor que se debe considerar en las LDR es que su constante de tiempo de subida al iluminar y su constante de tiempo de caída o extinción, son distintas y función tanto del material como del nivel de iluminación. La constante de tiempo al



iluminar es mucho menor que al extinguir la luz. La primera suele expresarse en milisegundos y la segunda kilohmios/segundo. Son sensibles también a la temperatura, que afecta su sensibilidad a la radiación incidente en un grado tanto más alto cuanto menor sea la iluminación. La temperatura es también causa del denominado ruido térmico, que es manifiesta como fluctuaciones de corrientes cuando se aplica una tensión a la fotorresistencia para poder medir su valor.

**Figura 25 Respuesta espectral de distintos fotoconductores intrínsecos**



Su respuesta espectral es estrecha, tal como se indica en la figura 25 para diversos materiales. La elección de estos debe hacerse, pues, en función de la longitud de onda a detectar, teniendo en cuenta, además, que los materiales deben ser transparentes a las longitudes de onda de interés, pues de lo contrario se producirá una excesiva reflexión superficial.

En la zona visible (0.38 a 0.75  $\mu\text{m}$ ) y del infrarrojo muy cercano (0.75 a 1.4  $\mu\text{m}$ ) se emplean compuestos de cadmio (SCd, SeCd, TeCd). En la zona del infrarrojo cercano (1.4 a 3  $\mu\text{m}$ ) se emplean compuestos de plomo (SPb, SePb, TePb). En la zona del infrarrojo medio (3 a 14  $\mu\text{m}$ ) y lejano (hasta 1mm) se emplean compuestos de indio (SbIn, ASIn), telurio y aleaciones de telurio, cadmio y mercurio, así como silicio y germanio dopados.



Los elementos provistos para longitudes de onda grande (baja energía) deben mantenerse a temperatura baja, mediante efecto Peltier inverso o en criostatos, para reducir el ruido térmico. En consecuencia, no se comercializan como resistores ordinarios.

Los fotoconductores más comunes, utilizables a temperatura ambiente, son el SCd, SPb, y SePb. Los tiempos de respuesta van desde los 100 ms de algunos modelos de SCd hasta los 2  $\mu$ s de algunos de SePb. La tensión máxima que aceptan sin iluminación puede ir de 100 a 600 V, y la disipación permitida a 25°C, de 50mW a 1W.

## **2.8. DISEÑO DE SOFTWARE**

Dentro del diseño del software están contenidos todos los programas que se deben utilizar para el funcionamiento de todo el proyecto. El primer software a diseñar es el que se usa para programar el microcontrolador, este deberá contener todas las instrucciones para que la plataforma de control realice todas sus funciones.

El siguiente paso es realizar el diseño del software de monitoreo, este deberá recoger las señales enviadas por el microcontrolador a través del puerto USB, y visualizarlas en el computador. Para la elección del software necesario se evaluaron los requerimientos del proyecto, como posibilidad de recoger datos por el puerto USB, otra importancia será la de poder programar en el un entorno de visualización lo mas completo y amigable para el usuario. También hay que realizar el diseño de un software que permita conectar todo el sistema domótico a

la tecnología Bluetooth, lo que permitirá poder monitorear los diferentes datos desde cualquier dispositivo que cuente con esta tecnología.

## **2.9. SOFTWARE DE COMUNICACIÓN ENTRE EL PC Y LA PLATAFORMA DE CONTROL**

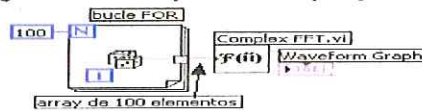
Para lograr una correcta comunicación entre el sistema domótico y el usuario se deberá utilizar un herramienta de programación que nos permita crear un sistema capaz de controlar los diferentes parámetros a controlar en este proyecto. Este entorno de comunicación deberá ser una herramienta de fácil uso para todos aquellos que deseen manejar el sistema.

### **2.9.1. SELECCIÓN DEL SOFTWARE**

Dentro de los diferentes programas que manejan los tipos de opciones que queremos usar, la herramienta que mas se adapta a nuestras necesidades es Labview.

2.9.1.1. LABVIEW. Es una herramienta gráfica de test, control y diseño. El lenguaje usado para su programación se llama lenguaje G. Fue creado por National Instruments (1976) para funcionar sobre maquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para plataformas Windows, Unix, Mac, Linux y su última versión es la 7.1.

Figura 26 Diagrama de bloques de un programa en LabVIEW



Tomado de: [www.ni.com](http://www.ni.com)

Los programas realizados con LabVIEW se llaman VI (Virtual Instrument). Uno de sus principales usos es el control de instrumentos. El lema de LabVIEW es “La Potencia está en el software”.

Principales usos:

- Adquisición de datos
- Control de instrumentos
- Automatización Industrial
- Diseños de control

Su principal característica es la facilidad de uso, ya que no se necesita de grandes conocimientos para realizar un programa avanzado, que para realizarlo en otros lenguajes se necesitaría de conocimientos avanzados de programación. Otra de sus grandes ventajas es la rapidez con que se pueden crear los diferentes programas, pero con LabVIEW también se pueden crear miles de VIS (páginas de código) para aplicaciones muy complejas, programas de automatización de decenas de miles de puntos de entradas/salidas, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación.

Presenta facilidades para el manejo de:

- Interfaces de comunicación
  - Puerto Serie
  - Puerto Paralelo
  - GPIB
  - PXI
  - VXI
  - TCP/IP
  - IrDA
  - **Bluetooth**
  - **USB**
  - Entre otros
  
- Capacidad de interactuar con otras aplicaciones
  - Dll
  - ActiveX
  - Matlab
  - Simulink
  - Entre otros
  
- Herramientas para el procesado digital de señales
- Visualización y manejo de gráficas con datos dinámicos
- Adquisición y tratamiento de imágenes
- Control de movimiento
- Tiempo real
- Sincronización
- Programación de FPGAS



## **2.10. INTERACCION MICROCONTROLADOR – LABVIEW**

Para que sea posible la comunicación entre Labview y el microcontrolador se debe utilizar un CIN (Nodo de Interfaz de Código), el cual es el que hace posible que el microcontrolador envíe y reciba datos provenientes del PC a través del Labview.

### **2.10.1. CIN (NODO DE INTERFAZ DE CODIGO)**

Cuando se abre un VI en LabVIEW, automáticamente se toma el código fuente G y se compila en código de máquina. El ejecutable de LabVIEW está escrito en C, sin embargo los VIs no hacen nunca uso de ninguna capa C.

Aunque LabVIEW ejecuta el código rápidamente, en algunos casos se necesitan hacer llamadas a código externo. Existen tres razones principales para esta necesidad:

- Ellos tienen código existente que desean utilizar dentro de un VI
- Ellos necesitan comunicarse con un hardware que LabVIEW no soporta directamente
- Se necesita hacer algo que es imposible o impráctico en el contexto de un diagrama de bloques

Para transformar el código C en una forma en que LabVIEW puede reconocerlo, tenemos que engañar al compilador de C para que crea que está generando un ejecutable a partir de un archivo make. De hecho, vamos a crear código objeto que LabVIEW puede incorporar directamente dentro del código de máquina del VI.

## **2.11. COMUNICACIÓN BLUETOOTH**

### **2.11.1 CONEXIÓN ENTRE EL PC – Y OTROS DISPOSITIVOS CON TECNOLOGÍA BLUETOOTH**

Después de ser recogida toda la información por los sensores (iluminación y presencia), necesitamos comunicar los diferentes sensores y la plataforma de control con las personas que están vigilando o monitoreando el sistema de control.

Para esto se utilizara el tipo de comunicación inalámbrica Bluetooth. La cual se realizara por medio del adaptador USB – BLUETOOTH (TBW-101UB) entre PC y PC o entre PC y otro dispositivo con tecnología Bluetooth (celulares, palm's ).

### **2.11.2. ADAPTADOR USB-BLUETOOTH (TBW-101UB)**

**Figura 27 Adaptador USB - Bluetooth REF: TBW-101UB**



Tomado de: [www.trendnet.com](http://www.trendnet.com)

El Adaptador USB Bluetooth TBW-101UB de TRENDnet permite hacer conexiones inalámbricas de corto alcance entre su computadora y dispositivos permitidos Bluetooth tales como teléfonos celulares, PDAs y computadoras. Es compacto y portátil, eliminando así la necesidad de usar cables y conexiones físicas entre dispositivos electrónicos. La transmisión del Adaptador USB Bluetooth asegura protección contra interferencia y transferencia de datos segura.

## **CARACTERÍSTICAS**

- Compatible con USB 1.1
- Usa la frecuencia de banda de 2.4Ghz ISM con la frecuencia estándar Hopping Spread Spectrum (FHSS)
- Transmisión inalámbrica de alta velocidad, hasta 723Kps
- Compatible con Windows 98/ME/2000/XP
- Permite a cualquier computadora con conexiones USB convertirse en una computadora permitida Bluetooth
- Permite que computadoras se conecten a impresoras Bluetooth y dispositivos móviles que permitan la función Bluetooth (PDAs y teléfonos móviles)
- Seguridad del cifrado 128-bit integrado para comunicación entre los dispositivos Bluetooth
- Sincroniza automáticamente dispositivos en la red de área personal "Personal Area Network" (PAN) para mantener datos constantes Plug & Play e instalación sencilla
- Rango de conectividad de hasta 10 metros

## **2.12. COMUNICACIÓN DEL SISTEMA DENTRO DE UNA LAN ( LOCAL ÁREA NETWORK )**

Después de tener todo el sistema funcionando correctamente a través de la comunicación inalámbrica bluetooth, se permitirá a cualquier usuario de Internet conectado dentro del entorno de la red de área local, acceder al sistema desde cualquier punto de esta red. Esto se lograra gracias a que el programa de interacción del sistema domótico (Labview) será hecho bajo el protocolo de comunicación TCP / IP.

## **2.13. PROTOCOLO DE COMUNICACION TCP / IP**

Es un protocolo para comunicación por red de datos. El más ampliamente utilizado es el Internet Protocol Suite, comúnmente conocido como TCP / IP. Es un protocolo DARPA que proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP / IP Proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto. El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa. TCP / IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET una red de área extensa del departamento de defensa.

Al concluir el diseño electrónico y el diseño de software, el siguiente paso es el desarrollo del mismo. Para esto se hará a continuación un recuento de este proceso.



### **3. DESARROLLO**

Al igual que en capítulo de diseño, el proyecto se divide en dos partes fundamentales. La primera el diseño electrónico, y posteriormente el diseño del software que se encargara del control del microcontrolador, del sistema de monitoreo y del sistema de comunicación Bluetooth.

La primera parte a desarrollar en este sistema, es la parte electrónica. El diseño electrónico esta dividido en dos partes, la primera parte es el diseño de la plataforma de control, que como anteriormente se ha mencionado, será la encargada de recoger las diferentes señales de los sensores y comunicarse con el sistema de control y monitoreo. La segunda parte del diseño electrónico esta destinada al sistema de potencia el cual tendrá como función gobernar el sistema de luces del recinto las cuales deberán encenderse o apagarse dependiendo de las condiciones del recinto (Presencia e Iluminación).

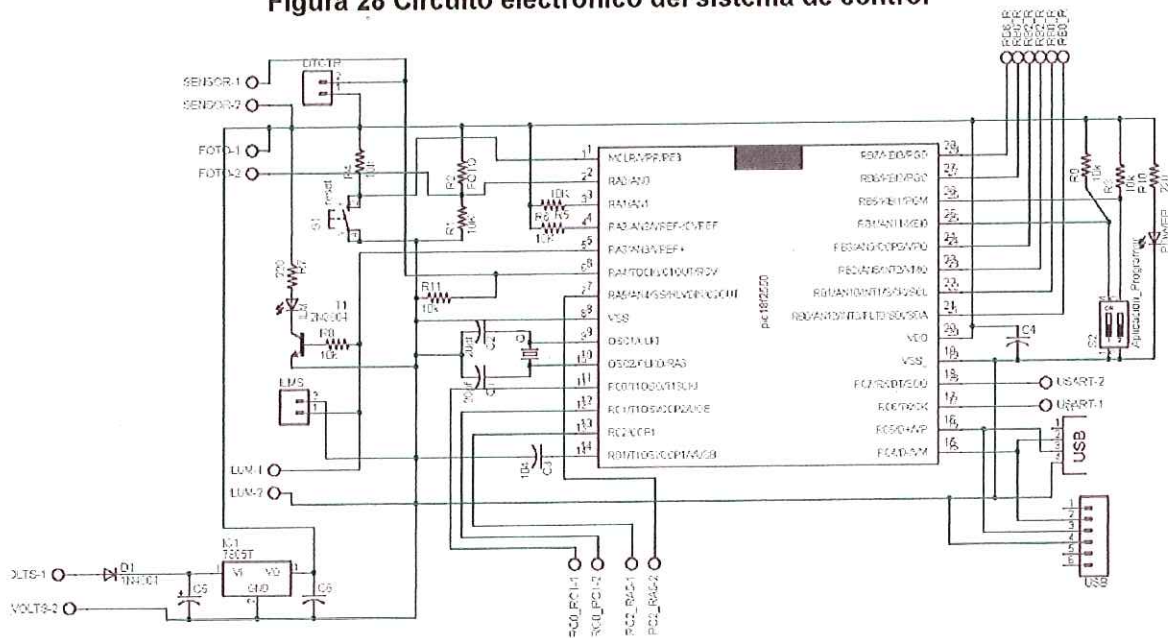
#### **3.1. CIRCUITO ELECTRÓNICO (INTERFASE SISTEMA DE CONTROL)**

Para lograr las diferentes funciones de la interfase de control, se diseño un circuito electrónico el cual tiene como pieza principal y fundamental el microcontrolador PIC18F2550. Este microcontrolador como anteriormente se menciona cuenta especialmente con un puerto de salida USB, el cual nos permite lograr una comunicación entre el sistema de control y un Computador de forma rápida y confiable. El microcontrolador tiene como función especifica recoger la señal de los diferentes sensores (Presencia e Iluminación) y deberá enviar dicha

información al sistema de monitoreo y dependiendo de los requerimientos del sistema, tomar la decisión correcta en cuanto a intensidad de luz.

Como mas adelante se explicara, el sistema esta basado en el Kit PICDEM FS USB DEMONSTRATION BOARD, para acoplar este KIT a este proyecto se tuvieron en cuenta los parámetros con los cuales viene definido este KIT, y los parámetros requeridos por el proyecto. En la figura 28 se puede observar el circuito electrónico del sistema de control. Mas adelante en el diseño de software se explicara su funcionamiento, y como se adapto este KIT al proyecto.

**Figura 28 Circuito electrónico del sistema de control**



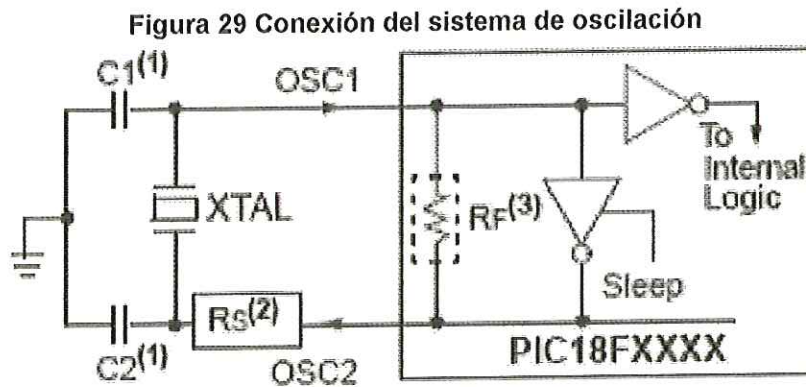
Los autores

A continuación, en la tabla 6 están detallados los diferentes elementos que conforman el circuito electrónico.

**TABLA 6 Elementos circuito electronico sistema de control**

<b>REFERENCIA</b>	<b>DESCRIPCION</b>
R1	10 K $\Omega$
R2	Fotorresistencia
R3	10 K $\Omega$
R4	10 K $\Omega$
R5	10 K $\Omega$
R6	10 K $\Omega$
R7	220 $\Omega$
R8	10 K $\Omega$
R9	10 K $\Omega$
R10	220 $\Omega$
R11	10 K $\Omega$
C1	20 Pf
C2	20 pF
C3	104 pF
T1	2N3904
S1	Swithc Reset
S2	Switch para modo de programación
LED1	Diodo Emisor de Luz
POWER	Diodo Emisor de Luz
PIC18F2550	Microcontrolador PIC18F2550
Sensor1	Sensor de Movimiento DS940PT
X1	Conector USB

Para lograr un buen rendimiento del sistema, hay que tener en cuenta las recomendaciones del fabricante del microcontrolador en cuanto a la elección y conexión del Cristal y de los condensadores, los cuales actuaran como el reloj del microcontrolador.



Tomado de Datasheet PIC18F2550 (Microchip)

En la figura 29 esta la recomendación del fabricante en cuanto a conexión de los condensadores y el cristal del microcontrolador. Para los valores correspondientes de los Capacitares y el Cristal la recomendación del fabricante es:

TABLA 7 Valores para cristal y consendadores (Microchip)

Tipo de Oscilador	Frecuencia del Cristal	Valor Capacitor	
		C1	C2
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

La resistencia RS<sup>(2)</sup> puede usarse en algunos casos para evitar sobrealimentar el cristal.

Considerando las recomendaciones del fabricante, y evaluando los requerimientos del sistema se usaron el condensador y el cristal con los siguientes valores:



- Condensadores
  - C1 (20 pF)
  - C2 (20 pF)
  
- Cristal (6 Mhz)

Para la medición de presencia se conecto el sensor de presencia (REF: DS940T) el cual se encarga de recoger la señal correspondiente a la presencia o no presencia de personas en el recinto y enviar dicha información al microcontrolador. Para el censado de Iluminación la fotorresistencia (LDR), esta conectada al sistema y como se menciona anteriormente esta dependiendo de la cantidad de luz, modifica su resistencia lo que nos permite obtener un cambio en la corriente, señal que es leída por el microcontrolador y que nos permite tener una información correcta acerca de la iluminación en el recinto a controlar.

Otra función principal del sistema de control, es la de comunicarse por medio del puerto USB con el sistema de monitoreo, para esto aprovechando la posibilidad de conectar el microcontrolador por medio del puerto USB, se acoplo al sistema un conector tipo hembra que permita la conexión entre los dos dispositivos (Plataforma de Control y PC) y con la ayuda de un cable USB lograr la correcta comunicación entre los dos sistemas.

El cable de bus USB es de 4 hilos, y comprende líneas de señal (datos) y alimentación. Existen dos tipos de cable: apantallado y sin apantallar. En el primer caso el par de hilos de señal es trenzado; los de tierra y alimentación son rectos, y la cubierta de protección (pantalla) solo puede conectarse a tierra en el anfitrión. En el cable sin apantallar todos los hilos son rectos. Las conexiones a 15 Mbps y superiores exigen cable apantallado.

En la tabla 8 se menciona el número de pines del puerto, así como su nombres, características y color.

TABLA 8 Características eléctricas de un conector USB

Pin	Nombre	Descripción	Color
1	VBUS	+ 5 V. CC	Rojo
2	D-	Data -	Azul
3	D+	Data +	Amarillo
4	GND	Tierra	Verde

Una forma de identificar que los concentradores (hubs) USB están realmente adaptados a la norma 2.0 es por su cable apantallado.

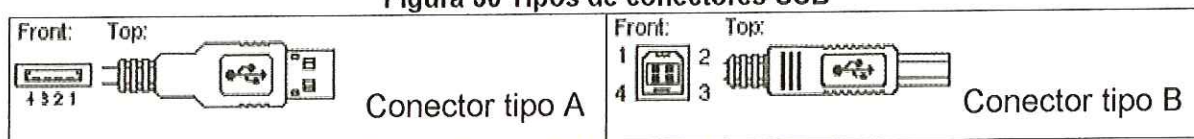
TABLA 9 Características de un cable USB

AWG	mm Ø	Longitud Maxima
28	0.321	0.81 m
26	0.405	1.31 m
24	0.511	2.08 m
22	0.644	3.33 m
20	0.812	5.00 m

Se utilizan diámetros estándar para los hilos de alimentación del bus. Para cada sección se autoriza una longitud máxima del segmento. Para la conexión se usan dos tipos de conectores, A y B. Ambos son polarizados, es decir solo pueden insertarse en una posición y utilizan sistemas de presión para sujetarse. Los de tipo A utilizan la hembra en el sistema anfitrión, y suelen usarse en dispositivos en los que la conexión es permanente (por ejemplo, ratones y teclados). Los de tipo B utilizan la hembra en el dispositivo USB (función), y se utilizan en sistemas

móviles (por ejemplo, cámaras fotográficas o altavoces). En general podemos afirmar que la hembra de los conectores A están en el lado del host (PC) o de los concentradores (hubs), mientras las de tipo B están del lado de los periféricos.

**Figura 30 Tipos de conectores USB**



Tomado de: <http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

Una vez diseñado el circuito, posteriormente se realizó el diseño del PCB, para ello y como mas adelante se explicara, con ayuda del Software Eagle se realizara este diseño.

### 3.2. DISEÑO DE PCB

Para una mejor ilustración de los circuitos y para garantizar que sea un circuito totalmente practico, se realizo el diseño de la placa donde se soldaran los diferentes elementos que hagan parte de los circuitos electrónicos.

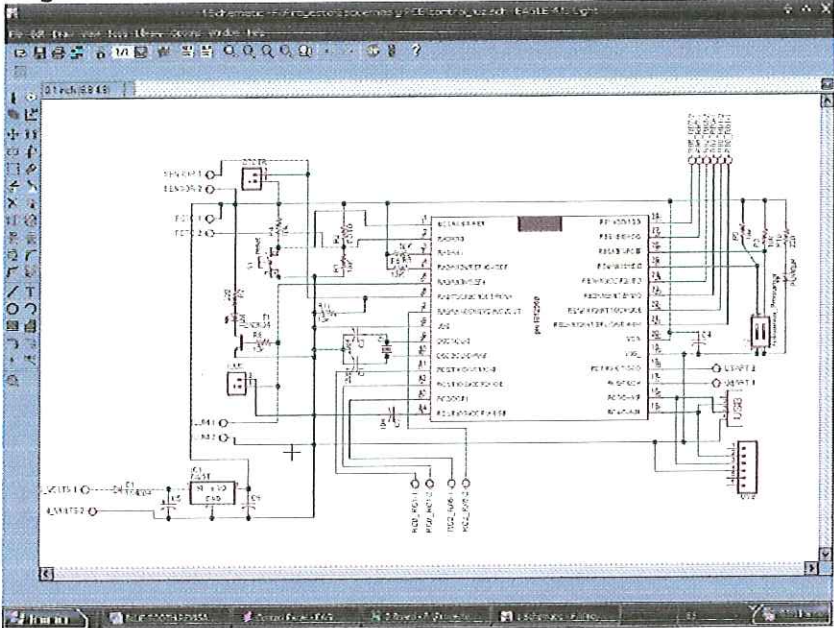
Para el diseño de este circuito, se utilizo el software Eagle. Este software cuenta la opción de creación del circuito PCB, y lo mas importante, el software nos da la mejor opción a la hora de realizar el diseño, en especial a la hora de realizar un circuito lo mas reducido posible.

Para el diseño, como se ve en la figura 31, lo primero es realizar es introducir en el software todas las partes del circuito, posteriormente al pasar el software al modo



diseño PCB, este generara un circuito lo mas pequeño posible, y también nos informara si el circuito tendrá que ser quemado en 1 sola cara o en 2.

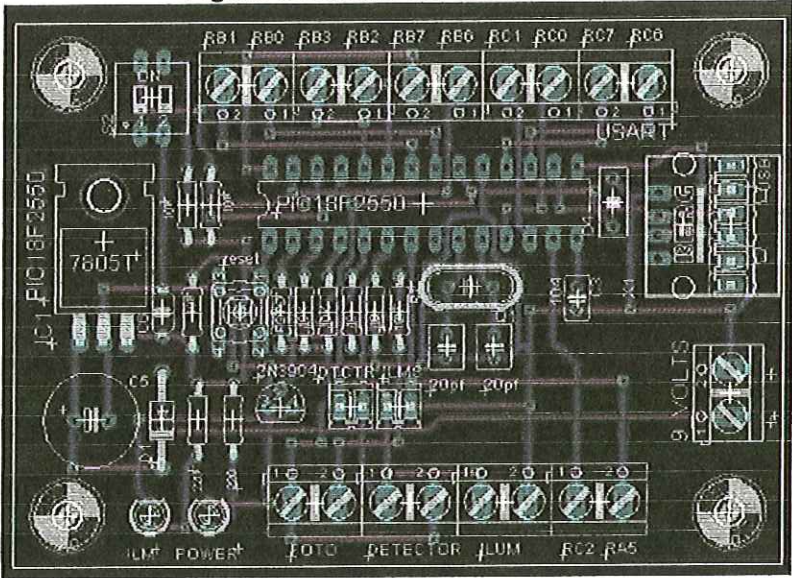
**Figura 31** Diseño del circuito PCB, ingreso de partes del circuito



Los autores

En la figura 32 podemos ver el circuito PCB de nuestra plataforma de control, diseñada con ayuda del software Eagle.

**Figura 32** Diseño del circuito PCB



Los autores



Para lograr un correcto funcionamiento, se instalaron borneras para conectar los diferentes sensores y en el caso del microcontrolador se instaló una base, la cual evita tener que soldar el Microcontrolador directamente en la placa, ya que si se presenta algún inconveniente con el microcontrolador, es mucho más difícil extraerlo.

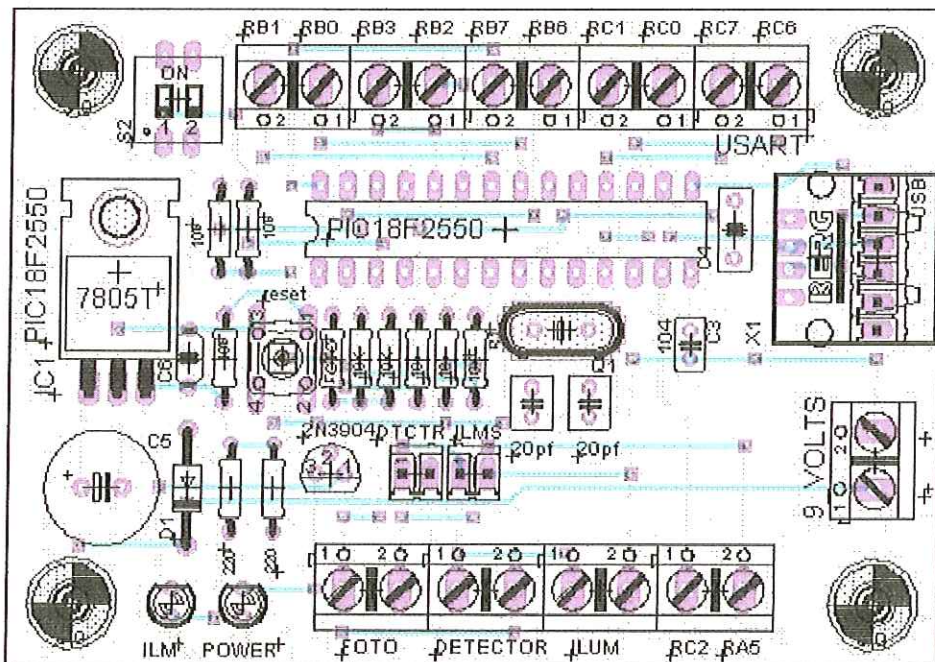
**Figura 33 Borneras instaladas en el Circuito**



Los autores

Como el sistema será de carácter didáctico, la placa cuenta con todos los nombres de los diferentes elementos que hacen parte de él, esto con el fin de una rápida y correcta identificación lo que hace más fácil su comprensión y funcionamiento Figura 34.

**Figura 34 Circuito PCB con los nombres de los elementos**



Los autores

### **3.3. DESARROLLO DEL SOFTWARE**

En esta etapa se desarrollarán los programas que se encargaran de permitir una correcta conectividad entre la plataforma de control y el sistema de monitoreo.

### **3.4. SOFTWARE PARA EL MICROCONTROLADOR PIC18F2550**

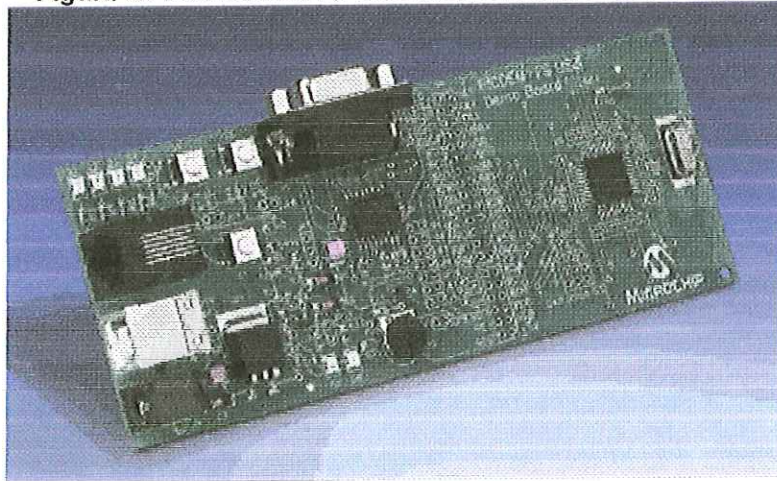
Como anteriormente se menciona, la familia de microcontroladores de Microchip PIC18f2455/2550/4555/4550, cuentan con una interfaz USB 2.0, con compatibilidad para LS y FS que permite una complejidad moderada a la hora de realizar una interfaz usando este puerto.

Estos dispositivos cuentan con el hardware necesario para realizar USB a LS o FS, pero estas operaciones deben ser programadas y pueden llegar a tener altos consumos de recursos del sistema, como lo son algunas interrupciones y muchos ciclos de maquina, de tal manera que seria necesario incluir en el programa un complejo set de rutinas solo para controlar el puerto USB. En conclusión esto incrementa en un alto grado la complejidad al desarrollar aplicaciones donde se requiere un alto rendimiento y estricto control del tiempo.

Para solucionar esto la firma Microchip creadora de la gama de microcontroladores PIC18F, creo una herramienta que agiliza las tareas de desarrollo.



**Figura 35 Kit PICDEM FS USB DEMONSTRATION BOARD**



Tomado de: [www.microchip.com](http://www.microchip.com)

Una de ellas, es el Kit PICDEM FS USB DEMONSTRATION BOARD, el cual consiste en un circuito el cual tiene como pieza principal un PIC18F4550 y entre otros elementos un potenciómetro y algunos leds y un sensor de temperatura.

La importancia de este KIT recae sobre el software de ejemplo en el cual se incluye todo el código fuente en C, donde se encuentran contenidas todas las rutinas necesarias para la comunicación USB. A las rutinas de programación se suman los controladores para el sistema operativo Windows, y una librería de acceso dinámico mpusbapi.dll que permite la conexión entre las aplicaciones para el PC y el microcontrolador.

Una de las características mas funcionales de este KIT es la posibilidad de programar el microcontrolador a través del puerto USB, sin necesidad de otros dispositivos y lo más importante sin necesidad del desmonte del microcontrolador, todo esto es posible gracias a la aplicación para PC que se incluye en el paquete.





Es importante anotar que la PICDEM FS USB DEMONSTRATION BOARD viene construida usando un cristal de 20 MHz, y si queremos usar otro tipo de cristal debemos incluir algunas líneas de configuración en el programa principal del BOOT LOADER.

```
#pragma config PLLDIV = 2
#pragma config CPUDIV = OSC3_PLL4
#pragma config USBDIV = 2
#pragma config FOSC = HSPLL_HS
```

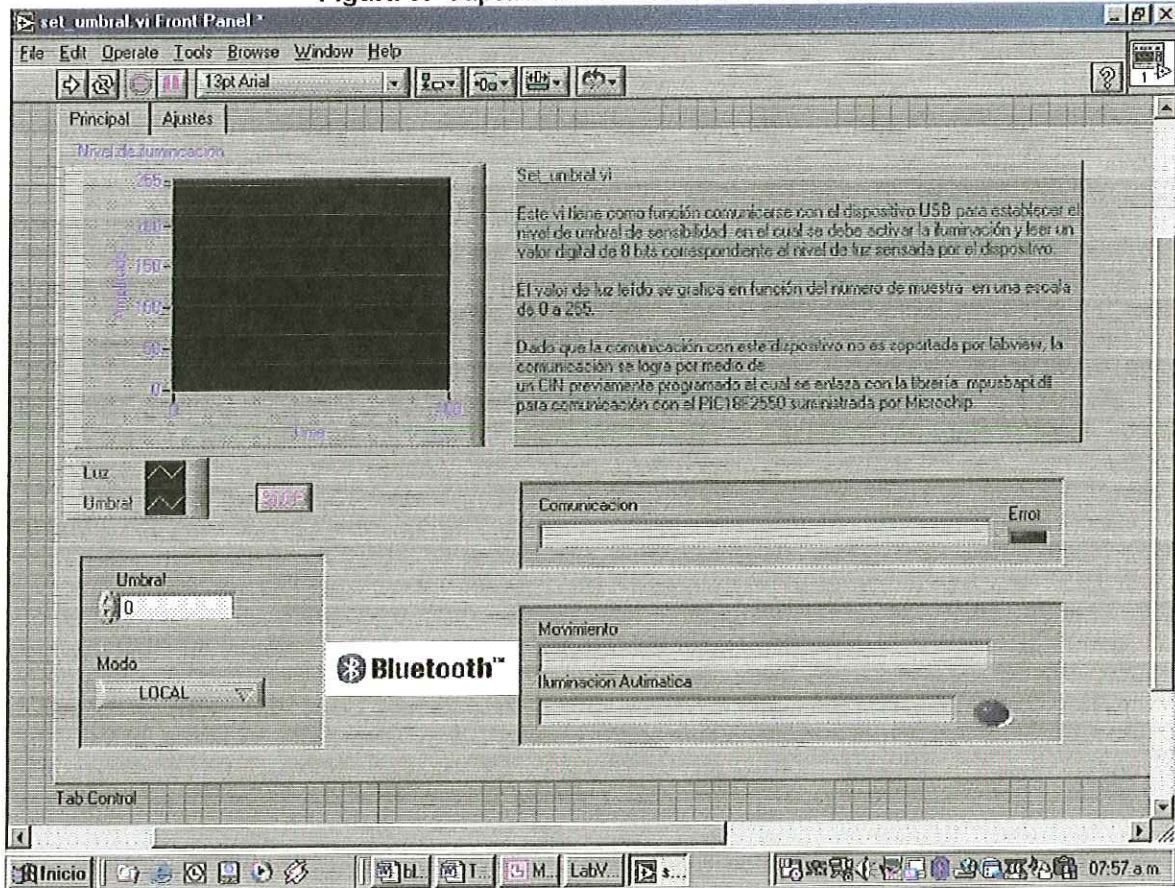
Para revisar el programa completo, encargado del funcionamiento del Microcontrolador, favor remitirse a los anexos.

### **3.6. SOFTWARE PARA COMUNIACION ENTRE LA PLATAFORMA DE CONTROL Y EL SISTEMA DE MONITOREO**

Para lograr la comunicación entre la plataforma de control y el sistema de monitoreo por medio del puerto USB, se realiza la conexión física por un cable tipo USB y el siguiente paso a realizar es diseñar un entorno de monitoreo que se encargue de visualizar los datos provenientes de los sensores y en nuestro caso permitir variar el umbral de luz, que es, el valor en el cual queremos que la luz se encienda.

Para esto usamos el software Labview 7 Express, en el cual realizamos el panel de control.

Figura 37 Captura del Sistema de Monitoreo



Los autores

Como podemos ver en la figura 37, el panel de monitoreo, presenta en tiempo real la información correspondiente a los sensores de presencia e iluminación. El panel de monitoreo, también informa si la luz se encuentra encendida o apagada, al igual que el tiempo que falta para que la luz se desactive por la no presencia de personas en el recinto. El sistema permite modificar el umbral, que permite al usuario programarle al sistema el nivel en el cual el sistema debe encender las luces.

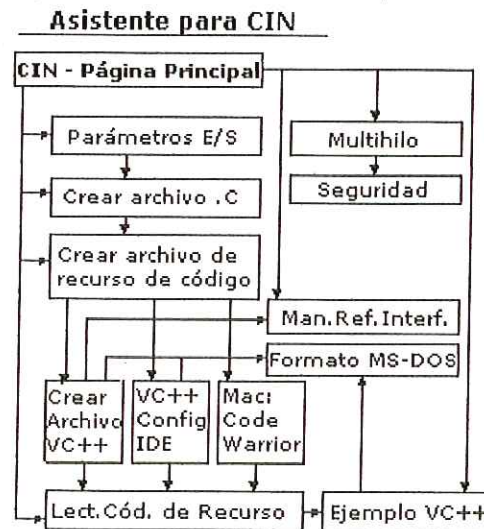
Como anteriormente explicamos, Labview tiene la oportunidad de llamar un código externo, ya que algunas aplicaciones no son soportadas directamente por

Labview. Las principales razones por las cuales se debe usar un CIN (Nodo de Interfaz de Código):

- Se necesita comunicar un hardware que Labview no soporta directamente.
- Se necesita hacer algo que es imposible o impracticable en el contexto de un diagrama de bloques.

Para la creación de un CIN, hay que seguir los siguientes datos, tomados del tutorial proporcionado por National Instruments Corporation.

**Figura 38 Asistente para la Creación de un CIN (Nodo de Interfaz de Código)**



Tomado de Tutorial [www.ni.com](http://www.ni.com)

Como podemos ver los principales pasos para la creación de un CIN, son:

1. Definición de los Parámetros de Entrada y Salida
2. Creación del Archivo Fuente
3. Creación del Archivo de Código de Recursos
4. Cargar el Archivo de Código de Recursos

Para más información [www.ni.com](http://www.ni.com).



### 3.6.1. CONEXION ENTRE DOS ESTACIONES USANDO LABVIEW 7

Para la conexión entre el sistema de monitoreo y otro dispositivo con tecnología Bluetooth, en nuestro caso otro PC, se utilizó el protocolo TCP/IP, el cual nos permite conectar a través de los módulos Bluetooth todo el sistema a una red y poder ser monitoreado desde cualquier punto de la misma.

Para poder establecer una comunicación de red utilizando el protocolo TCP/IP usando Labview 7 Express es posible usando los VIS para comunicación en red que ofrece este software.

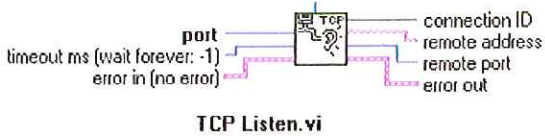
Una comunicación de red que use TCP/IP el cual es un protocolo orientado a conexión y con control de errores, lo que garantiza la integridad de la información. Es ideal para aplicaciones de automatización y control. Este protocolo unido a una poderosa herramienta como es el Labview 7 se convierte en una herramienta excelente a la hora de diseñar sistemas de control y automatización.

Para realizar una comunicación TCP/IP deben existir al menos 2 estaciones entre las que se va a realizar la comunicación. La primera se denomina Estación Pasiva la cual espera por una conexión entrante y una estación activa la cual inicia la comunicación realizando una llamada al número IP y puerto de la estación pasiva.

**Estación Pasiva:** El bloque encargado de esperar por una conexión TCP/IP se puede encontrar en *All functions -> Comunicación -> TCP* y su nombre es "TCPlisten.vi" y cuyo icono se presenta a continuación

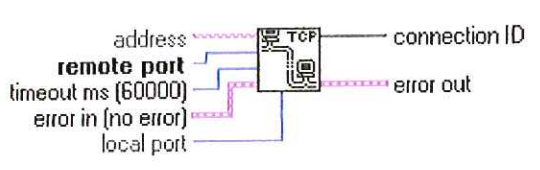


TABLA 10 Configuración de una estación pasiva en Labview

ICONO	CONECTOR	FUNCION - DESCRIPCION
	Port	<p>Esta entrada establece el puerto a través del cual se realizara la comunicación con la otra estación ó dispositivo.                  NOTA: Las dos estaciones deben tener el mismo número de puerto.</p>
	Coneccion ID	<p>Nos suministra un identificador para hacer accesos a esta conexión una vez establecida, esta salida debe estar cableada a todos los bloques TCP involucrados en la comunicación con el fin de garantizar claridad respecto a cual canal de comunicación usar.</p>
	<i>remote address</i>	<p>La cual reporta la dirección IP de la estación activa que inicio la conexión una ves esta a sido establecida.</p>
	<i>error out</i>	<p>Informa de errores surgidos en el proceso de conexión TCP</p>
	<i>timeout ms</i>	<p>Por medio de la cual se establece el lapso de tiempo durante el cual se esperará por una conexión TCP entrante, si transcurrido el tiempo programado no se a realizado una llamada de otra estación este conector generará un error.</p>

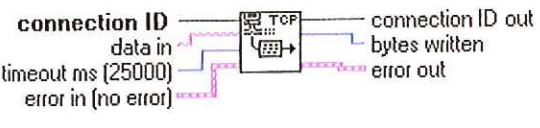
**Estación Activa:** Su función es la de iniciar una comunicación llamando a una estación pasiva usando el numero IP y puerto de la estación con la cual se requiere realizar la comunicación. El bloque principal que hace a una estación activa es *TCP Open Conection* el cual se puede encontrar en *All functions -> Comunicación -> TCP* y cuyo icono se muestra a continuación.

TABLA 11 Configuración de una estación activa en Labview

ICONO	CONECTOR	FUNCION - DESCRIPCION
	Address	Se configura la dirección IP de la estación o dispositivo al cual se esta llamando.
	Remote Port	Esta entrada establece el puerto por el cual se realizara la conexión.
	<i>time out</i>	mide el tiempo limite para que una estación pasiva conteste
	<i>error out y connection ID</i>	Tenemos las salidas <i>error out</i> y <i>connection ID</i> que tienen las mismas funciones que para el bloque <i>TCP Listen.vi</i>

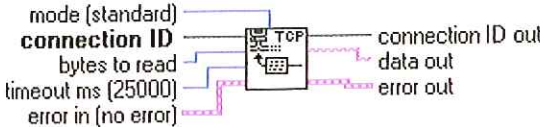
**Bloques para leer y escribir:** Una vez establecida la conexión los datos pueden fluir entre la estación pasiva y la estación activa o al revés. Este flujo de información se controla con los bloques TCP read y TCP writer para recibir y enviar datos, pero específicamente por medio del bloque TCP writer se envían datos de una estación a otra.

TABLA 12 Configuración de bloques para leer y escribir en Labview

ICONO	CONECTOR	FUNCION - DESCRIPCION
	Connection ID	Debe ir cableado a la salida con el mismo nombre del bloque <i>TCP Listen</i> o <i>TCP Open Connection</i> con el cual se inicio la comunicación para que se indique cual conexión se deben utilizar para enviar los datos.
	Data In	Se ingresa la cadena de caracteres que se desee enviar.
	<i>timeout ms</i>	Vigila que no se exceda el tiempo máximo que puede transcurrir mientras la otra estación recibe los datos, en caso de que este tiempo se exceda la salida <i>error out</i> generará un mensaje de error.
	<i>bytes griten</i>	Indica cuantos bytes se han enviado satisfactoriamente.

TCP read tiene como función recibir datos enviados desde otra estación, y su icono se encuentra a continuación.

TABLA 13 Configuración para recibir datos con Labview

ICONO	CONECTOR	FUNCION - DESCRIPCION
	<i>Connection ID</i>	Al igual que en <i>TCP Write</i> debe ir cableado a la salida del mismo nombre del bloque <i>TCP Listen</i> o <i>TCP Open Connection</i> con el cual se inicio la comunicación para que se indique cual conexión se debe usar para recibir los datos.
	<i>Bytes to read</i>	Se programa el numero de bytes que se esta esperando en esta recepción
	<i>Timeout ms</i>	Al igual que en los vi anteriormente explicados tiene como función permitir establecer el tiempo máximo que se esperará para que se realice la operación de recepción de datos.
	<i>Error out</i>	<i>out</i> reporta los errores ocurridos durante esta operación
	<i>Data out</i>	Retorna los bytes recibidos dependiendo del valor de la entrada <i>mode</i> la cual establece uno de cuatro modos de operación.

## MODOS DE OPERACIÓN DE TCP read

Utilizando la entrada *mode* se especifica el modo de recepción de datos, el cual puede tomar uno de cuatro valores, *Standard*, *Buffered*, *CRLF* o *Immediate*:

**STANDARD:** El cual es el modo por defecto, este bloque espera a que todos los bytes programados en la entrada *bytes to read* arriben o a que expire el tiempo establecido con *timeout ms* antes de retornar cualquier valor, en caso de que expire el tiempo establecido se retornarán los bytes que se alcanzaron a recibir por la salida *data out* y se reportara un error.



**BUFFERED:** Este vi funciona casi igual que en el modo *Standard* con la única diferencia de que en caso de que expire el tiempo programado y solo se han recibido una parte de los bytes programados no se retorna ningún byte y se reporta un error.

**CRLF:** Este vi espera por la cantidad de bytes programados seguidos por un carácter de return CR seguido de una alimentación de línea LF o a que expire el tiempo establecido para la recepción antes de retornar cualquier valor, en caso de que se reciban datos sin el carácter return seguido de la alimentación de línea no se retorna ningún dato y se reporta un error.

**IMMEDIATE:** Este vi retorna todos los bytes recibidos en el momento en que estos arriban, y solo espera que el tiempo programado expire si no se a recibido ningún dato.

## **FLUJO BÁSICO DE LAS RUTINAS PARA COMUNICACIÓN TCP/IP EN LABVIEW**

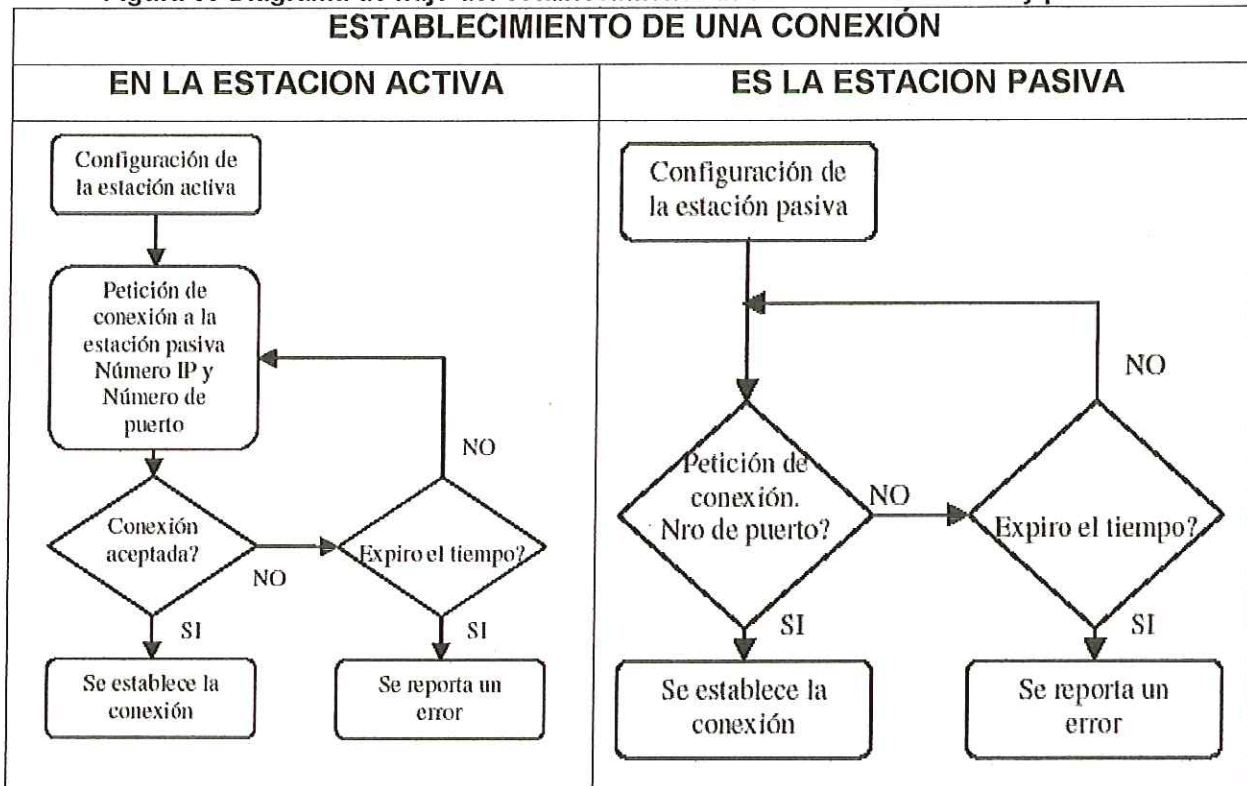
Se puede decir que hay dos flujos básicos para la realización de una rutina que soporte comunicación TCP/IP en labview, el establecimiento de la comunicación y el intercambio de datos.

### **ESTABLECIMIENTO DE LA COMUNICACIÓN**

La estación pasiva espera el tiempo establecido a que otra estación inicie una comunicación por el puerto programado, la estación activa llama a la estación

pasiva y solicita la conexión por un puerto específico, si la estación pasiva detecta que una conexión se está solicitando a través del puerto establecido se establece la comunicación y tanto la estación activa como pasiva queda en capacidad para enviar y recibir datos,

Figura 39 Diagrama de flujo del establecimiento de las estaciones activa y pasiva



Los autores

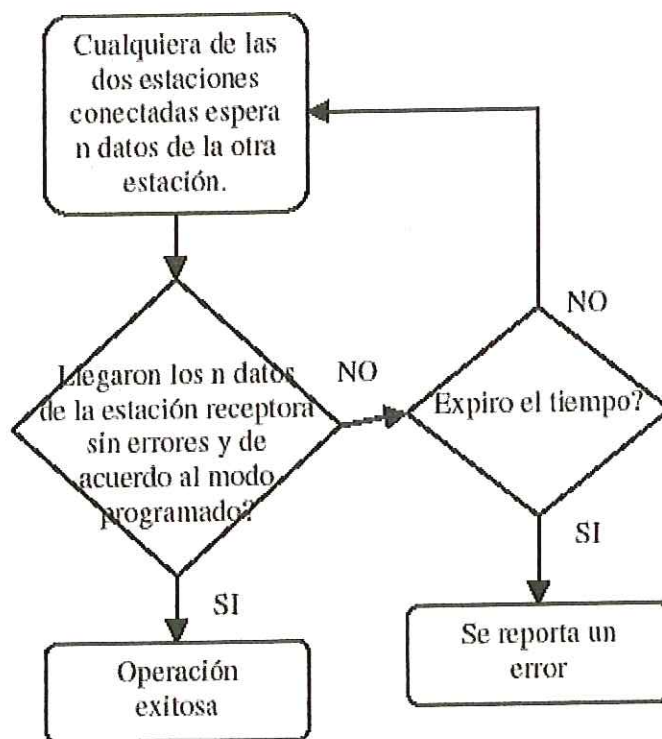
### ENVIÓ DE DATOS O ESCRITURA

Cualquiera de las dos estaciones envía un número de datos determinado, si la estación receptora no recibe todos los datos y sin errores se reporta un error en la estación emisora.

## RECEPCIÓN DE DATOS O LECTURA

Durante una operación de lectura cualquiera de las dos estaciones que han establecido una conexión espera por datos, y dependiendo del modo programado puede actuar de cuatro maneras diferentes, como se puede ver en la figura 40

Figura 40 Operación de lectura o recepción de datos de una conexión TCP/IP



Los autores

Para revisar el programa completo en Labview, que esta encargado de realizar la conexión entre el sistema de control, el sistema de monitoreo y otros dispositivos por medio de la tecnología Bluetooth, por favor remitirse a los anexos.



#### 4. PRACTICAS

Las practicas planteadas a continuación, tienen como fin, enriquecer el conocimiento de los estudiantes de la facultad de Ingeniería Mecatronica de la Universidad Autónoma de Bucaramanga y permitir que este proyecto tenga un mayor aporte en la sociedad, aparte de lo ya mencionado anteriormente sobre el conocimiento a fondo sobre lo que es la domótica, sus aplicaciones, sus medios de comunicación, sus ventajas y desventajas.

Las prácticas planteadas a continuación son dos, la primera de ellas trata sobre la identificación de los componentes del sistema de control domótico y la implementación de este sistema en un entorno seleccionado. La segunda practica trata de involucrarse un poco más en la forma en que se logra la interacción entre Labview, el sistema de comunicación inalámbrica Bluetooth y la plataforma de control (Microcontrolador).

Para el desarrollo de estas practicas el estudiante deberá reforzar sus conocimientos acerca de la Domótica, comunicación inalámbrica Bluetooth, Protocolo de comunicación TCP/IP, el Software de control Labview, Nodo de Interfaz deCodigo (CIN), estos temas de consulta están debidamente repartidos en las dos practicas planteadas, los cuales no se repiten ya que las practicas deben realizarse de manera secuencial.

Esperamos con estas prácticas que los estudiantes logren adquirir nuevos conocimientos, tanto en la domótica como en la comunicación inalámbrica a través de la tecnología Bluetooth y así podamos cumplir con el objetivo trazado inicialmente.

## PRACTICA No. 1

### CONEXIÓN DEL SISTEMA DOMOTICO

#### 1. OBJETIVOS

- Identificar los componentes y entender su funcionamiento dentro del sistema de control domótico.
- Lograr la interacción entre los diferentes dispositivos que componen el sistema de control domótico ( sensores, Bluetooth etc.)

#### 2. MATERIALES

- Sistema de control domótico con tecnología Bluetooth.
- Guia de Conexión

#### 3. TEMAS DE CONSULTA

- Domotica
- Bluetooth
- Puerto USB
- Labview

#### **4. PROCEDIMIENTO**

- 4.1 Seleccionar un recinto en el cual se pueda instalar el sistema de control y monitoreo domótico.
- 4.2 Conectar el sistema teniendo como referencia los pasos mencionados en la sección de anexos ( guía de conexión )
- 4.3 Hacer interactuar el sistema de control, con el programa de monitoreo.
- 4.4 Identificar la cobertura que tiene el sensor de presencia dentro del recinto en que se implemento el sistema de control domótico.
- 4.5 Establecer una comunicación Bluetooth entre el PC de monitoreo y cualquier otro PC existente en el lugar.

#### **4. OBSERVACIONES Y CONCLUSIONES.**



## PRACTICA No. 2

### BLUETOOTH CON LABVIEW

#### 1. OBJETIVOS

- Conocer la interacción entre la herramienta grafica de control y diseño labview y el sistema domótico con tecnología bluetooth.
- Identificar los cambios existentes en el entorno al modificar los datos de programación de la plataforma de monitoreo del sistema domótico.

#### 2. MATERIALES

- Sistema de control domótico con tecnología Bluetooth.
- Tutorial para la creación de un CIN(Nodo de Interfaz de Código), suministrado por Nacional Instruments ([www.ni.com](http://www.ni.com))

#### 3. TEMAS DE CONSULTA

- Comunicación TCP/IP con labview
- Nodo de Interfaz de Codigo (CIN)

### **3. PROCEDIMIENTO**

- 3.1 Teniendo como referencia el tutorial suministrado por National Instruments, para la creación de un CIN (Nodo de Interfaz de Código), establecer por medio de este la comunicación entre Labview y la Plataforma de Control.
- 3.2 Monitorear el funcionamiento del sistema de control domótico desde un punto alejado del recinto utilizando la red como medio de acceso.
- 3.3 Mediante el software de control labview, modificar los parámetros del umbral de funcionamiento del sensor de luminosidad del sistema domótico.
- 3.4 Hacer 5 modificaciones de este umbral y observar los cambios existentes en el entorno donde está implantado el sistema de control domótico.
- 3.5 Tabular los datos.
- 3.6 Modificar la secuencia o lógica completa de funcionamiento de la plataforma de monitoreo, variando la sintaxis del programa original en labview.
- 3.7 Analizar el comportamiento del sistema de control domótico monitoreado desde este nuevo punto de acceso con la nueva programación y establecer las diferencias de funcionamiento entre el antiguo y el nuevo monitoreo.

### **4. OBSERVACIONES Y CONCLUSIONES.**

## 5. CONCLUSIONES

El proyecto logró cumplir con su objetivo primordial de diseñar y construir un sistema domótico para monitorear tanto la presencia como la iluminación en un recinto y enviar este monitoreo a un PC empleando la comunicación inalámbrica Bluetooth.

Al iniciar este proyecto se comenzó con la idea de utilizar un microcontrolador como plataforma de control, el cual enviaría los datos recibidos de los dos sensores a utilizar como lo son el de presencia e iluminación, directamente al PC, utilizando la comunicación inalámbrica Bluetooth, pero al investigar el funcionamiento de los módulos Bluetooth, pudimos darnos cuenta que estos módulos no trabajan en dispositivos esclavos ( microcontroladores en nuestro caso ), sino que solamente funcionan comunicando dispositivos maestros ( PC a PC ) es decir, no pudimos comunicar directamente el PC con el microcontrolador a través de Bluetooth. Por esta razón, debimos pensar primero en una comunicación PC y microcontrolador cableada antes de utilizar los módulos Bluetooth para establecer una comunicación inalámbrica entre dos PC.

Debido al problema mencionado anteriormente, se investigo y se conoció a fondo el funcionamiento del puerto de comunicación USB ya que a través de este, pudimos establecer la comunicación entre el microcontrolador pic18f2550 y el PC.

Se realizo un programa en labview a través del cual podemos tanto monitorear en tiempo real lo que sucede con los dos sensores utilizados y la plataforma de control (microcontrolador), como variar los umbrales de funcionamiento de los



sensores. Al intentar establecer directamente esta comunicación entre el microcontrolador y el PC utilizando como software a labview, se encontró que este software no permite este tipo de conexión directamente. Para solucionar esto, se utilizó un CIN, el cual es un nodo de interfaz de comunicación, que nos permitió lograr un correcto envío y recibimiento de datos entre el microcontrolador y el PC utilizando como herramienta a labview.

Al momento de establecer la comunicación Bluetooth entre los dos PC, para lo cual se instaló un módulo Bluetooth TBW 100 en cada uno de ellos, tuvimos que investigar sobre el protocolo TCP / IP el cual es el mismo que se maneja en las conexiones de Internet, ya que estos módulos al servir como emisor y receptor generan en cada PC un puerto serial. Al crearse estos puertos seriales en cada PC, los módulos bluetooth se convierten prácticamente en una conexión serial imaginaria y este protocolo nos permitió lograr tanto el envío como la recepción de datos a través de este puerto serial imaginario. Además de permitirnos la conexión del sistema a cualquier red LAN, dando la posibilidad a los usuarios de la red de monitorear el sistema desde cualquier punto de esta.

En el momento de escoger el microcontrolador se tuvieron en cuenta sus características, principalmente que contara con puerto de conexión USB, entre los posibles microcontroladores a utilizar estaban el PIC16C745 y el PIC18F2550. se escogió el segundo de estos debido a que además de contar con puerto USB, la cual era una característica esencial debido a que los módulos Bluetooth manejan este tipo de puerto, este microcontrolador cuenta con memoria flash lo que permite al usuario programarlo varias veces.

Después de un completo estudio, se pudo comprender el funcionamiento de la tarjeta PICDEMFS USB DEMONSTRATION BOARD de Microchip, la cual nos



servió como base, para entender el acoplamiento de los sensores tanto de presencia como de luminosidad utilizados en nuestro caso, dentro del sistema trabajo del microcontrolador PIC18F2550, el cual como ya lo dijimos anteriormente es uno de los pertenecientes a la familia de microcontroladores que trabajan con puerto USB.

Para la elección del sensor de presencia se tuvo en cuenta sus características y funcionamiento, así como el área de cobertura y en especial la inmunidad a corrientes de aire ya que estas en algunos casos pueden ocasionar falsas alarmas en este tipo de sensores. Además de lo anterior, este sensor cuenta con inmunidad a mascotas hasta de 13.6 Kg., lo que es de gran utilidad si el sistema se piensa usar en lugares donde haya presencia de animales.

Dentro de las posibilidades existentes en el mercado en sensores de iluminación existían varias opciones, entre ellas la utilización de un luxometro o la utilización de una LDR. La primera opción cumplía con todas las características que necesitábamos de este sensor para nuestro proyecto, pero presentaba el inconveniente de ser un sensor demasiado costoso. Por esta razón se decidió la utilización de una LDR, la cual nos cumple con los parámetros establecidos al comienzo del proyecto.

Se pudo implantar adecuadamente un sistema de seguridad para el recinto Domotico a controlar, gracias a este sistema de seguridad, el proyecto abarca todas las ramas que involucran a la domótica como lo son: energía, confort, telecomunicaciones y seguridad.

Un aporte adicional de este proyecto fue el poder establecer el diseño metodológico a seguir en la construcción de un sistema domótico. Para lograr este

diseño se debe tener en cuenta el entorno en donde se desee implementar el sistema, esto conlleva estudiar los diferentes dispositivos a usar, la tecnología existente en el mercado, sus beneficios , y que todos los dispositivos seleccionados ya sea de la parte de software, de electrónica o de sensorica se logren acoplar e interactuar entre ellos adecuadamente, todo esto con miras a tener una buen resultado final.

Uno de los inconvenientes encontrados durante la realización de este proyecto, fue la falta de documentación y conocimiento sobre el tema, para esto debimos investigar a fondo sobre las tecnologías usados en nuestro proyecto y en especial sobre la comunicación inalámbrica Bluetooth. Y ante el poco uso de esta tecnología Bluetooth en nuestro país es difícil conseguir productos que vayan de la mano con esta tecnología. Para esto recurrimos a los pocos distribuidores de estos productos existentes en el mercado nacional y evaluamos las diferentes posibilidades que nos ofrecían y así poder escoger la que mejor se adaptaba a nuestras necesidades.

Una de las finalidades del proyecto es aprovecharlo para que sirva como fuente de practicas del semillero de investigación de la facultad, para esto se realizaron dos practicas que permiten la manipulación y el conocimiento de las tecnologías usadas en él. También puede ser utilizado como base o fundamente para futuros proyectos o investigaciones los cuáles puedan llegar en algún momento a presentar la posibilidad de realizarle mejoras a este.

## BIBLIOGRAFIA

FLOYD, Thomas L. Fundamentos de Sistemas Digitales, Séptima Edición. Madrid: Prentice Hall, 2000, 1128 p.

PALLAS, Ramón, Sensores y Acondicionadores de Señal, Tercera Edición. Barcelona: Alfaomega, 2001, 480 p.

NASHELSKY, Luis, BOYLESTAD, Robert. Electrónica, Teoría de Circuitos, Sexta Edición. México: Pearson, 1997, 949 p.

TAFANERA, Antonio. Teoría y diseños con microcontroladores PIC. Buenos Aires: Autores Editores, 1998, 294 p.

BRAY, Jennifer. Bluetooth: Connect without cables. Upper Saddle River: Prentice Hall, 2001, 495 p.

CUENCA, Martín. Microcontroladores PIC: La solución en un chip, Quinta Edición. Madrid: Paraninfo, 2001, 489 p.

MICROCHIP, PIC 18F2455/2550/4550. Data Sheet

MICROCHIP, Picdem Fs USB Demonstration Board User's Guide

## PAGINAS DE INTERNET

[www.bluetooth.org](http://www.bluetooth.org)

[www.domointel.com](http://www.domointel.com)

[www.ni.com](http://www.ni.com)

[www.bosch.com.co](http://www.bosch.com.co)

<http://www.monografias.com/trabajos12/tecninal/tecninal.shtml#intro>

[www.atenea.unicauca.edu.co/~dabravo/bluetooth/index.htm](http://www.atenea.unicauca.edu.co/~dabravo/bluetooth/index.htm)

[http://web.frm.utn.edu.ar/comunicaciones/modelo\\_osi.html#1](http://web.frm.utn.edu.ar/comunicaciones/modelo_osi.html#1)

<http://www.monografias.com/trabajos11/usbmem/usbmem.shtml>

[www.trendnet.com](http://www.trendnet.com)

[www.microchip.com](http://www.microchip.com)

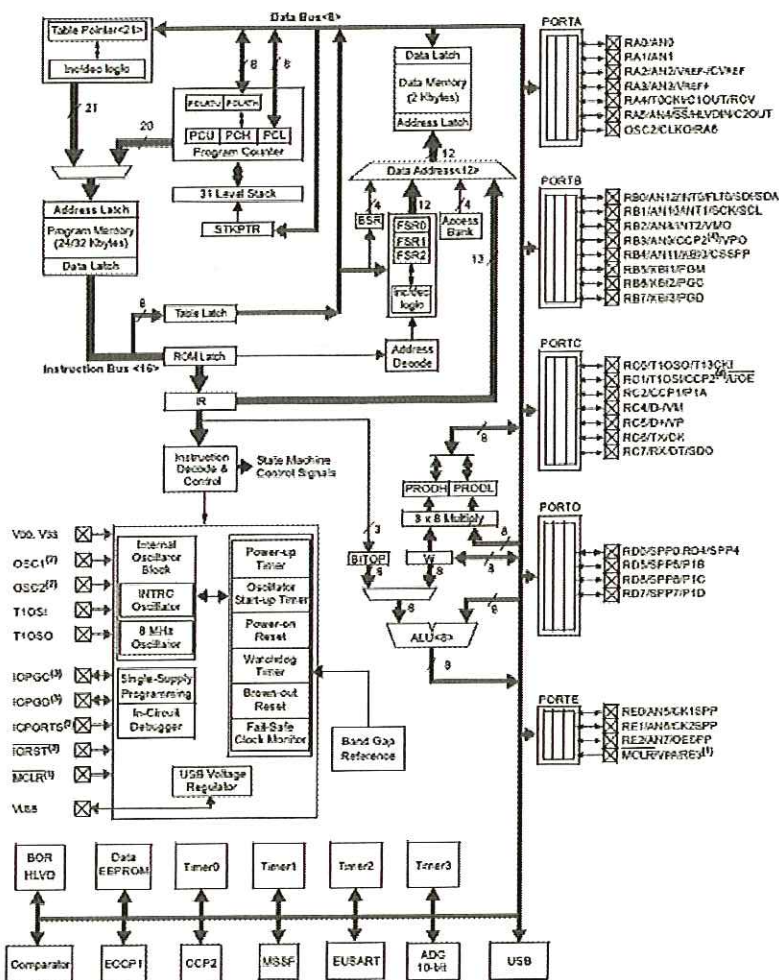


# **ANEXOS**

# ANEXO1 MICROCONTROLADOR DE LA FAMILIA 18F2455/2550/4555/4550

## SISTEMAS DE MEMORIA

El contador de programa tiene 21 bits, lo que permite direccionar hasta 2Mbytes. Las posiciones por superiores a los 24KB para 18F4555 y 32KB para 18F4550 se leen como '0' lo que se corresponde con la instrucción NOP.



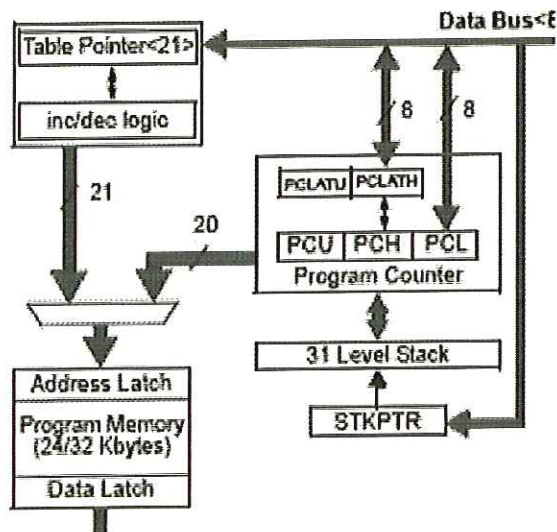
## DIRECCIONAMIENTO DE LA MEMORIA

La unidad de direccionamiento de la memoria es el byte. Las instrucciones se almacenan ocupando cada una de ellas dos posiciones de memoria con ordenación little endian (byte menos significativo en la posición más baja).

Para mantener la alineación del PC con las direcciones de comienzo de las instrucciones, el PC siempre se incrementa en dos posiciones. Las instrucciones CALL, GOTO, LSRF y MOVFF están compuestas por 2 palabras. La segunda palabra almacena 12bits del valor literal junto con el valor '1111', como bits más significativos. El PC está formado por 21 bits distribuidos en 3 registros: PCU<20:16>;PCH<15:8>;PCL<7:0>.

El PCL es el único que se puede leer y escribir directamente y para evitar problemas de alineación con las direcciones de las instrucciones el bit menos significativo está fijo a '0'. Las escrituras o lecturas de PCU y PCH se llevan a cabo mediante PCLATU y PCLATH, respectivamente. Una lectura de PCL, lleva el valor de PCU y PCH hasta PCLATU y PCLATH. De la misma forma una escritura sobre PCL, lleva el valor de PCLATU y PCLATH a PCU y PCH. Las únicas instrucciones que pueden escribir directamente todo el PC son CALL, RCALL y GOTO, con ellas el valor de PCLATU y PCLATH no se lleva al PC.

Memoria de programa Posiciones →			Dirección de palabra		
			LSB = 1	LSB = 0	
				000000h	
				000002h	
				000004h	
				000006h	
Instruc. 1:	MOVLW	05 5h	0Fh	55h	000008h
Instruc. 2:	GOTO	00 06h	EFh	03h	00000Ah
			F0h	00h	00000Ch
Instruc. 3:	MOVFF	12 3h, 45 6h	C1h	23h	00000Eh
			F4h	56h	000010h
					000012h
					000014h



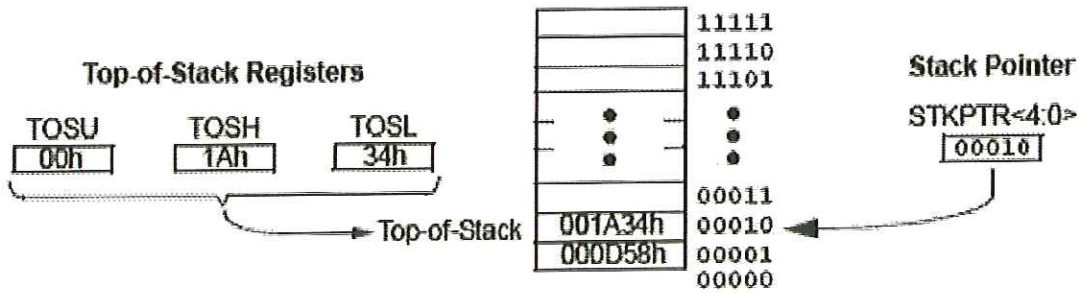
## PILA

La pila sirve para almacenar las direcciones de vuelta para las llamadas a subrutina o las interrupciones. El PC se guarda en la pila cuando se ejecuta la instrucción CALL, RCALL o se produce una interrupción. El PC se desapila cuando se ejecuta la instrucción RETURN, RETLW o RETFIE. PCLATU y PCLATH no se ven afectadas por las instrucciones RETURN or CALL. La pila tiene 31 posiciones de 21 bit cada una y con un puntero a pila de 5 bits el STKPTR.

A la cima de la pila apunta el registro STKPTR. La dirección situada en la cima de la pila puede ser escrita o leída mediante tres registros TOSU:TOSH:TOSL, donde está almacenada dicha dirección.



**Dirección de vuelta <20:0>**



El registro STKPTR contiene además dos bits para indicar cuando la pila se ha sobrellenado o cuando sobre-vaciado. El bit STKFUL indica que se han apilado 31 veces consecutivas el PC sin que se haya desapilado ninguna vez. La activación de STKFUL produce un reset si el bit STVREN del registro de configuración está activo. Sino los siguientes valores no se vuelven a apilar y el Stack Pointer permacene a 31. Cuando se desapilan todos los elementos de la pila, el bit STKUNF se activa y el valor desapilado es cero. Existen dos instrucciones para el manejo de la pila: PUSH y POP. La instrucción PUSH pone el PC en la Pila y la instrucción POP descarta el valor actual en la cima de la pila.

**REGISTRO: STKPTR: STACK POINTER REGISTER**

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
STKFUL <sup>(1)</sup>	STKUNF <sup>(1)</sup>	—	SP4	SP3	SP2	SP1	SP0	
bit 7								bit 0

- bit 7 **STKFUL: Pila llena<sup>(1)</sup>**  
1 = Pila está llena o sobre escrita  
0 = Pila no está llena
- bit 6 **STKUNF: Pila está sobre leída<sup>(1)</sup>**  
1 = Ha ocurrido la sobre lectura de la pila  
0 = No ha ocurrido la sobre lectura
- bit 5 **No implementado: Se lee como '0'**
- bit 4-0 **SP4:SP0: Bits del Stack Pointer**

**Note 1:** Bit 7 y bit 6 se desactivan por software o por el POR.

En el caso de producirse una llamada a subrutina o interrupción, el código de dichas funciones debe de salvar el contenido de todos aquellos registros que vaya a modificar. Tres registros cuyo contenido es importante salvar son:

STATUS (reg. de estado), WREG (Acumulador) y el BSR (Selección de banco). Los PIC 18 implementan una pila de un solo nivel para almacenar el contenido de estos tres registros de forma rápida. Cada posición dentro de la pila no se puede leer ni escribir. Cuando se produce una interrupción el contenido de estos tres registros se almacena en la pila. Al terminar la subrutina de interrupción, se puede restaurar el contenido de los registros mediante la instrucción RETFIE, FAST.

Si no se utilizan las interrupciones, los registros de pila rápidos se pueden utilizar en llamadas a subrutina. En ese caso la llamada debe ser CALL label, FAST y la finalización de la subrutina RETURN, FAST

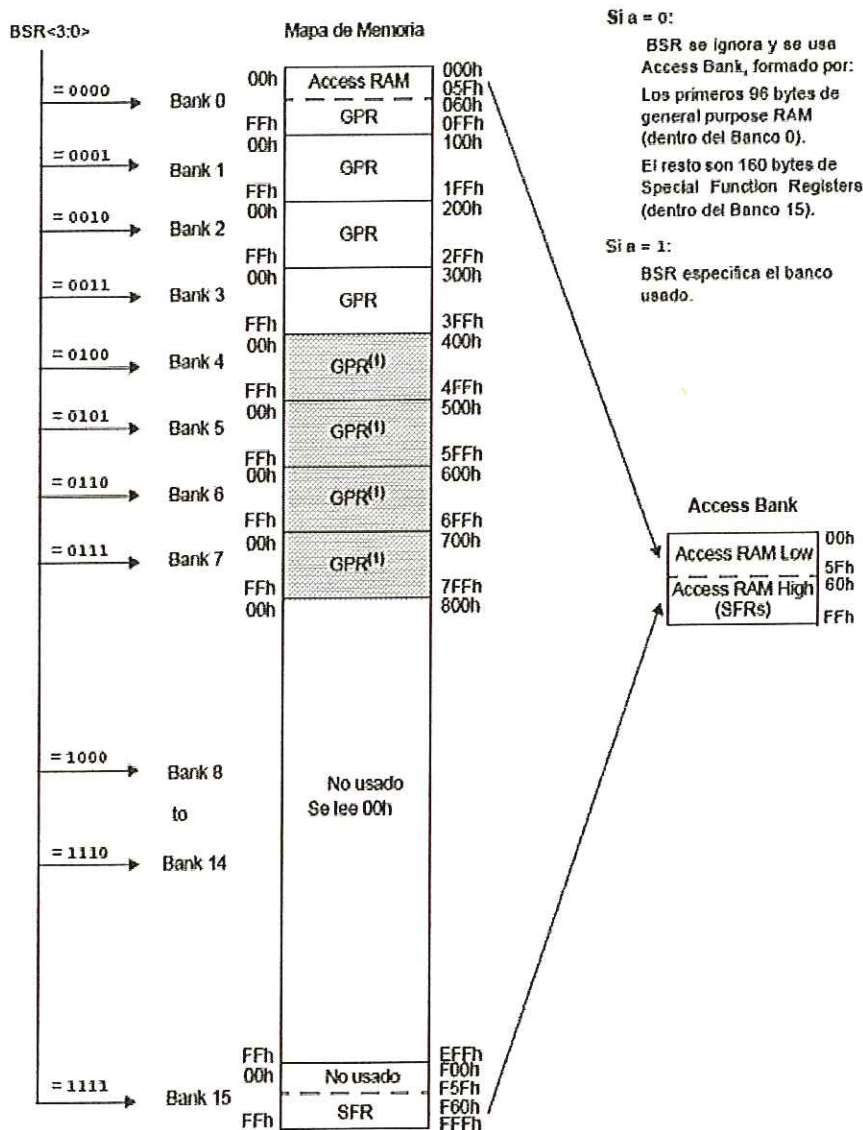
CALL	SUB1, FAST	;STATUS, WREG, BSR
		;SAVED IN FAST REGISTER
		;STACK
	•	
	•	
SUB1	•	
	•	
	RETURN, FAST	;RESTORE VALUES SAVED
		;IN FAST REGISTER STACK

## MEMORIA DE DATOS

Todos los dispositivos tienen una memoria RAM de datos de 2048 bytes, implementada mediante tecnología SRAM. La memoria de datos está dividida en dos tipos de posiciones:

- SFR (Special function registers): para controlar el estado del micro y los periféricos.
- GPR (General purpose registers): son posiciones para almacenar datos de carácter general.

Para direccionar la memoria es necesario generar direcciones de 12 bits, permitiendo acceder a 4096 posiciones de las que 2048 no están implementadas. Los 2048 bytes de memoria implementada en los PIC18F2455/2550/4455/4550 se dividen en 8 bancos de 256 bytes.



Existen cuatro modos de direccionamiento de la memoria:

- Inherente.
- Literal.
- Directo

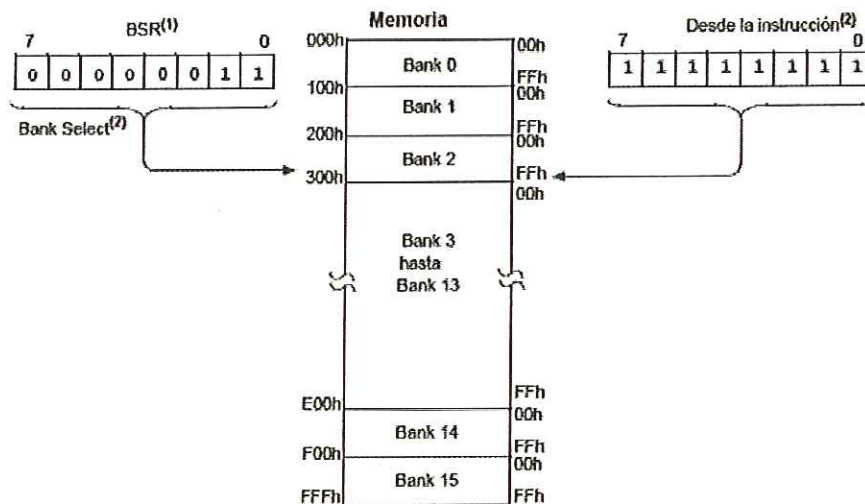
- Indirecto.

En el direccionamiento inerente, se realiza con instrucciones que siempre se aplican sobre el mismo destino. Siempre operan de forma implícita sobre los mismos registros. Ejemplo: SLEEP, RESET, entre otros.

En el direccionamiento literal, se utiliza una constante (literal) que acompaña al código de operación dentro de la propia instrucción.

En el direccionamiento directo, se especifica la dirección del operando. Pero esta dirección se especifica con un campo de 8 bits dentro de la instrucción y por lo tanto es necesario extenderla hasta los 12 bits para obtener una dirección de memoria correcta.

Extensión mediante BSR (Bank Select Register)



Nota 1: El bit de Access RAM bit de la instrucción fuerza una sobre escritura del sobre los registros asociados al modo Access Bank.

2: La instrucción MOVFF incluye los 12 bits.

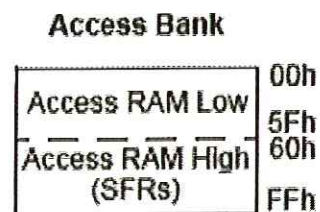
## DIRECCIONAMIENTO DIRECTO

Extensión mediante Access Bank. El uso de BSR para acceder a los distintos bancos de memoria tiene el inconveniente de que es necesario, saber



exactamente cuál es el banco que actualmente está seleccionado. El Access Bank acelera el acceso a las posiciones más comunes de memoria, ignorando el valor del BSR. Está formado sólo por 256 bytes (de esta forma con los 8 bits provenientes de la instrucción es suficiente para direccionar todo el Banco). Las primeras 96 posiciones son de GNR (posiciones de memoria más bajas) y 160 siguientes los SFR.

Una instrucción puede utilizar cualquiera de los dos métodos de acceso directo. El método utilizado en un determinado instante, se especifica mediante el parámetro a dentro de la propia instrucción. Si 'a=1' usa BSR, si 'a=0' utiliza access ram y el BSR se ignora.



## DIRECCIONAMIENTO INDIRECTO

En la instrucción se especifica la dirección de un registro donde se encuentra la dirección del operando. Estos registros punteros donde se almacenan las direcciones se denominan File Select Registers (FSR). Sólo existen tres registros para realizar direccionamiento indirecto: FSR0, FSR1 y FSR2. Cada registro es de 16 bits y representa la concatenación de dos registros de 8 bits: FSR0=> FSR0H:FSR0L; FSR1=> FSR1H:FSR1L; FSR2=> FSR2H:FSR2L.

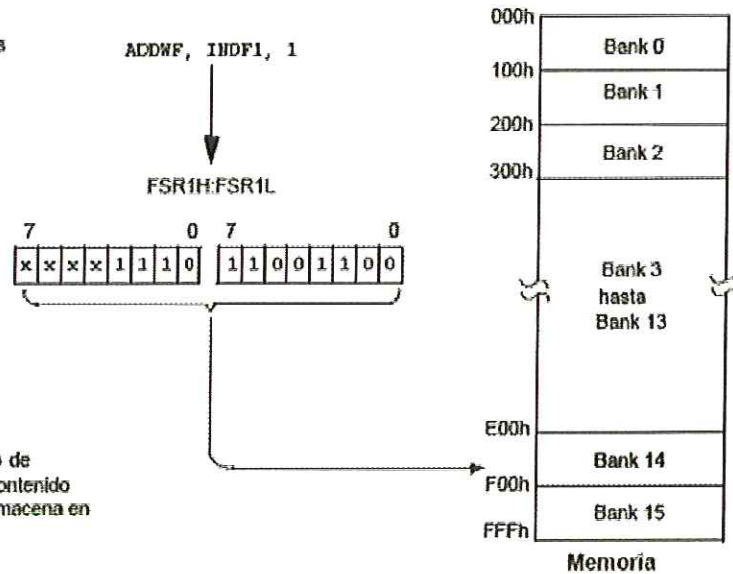
En ellos se almacenan direcciones de 12 bits, por lo tanto los 4 bits más significativos no se utilizan. El direccionamiento indirecto se indica en la instrucción con un conjunto de registros operandos INDF0, INDF1 e INDF2 considerados como virtuales (no existen físicamente).

La instrucción usa uno de los registros de direccionamiento indirecto como operando...

...usa los 12-bit almacenados en el el FSR asociado con el registro...

...para determinar la dirección que se usa en la operación.

En este caso, el FSR1 contiene ECCh. Esto significa que el contenido de la posición ECCh se sumará con el contenido del registro WREG y el resultado se almacena en ECCh.



Con el INDF, se utiliza la dirección contenida en el FSR asociado, sin modificar dicha dirección. Además de los operandos INDF para cada par de registros FSR, existen otras cuatro opciones adicionales. Como en el caso del INDF, estos son operandos virtuales porque no existen físicamente. Al utilizar estos operandos se utiliza como dirección la contenida en el FSR asociado y además se realiza una determinada acción:

- POSTDEC: Usa la dirección en el FSR y luego la decremента en '1'.
- POSTINC: Usa la dirección en el FSR y luego la incrementa en '1'.
- PREINC: Primero incrementa la dirección en '1' y luego la utiliza en la operación.
- PLUSW: Suma el valor con signo del WREG al del FSR y usa el resultado en la operación.

### (BANK 1) CON INDIRECT ADDRESSING

```

LFSR   FSR0, 100h ;
NEXT   CLRF   POSTINC0 ; Clear, pone a '0'
        ; luego
        ; inc puntero
        BTFSC FSR0H, 1 ; Se acabo el
        ; Bank1?
        BRA   NEXT     ; NO, clear siguiente
CONTINUE ; YES, continúa
    
```

## CARACTERISTICAS ESPECIALES

Los PIC18F2455/2550/4455/4550 incluyen varias características especiales pensadas para maximizar las prestaciones y minimizar el costo de los diferentes componentes externos. Estos elementos permiten además ahorrar energía y proteger el código frente a copia.

Los elementos que integra son:

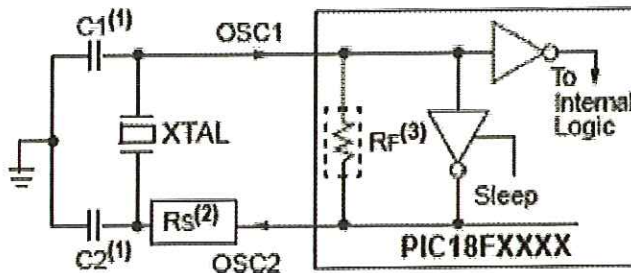
- Varias fuentes de osciladores diferentes.
- Diversos tipos de RESET:
  - Por conexión de la alimentación. (Power on reset, POR).
  - Por caída de tensión. (Brown-out reset, BOR), con:
    - Power-up timer.(PWRT).
    - Oscillator start-up timer.(OST).
- Varias fuentes de interrupción, con dos niveles de prioridad.
- Temporizador Watchdog.
- Varios modos de bajo consumo.
- Monitor de fallo del reloj.
- Start-up de dos velocidades.
- Protección de código.
- Posiciones de identificación ID.
- Circuitería de programación serie en circuito.
- Los PIC 18F2455/2550/4455/4550 incorporan múltiples fuentes de reloj. Estas fuentes se dividen en tres tipos:

**Osciladores primarios:** son los cristales externos, que pueden ir acompañados de los osciladores internos. El modo exacto se define mediante los bits de configuración FOSC3:FOSC0 y hay 12 combinaciones.

**Osciladores secundarios:** son las fuentes externas conectadas al oscilador del Timer 1, estas fuentes pueden seguir funcionando aunque el micro esté en modo de bajo voltaje.

**Bloque del oscilador interno:** se utiliza como fuente de reloj para el modos debajo consumo.

### CONFIGURACION DEL OSCILADOR EXTERNO

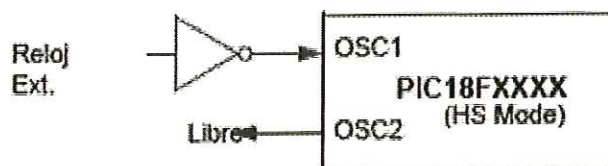


A continuación se presentan los valores sugeridos para los condensadores:

Tipo de Oscilador	Frecuencia del Cristal	Valor Capacitor	
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Una capacidad mayor produce mayor estabilidad del cristal pero también un tiempo de arranque mayor. La resistencia Rs puede ser necesaria para no sobre alimentar al cristal.

### CONFIGURACION MEDIANTE RELOJ EXTERNO





El pin OSC2 queda libre en el modo HS, saca FOSC/4 en modo EC y es RA6 en modo ECPIO.

## **CONFIGURACION DE LOS OSCILADORES INTERNOS**

Existen dos señales de reloj internas que permiten el funcionamiento del micro sin necesidad de oscilador externo:

El INTOSC es de 8Mhz y puede ser dividido mediante el postscaler dentro de un rango de 31Khz a 4Mhz. La salida se habilita automáticamente cuando se selecciona una salida desde 125Khz hasta 8 Mhz.

El otro reloj es el INTRC que proporciona una señal de reloj de 31Khz. Se habilita cuando se selecciona como fuente de reloj del micro o cuando se activa:

Power-up timer. Fail-safe clock monitor. Watchdog. El Start-up de doble velocidad.

Debido a que la frecuencia de oscilación puede variar con la temperatura el software puede calibrar y ajustar dicha frecuencia, escribiendo en el registro OSCTUNE.

REGISTRO:		OSCTUNE OSCILLATOR TUNING REGISTER						
	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	INTSRC	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
	bit 7							bit 0
bit 7	INTSRC: Bit de selección de la Frecuencia Interna del Oscilator 1 = 31.25 kHz el reloj se deriva de los 8 MHz del INTOSC (divide-por-256) 0 = 31 kHz el reloj se deriva del INTRC							
bit 6-5	No implementado: Se lee como '0'							
bit 4-0	TUN4:TUN0: Bits de ajuste de la frecuencia 01111 = Máxima frecuencia . . . 00001 00000 = Frecuencia central. El módulo del oscilador corre a la frecuencia calibrada. 11111 . . . 10000 = Mínima frecuencia							

Los bits OSTS, IOFS y T1RUN indican que fuente de reloj está actualmente suministrando la fuente de reloj.

- OSTS indica que está seleccionado el oscilador primario en algunos de sus modos y está proporcionando la señal de reloj.
- IOFS indica que el oscilador INTOSC está estabilizado y proporciona la señal de reloj.
- OSTS e IOFS se pueden activar a la vez si el oscilador es primario e interno.
- T1RUN (T1CON<6>) indica que el oscilador del timer 1 está proporcionando la señal de reloj en modo de reloj secundario.

Si ninguno de los bits está activo el oscilador INTRC es el que proporciona la señal de reloj o es el oscilador interno pero acaba de arrancar y todavía no es estable.

## RESET

Existen varios tipos de resets:

- #MCLR. Produce un reset mediante un impulso a nivel bajo en este pin.
- Power on Reset (POR), el reset se produce porque la tensión de alimentación sube por encima de 0,7 Voltios.
- WDT. Desbordamiento del Watchdog.

- Instrucción RESET.
- Pila llena.
- Pila vacía.
- Programable Brown-out reset. (BOR). Se produce un reset si la tensión de alimentación cae por debajo del valor VBOR durante un tiempo mayor de TBOR. El chip queda en reset durante un todo el tiempo en el que la alimentación está por debajo de VBOR. El valor exacto de VBOR se controla con los bits BORV1:BORV0.

### Configuraciones de los temporizadores de resets:

Hay 3 temporizadores separados en el chip cuya función es mantener el reset:

- Power-up timer (PWRT). Es un contador de 11bits que usa como fuente el INTRC. Esto da como resultado un periodo de desbordamiento de  $2048 \times 32 \mu\text{s} = 65.5 \text{ ms}$ . Se habilita con el bit PWRTEN
- Oscillator Start-up (OST). Proporciona 1024 ciclos después del PWRT para que se estabilice el oscilador. Sólo se habilita cuando el reloj proviene de un oscilador primario externo: modos HS, XT, HSPLL, XTPLL, INTHS e INTXT.
- PLL Lock-Time-out. Es un periodo necesario para que el PLL se sincronice con la señal de reloj del oscilador principal. Es típicamente 2 ms.

Oscillator Configuration	Power-up <sup>(2)</sup> y Brown-out		Al salir de bajo consumo
	PWRTEN = 0	PWRTEN = 1	
HS, XT	$66 \text{ ms}^{(1)} + 1024 \text{ Tosc}$	1024 Tosc	1024 Tosc
HSPLL, XTPLL	$66 \text{ ms}^{(1)} + 1024 \text{ Tosc} + 2 \text{ ms}^{(2)}$	$1024 \text{ Tosc} + 2 \text{ ms}^{(2)}$	$1024 \text{ Tosc} + 2 \text{ ms}^{(2)}$
EC, ECIO	$66 \text{ ms}^{(1)}$	—	—
ECPLL, ECPIO	$66 \text{ ms}^{(1)} + 2 \text{ ms}^{(2)}$	$2 \text{ ms}^{(2)}$	$2 \text{ ms}^{(2)}$
INTIO, INTCKO	$66 \text{ ms}^{(1)}$	—	—
INTHS, INTXT	$66 \text{ ms}^{(1)} + 1024 \text{ Tosc}$	1024 Tosc	1024 Tosc

Note 1: 66 ms (65.5 ms).  
2: 2 ms.

Para determinar el tipo de reset que se ha producido podemos consultar el estado del registro RCON:

REGISTRO: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1 <sup>(1)</sup>	U-0	R/W-1	R-1	R-1	R/W-0 <sup>(2)</sup>	R/W-0
IPEN	SBOREN	—	RI	TO	PD	POR	BOR
bit 7							bit 0

bit 7 IPEN: Bit de habilitación de la prioridad en las interrupciones  
 1 = Habilita los niveles de prioridad  
 0 = Deshabilita los niveles de prioridad

bit 6 SBOREN: Bit de habilitación software del BOR  
 If BOREN1:BOREN0 = 01;  
 1 = BOR habilitado  
 0 = BOR deshabilitado  
 If BOREN1:BOREN0 = 00, 10 OR 11;  
 Bit se lee como '0'.

bit 5 No implementado: Se lee como '0'

bit 4 RI: RESET Instruction Flag bit  
 1 = No se ha ejecutado la instrucción de RESET  
 0 = El RESET se debe a que se ha ejecutado la instrucción de RESET

bit 3 TO: Watchdog Time-out Flag bit  
 1 = Se pone a uno por un power-up, o las instrucciones CLRWDT o SLEEP  
 0 = El reset se debe a un desbordamiento del WDT

bit 2 PD: Power-Down Detection Flag bit  
 1 = Activo por un power-up o por la instrucción CLRWDT  
 0 = Se pone a cero al ejecutar la instrucción SLEEP

bit 1 POR: Power-on Reset Status bit<sup>(2)</sup>  
 1 = Un Power-on Reset no ha ocurrido  
 0 = Un Power-on Reset ha ocurrido

bit 0 BOR: Brown-out Reset Status bit  
 1 = Un Brown-out no ha ocurrido  
 0 = Un Brown-out Reset ha ocurrido

## BAJO CONSUMO

En estos modos el reloj está activo para el micro y los periféricos y el micro está ejecutando instrucciones. La diferencia entre ellos es la fuente de la señal de reloj:

- PRI\_RUN: es el modo normal de ejecución, donde está seleccionado el oscilador primario.



- SEC\_RUN: es el modo compatible para el "clock switching", en el que la CPU y los periféricos reciben la señal de reloj proveniente del Timer 1. Esta opción permite un menor consumo con un reloj muy fiable.
- RC\_RUN: la CPU y los periféricos reciben la señal de reloj desde el oscilador interno usando el multiplexor INTOSC y el reloj primario se apaga. Cuando se usa la fuente del INTRC se conserva la mayor cantidad de potencia mientras el micro sigue ejecutando instrucciones.

Modo	OSCCON Bits		Modo del reloj		Reloj habilitado y fuente del Oscilador
	IDLEN <sup>(1)</sup>	SCS1:SCS0	CPU	Periféricos	
Sleep	0	N/A	Off	Off	Ninguno – la ejecución de intruc. deshabilitada
PRI_RUN	N/A	00	Clocked	Clocked	Primario. Este es el modo normal de ejecución.
SEC_RUN	N/A	01	Clocked	Clocked	Secundario – Timer1 oscilador
RC_RUN	N/A	1x	Clocked	Clocked	Interno oscilador <sup>(2)</sup>
PRI_IDLE	1	00	Off	Clocked	Primario –
SEC_IDLE	1	01	Off	Clocked	Secundario – Timer1 oscillator
RC_IDLE	1	1x	Off	Clocked	Interno oscilador <sup>(2)</sup>

Note 1: IDLEN refleja su valor cuando se ejecuta la instrucción SLEEP.

2: Incluye INTOSC e INTRC.

**Modos de funcionamiento en bajo consumo.** En estos modos el reloj está activo sólo para los periféricos y el micro deja de ejecutar instrucciones. En los modos IDLE se entra ejecutando la instrucción SLEEP y con el bit IDLEN activo.

**PRI\_IDLE:** El oscilado primario no se apaga, pero el micro deja de ejecutar instrucciones. La vuelta al modo normal de ejecución es muy rápida ya que no es necesario esperar a la inicialización del oscilador.

**SEC\_IDLE:** El oscilador del Timer 1 sigue funcionando aunque el micro se para.

**RC\_IDLE:** la se para CPU y los periféricos reciben la señal de reloj desde el oscilador interno usando el multiplexor INTOSC. Cuando se usa la fuente del INTRC se conserva la mayor cantidad de potencia mientras el micro sigue ejecutando instrucciones.

Modo	OSCCON Bits		Modo del reloj		Reloj habilitado y fuente del Oscilador
	IDLEN <sup>(1)</sup>	SCS1:SCS0	CPU	Periféricos	
Sleep	0	N/A	Off	Off	Ninguno – la ejecución de intruc. deshabilitada
PRI_RUN	N/A	00	Clocked	Clocked	Primario. Este es el modo normal de ejecución.
SEC_RUN	N/A	01	Clocked	Clocked	Secundario – Timer1 oscilador
RC_RUN	N/A	1x	Clocked	Clocked	Interno oscilador <sup>(2)</sup>
PRI_IDLE	1	00	Off	Clocked	Primario –
SEC_IDLE	1	01	Off	Clocked	Secundario – Timer1 oscilator
RC_IDLE	1	1x	Off	Clocked	Interno oscilador <sup>(2)</sup>

Note 1: IDLEN refleja su valor cuando se ejecuta la instrucción SLEEP.

2: Incluye INTOSC e INTRC.

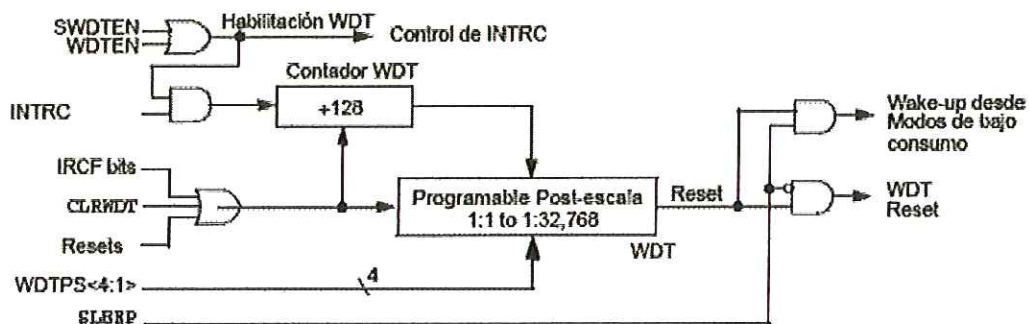
Modo de funcionamiento dormido. En este modo es similar al modo SLEEP de los dispositivos PIC16. Se entra en este modo ejecutando la instrucción SLEEP mientras el bit IDLEN está a cero. En este modo la CPU se para y los periféricos también. Todas las fuentes de reloj se paran salvo dos excepciones:

- Si el WDT está habilitado el INTRC sigue funcionando.
- Si el oscilador del Timer 1 está habilitado también sigue corriendo.
- Se sale de estos modos mediante una interrupción, un Reset o el WDT.
- Cualquiera de las fuentes de interrupción del micro puede sacarlo de estos modos. Si además el bit global de habilitación de GIE está activo el micro salta a ejecutar la subrutina de tratamiento de la interrupción. Sino se despierta y sigue la ejecución donde la había dejado.
- Un desbordamiento del WDT produce un reset si el micro esta en modo de ejecución. Si está en algún modo de bajo consumo sale de estos modos y sigue la ejecución en el punto donde estaba.
- Si se produce un reset el micro permanece en este estado hasta que el oscilador primario está preparado.

Microcontrolador reloj		Retraso	Señal de reloj Bit (OSCCON)
Antes de despertar	Después		
PRI_IDLE (PRI_IDLE mode)	XT, HS	None	OSTS
	XTPLL, HSPLL		
	EC		IOFS
	INTOSC <sup>(3)</sup>		
SEC_IDLE	XT, HS	$T_{OST}^{(4)}$	OSTS
	XTPLL, HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC	$T_{csd}^{(2)}$	IOFS
	INTOSC <sup>(2)</sup>	$T_{IOBST}^{(5)}$	
RC_IDLE	XT, HS	$T_{OST}^{(4)}$	OSTS
	XTPLL, HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC	$T_{csd}^{(2)}$	IOFS
	INTOSC <sup>(2)</sup>	None	
None (Sleep)	XT, HS	$T_{OST}^{(4)}$	OSTS
	XTPLL, HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC	$T_{csd}^{(2)}$	IOFS
	INTOSC <sup>(2)</sup>	$T_{IOBST}^{(5)}$	

## WATCHDOG (WDT)

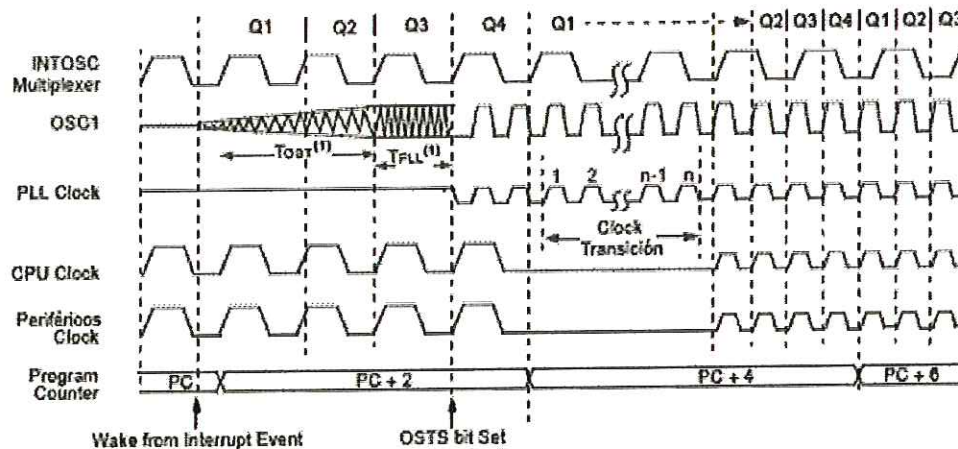
El Watchdog timer (WDT) se incrementa con el impulsos del reloj interno INTRC. Al habilitar el WDT se habilita también el oscilador INTRC. El periodo es de 4 ms y tiene la misma estabilidad que el oscilador INTRC. El periodo de 4 ms del WDT se multiplica por un postscaler de 16 bits. (Rango desde 4 ms hasta 2.18 min.) El WDT se limpia con las instrucciones SLEEP y CLRWDT.





## START-UP DE DOS VELOCIDADES

El start-up de dos velocidades permite minimizar el periodo de latencia desde el arranque del oscilador hasta la ejecución de código. Durante este periodo de arranque se permite el uso del oscilador interno INTRC. Se habilita mediante el bit IESO (CONFIG1H <7>) y solamente si el oscilador primario está en los modos XT, HS, XTPLL o HSPLL. Cuando se habilita después de un **Reset** o **salida del modo SLEEP** causan que el dispositivo se autoconfigure y seleccione el oscilador interno como fuente de reloj. Una vez que el periodo OST ya se ha acabado se vuelve al modo PRI\_RUN. En los demás modos de bajo consumo el start-up de dos velocidades no se utiliza y el dispositivo permanece con el reloj actual hasta que el oscilador primario se estabiliza.

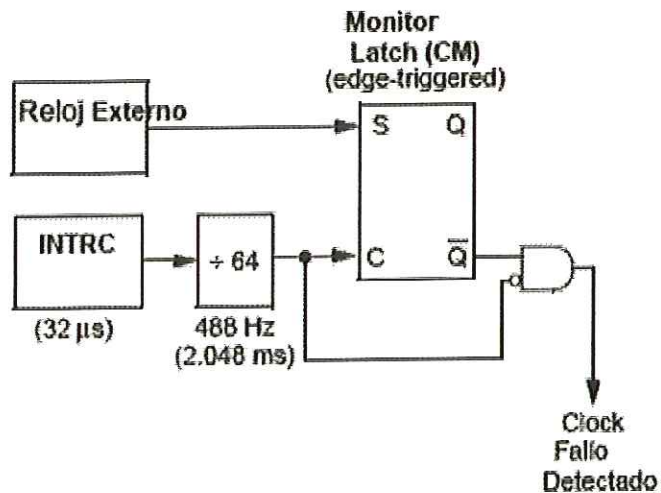


Nota 1:  $T_{OST} = 1024 T_{OSC}$ ;  $T_{PLL} = 2 \text{ ms (approx)}$ .

## MONITOR DEL RELOJ

El Monitor de reloj (FSCM) permite al micro continuar la operación aunque haya un fallo del oscilador tanto primario como secundario que actualmente está usando el micro. Este módulo automáticamente inicia el oscilador interno que actualmente esté seleccionado en el registro (OSCCON) como señal de reloj. Se habilita mediante el bit FCMEN (CONFIG1H <6>). El circuito de detección muestrea el reloj externo y el reloj interno mediante un latch.





Cuando se detecta un fallo:

- El reloj se cambia al oscilador interno.
- El WDT se resetea.
- Se produce una interrupción activando el bit OSCFIF (PIR2<7>)

## INTERRUPCIONES

Existen 20 o 21 fuentes de interrupción distintas:

- Por pines externos: INT0, INT1, INT2, RB. Puerto paralelo: SPP.
- Temporizadores: TMR0, TMR1, TMR2, TMR3.
- Módulos CCP: CCP1, CCP2.
- EUSART: RC, TX.
- MSSP: SSP.
- Conversor A/D: AD.
- Fallo del oscilador: OSCF.
- Comparador: CM.
- Módulo USB: USB.
- Escritura EEPROM/FLASH: EE.
- Colisión del bus: BCL.
- Módulo de detección de voltaje: HLVD.

### **Cuando se produce una interrupción:**

1. Se salva la dirección de vuelta en la pila y se deshabilitan las interrupciones de forma automática (bit GIE o GIEH/GIEL).
2. Se salta a la dirección del vector de interrupción (000008h o 000018h)
3. Una vez se ejecuta la subrutina de interrupción. Ésta debe determinar la fuente de interrupción por software, mediante el muestreo de los bits de solicitud de interrup.
4. El bit de solicitud de interrupción debe limpiarse por software.
5. Se ejecuta la instrucción RETFIE, activa el bit de habilitación (bit GIE o GIEH/GIEL) y desapila la dirección de vuelta.

### **MODOS DE FUNCIONAMIENTO**

**Sin prioridades:** es el funcionamiento normal similar al modo de los PIC16.

- En este caso el bit IPEN es '0' (RCON<7>).
- GIE habilita de forma global todas las interrupciones, PEIE habilita la de los periféricos.
- Todas las interrupciones saltan a la dirección 000008h.

**Con prioridades:** existen dos prioridades alta y baja.

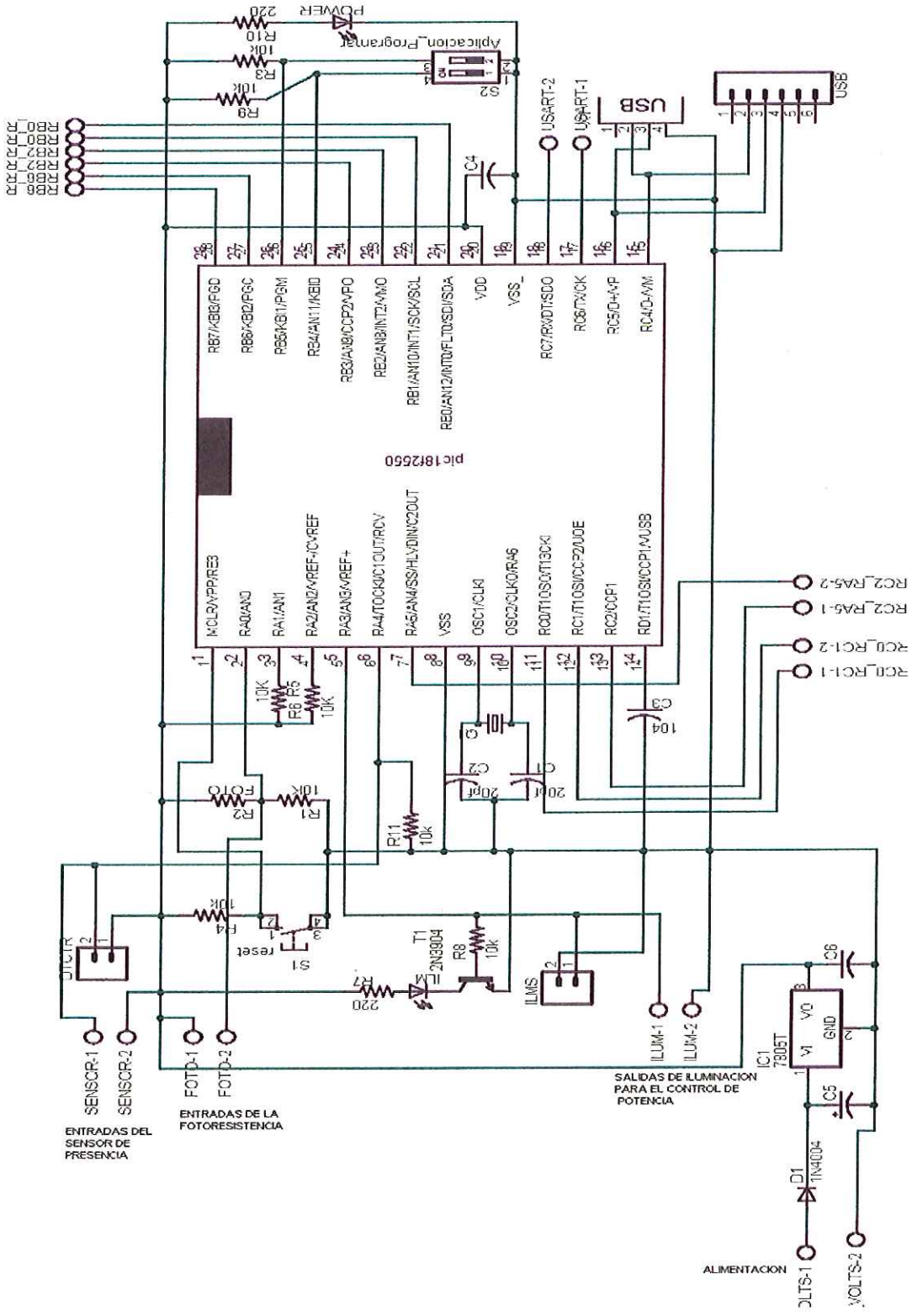
- En este caso el bit IPEN es '1' (RCON<7>).
- GIE=GIEH habilita las interrupciones de alta prioridad, PEIE=GIEL las de baja.
- Cada interrupción se puede configurar como de alta o baja prioridad.
- Las interrupciones de alta prioridad saltan a la dirección 000008h las de baja a 000018h.
- Una interrupción de alta puede interrumpir a una de baja. Una de baja no puede interrumpir otra interrupción

Cada fuente de interrupción tiene tres bits de control:

- El bit de habilitación que permite que esa fuente de interrupción sea procesada. (xIE)
- El bit de solicitud que indica cuando una interrupción se ha producido. (xIF)
- El bit de prioridad para configurarla como de alta o baja.(xIP)
- Existen 10 registros para el control de las interrupciones:
- Control y habilitación: RCON, INTCON, INTCON2, INTCON3, PIE1 y PIE2.
- Solicitud: PIR1, PIR2.

Prioridad: IPR1, IPR2.

# ANEXO 2. CIRCUITO ELECTRONICO - SISTEMA DE CONTROL



Circuito Electrónico Sistema de Control



### ANEXO 3 PROGRAMA DEL MICROCONTROLADOR PIC18F2550 (SISTEMA DE CONTROL)

```
/******  
*  
*      Microchip USB C18 Firmware Version 1.0  
*  
*****  
* FileName:      user.c  
* Dependencies:  See INCLUDES section below  
* Processor:     PIC18  
* Compiler:      C18 2.30.01+  
* Company:       Microchip Technology, Inc.  
*  
* Acuerdo de Licencia de software  
*  
* El software proporcionado por Microchip para su PICmicro® Microcontrolador se  
* piensa para el uso solamente y exclusivamente en microcontroladores de productos  
* Microchip. El software esta protegido bajo las leyes del derechos de propiedad  
* literaria . Todos los derechos son reservados.  
* Cualquier uso en la violación de las restricciones anteriores puede sujetar al usuario,  
* sanciones delictivas bajo las leyes aplicables, así como la obligación civil para  
* romper los términos y condiciones de esta licencia.  
*  
*  
* Autor          Fecha      Comment  
* ~~~~~  
* Rawin Rojvanit  11/19/04   Original.  
*****  
/** I N C L U D E S  
*****  
#include <p18cxxx.h>  
#include <usart.h>  
#include "system\typedefs.h"  
  
#include "system\usb\usb.h"  
  
#include "io_cfg.h"      // Pines de entrada y salida  
#include "user\user.h"  
#include "user\temperature.h"  
  
/** V A R I A B L E S  
*****  
#pragma udata  
byte old_sw2,old_sw3;  
byte counter;  
byte trf_state;  
byte temp_mode;
```

```
byte umbral_luz;           //Nivel de luz en el cual se activa la iluminación
byte segundos;           //segundos que debe estar encendida la luz al detectar movimiento
```

```
DATA_PACKET dataPacket;
```

```
byte pTemp;
byte valid_temp;
word temp_data[30];
```

```
// Timer0 - 1.39 segundos
// Fosc/4 = 12MHz
// Use /256 prescalador, esto trae la frecuencia del contador por debajo de 46,875 Hz
```

```
//f=12Mhz/256/46875=1Hz
```

```
// Timer0 should = 65536 - 46875 = 18661 or 0x48E5
```

```
#define TIMER0L_VAL    0xE5
```

```
#define TIMER0H_VAL    0x48
```

```
/** PRIVATE PROTOTYPES
```

```
*****
```

```
void BlinkUSBStatus(void);
BOOL Switch2IsPressed(void);
BOOL Switch3IsPressed(void);
void ResetTempLog(void);
void ReadPOT(void);
void ServiceRequests(void);
```

```
/** ***** Mi control automático de luz *****
```

```
void verificar_presencia(void);
void VerificarTimepo(void);
int Presencia(void);
```

```
// For board testing purpose only
```

```
void PICDEMFSUSBDemoBoardTest(void);
```

```
/** DECLARATIONS
```

```
*****
```

```
#pragma code
```

```
void UserInit(void)
```

```
{
```

```
// mInitAllLEDs();
```

```
  mInitAllSwitches();
```

```
  old_sw2 = sw2;
```

```
  old_sw3 = sw3;
```

```

// InitTempSensor();
  mInitPOT();
  ADCON2bits.ADFM = 0;

  ResetTempLog();           //Inicial los contadores a Cero
  temp_mode = TEMP_REAL_TIME;

  /* Init Timer0 for data logging interval (every 1 second) */
  T0CON = 0b10010111;
/*
Configuración del timer 0
Modo de 16 bits
Fuente de reloj rececuancia del micro/4 =48Mhz/4
Prescalador en 256
*/

T0CONbits.TMR0ON = 1;      //Contador encendido
T0CONbits.T08BIT = 0;     // modo 16-bit
T0CONbits.T0CS = 0;      // Selecciona Fosc/4
T0CONbits.PSA = 0;      // Asigna prescalador (el valor por defecto es /256)

/* Timer0 esta ya habilitado por defecto*/

//Iniciación del sensor de presencia
tris_presencia=INPUT_PIN;
segundos=0;

} //end UserInit

/*****
* Function:      void ProcessIO(void)
*
* Precondition:  None
*
* Input:         None
*
* Output:        None
*
* Side Effects:  None
*
* Overview:      Esta función es un poseedor del lugar para otras rutinas del usuario.
*                Es una mezcla de tareas tanto de USB como no-USB
* Note:          None
*****/
void ProcessIO(void)
{
  BlinkUSBStatus();
  // Tareas de aplicación del usuario USB

```

```
//if((usb_device_state < CONFIGURED_STATE)||(UCONbits.SUSPND==1)) return;
```

```
ServiceRequests();
```

```
VerificarTimepo();
```

```
ReadPOT();
```

```
//Lee el nivel de luz actual
```

```
if(ADRESH<umbral_luz) //Enciende la luz si es necesario  
{  
    verificar_presencia();  
    if(segundos>0) luz=1;}
```

```
if (segundos==0)
```

```
{  
    luz=0; //Apaga las luces si hay suficiente luz no queda tiempo  
}
```

```
}//end ProcessIO
```

```
void VerificarTimepo(void)
```

```
{  
    if(INTCONbits.TMR0IF)  
    {  
        INTCONbits.TMR0IF=0;  
        if(segundos>0)  
            {segundos=segundos-1;}  
    };  
}
```

```
void verificar_presencia(void)
```

```
{  
    if (Presencia())  
    {  
        segundos=10;  
    }  
}
```

```
//Presencia(void) retorna 1 cuando el sensor de presencia a detectado movimiento
```

```
int Presencia(void)
```

```
{  
    return !presencia;  
}
```



```

void ResetTempLog(void)
{
    pTemp = 0;
    valid_temp = 0;
} //end ResetLog

void ReadPOT(void)
{
    ADCON0bits.GO = 1;          // Comienza la conversión DC
    while(ADCON0bits.NOT_DONE); // Espera por la conversión
    return;
} //end ReadPOT

void ServiceRequests(void)
{
    byte index;

    if(USBGenRead((byte*)&dataPacket,sizeof(dataPacket)))
    {
        counter = 0;
        switch(dataPacket.CMD)
        {
            case READ_VERSION:
                //dataPacket._byte[1] is len
                dataPacket._byte[2] = MINOR_VERSION;
                dataPacket._byte[3] = MAJOR_VERSION;
                counter=0x04;
                break;

            case ID_BOARD:
                counter = 0x01;
                if(dataPacket.ID == 0)
                {
                    // mLED_3_Off();mLED_4_Off();
                }
                else if(dataPacket.ID == 1)
                {
                    // mLED_3_Off();mLED_4_On();
                }
                else if(dataPacket.ID == 2)
                {
                    // mLED_3_On();mLED_4_Off();
                }
                else if(dataPacket.ID == 3)
                {
                    // mLED_3_On();mLED_4_On();
                }
                else

```

```

        counter = 0x00;
        break;

case UPDATE_LED:
    // LED1 & LED2 son usados como indicadores de evento USB.
    if(dataPacket.led_num == 3)
    {
//        mLED_3 = dataPacket.led_status;
        counter = 0x01;
    }//end if
    else if(dataPacket.led_num == 4)
    {
//        mLED_4 = dataPacket.led_status;
        counter = 0x01;
    }//end if else
    break;

case SET_TEMP_REAL:
    temp_mode = TEMP_REAL_TIME;
    ResetTempLog();
    counter = 0x01;
    break;

case RD_TEMP:
    ReadPOT();
    dataPacket._byte[1] = ADRESL;
    dataPacket._byte[2] = ADRESH;
    counter=0x03;
    break;

case SET_TEMP_LOGGING:
    ReadPOT();
    dataPacket._byte[1] = ADRESL;
    dataPacket._byte[2] = ADRESH;
    counter=0x01;
    break;

case RD_TEMP_LOGGING:

    break;

case RD_POT:
    ReadPOT();
    dataPacket._byte[1] = ADRESL;
    dataPacket._byte[2] = ADRESH;
    counter=0x03;
    break;

case RESET:
    Reset();

```

```

        break;
        case SET_UMBRAL:
            umbral_luz = dataPacket._byte[6];
            ReadPOT();
            dataPacket._byte[1] = ADRESL;
            dataPacket._byte[2] = ADRESH;

            dataPacket._byte[5] = umbral_luz;
            if(luz) dataPacket._byte[4] =1;
            else dataPacket._byte[4] =0;
            dataPacket._byte[3] = segundos;

            counter=0x06;
            break;

        default:
            break;
    } //end switch()
    if(counter != 0)
    {
        if(!mUSBGenTxIsBusy())
            USBGenWrite((byte*)&dataPacket,counter);
    } //end if
} //end if

} //end ServiceRequests

/*****
* Function:      void BlinkUSBStatus(void)
*
* Precondition:  None
*
* Input:         None
*
* Output:        None
*
* Side Effects:  None
*
* Overview:      BlinkUSBStatus prende y apaga los LEDs correspondientes al
*                estado USB del dispositivo.
*
* Note:          mLED puede ser encontrado en io_cfg.h
*                usb_device_state es declarado en usbmmmap.c y es modificado
*                en usbdv.c, usbcrtltrf.c, y usb9.c
*****/

```

```

void BlinkUSBStatus(void)
{

```

```

static word led_count=0;

if(led_count == 0)led_count = 10000U;
led_count--;

// #define mLED_Both_Off()      {mLED_1_Off();mLED_2_Off();}
// #define mLED_Both_On()      {mLED_1_On();mLED_2_On();}
// #define mLED_Only_1_On()    {mLED_1_On();mLED_2_Off();}
// #define mLED_Only_2_On()    {mLED_1_Off();mLED_2_On();}

if(UCONbits.SUSPND == 1)
{
    if(led_count==0)
    {
//      mLED_1_Toggle();
//      mLED_2 = mLED_1;    // Ambos parpadean al mismo tiempo
    }//end if
    }
else
{
    if(usb_device_state == DETACHED_STATE)
    {
//      mLED_Both_Off();

        PICDEMFSUSBDemoBoardTest();
    }
    else if(usb_device_state == ATTACHED_STATE)
    {
//      mLED_Both_On();
    }
    else if(usb_device_state == POWERED_STATE)
    {
//      mLED_Only_1_On();
    }
    else if(usb_device_state == DEFAULT_STATE)
    {
//      mLED_Only_2_On();
    }
    else if(usb_device_state == ADDRESS_STATE)
    {
        if(led_count == 0)
        {
//          mLED_1_Toggle();
//          mLED_2_Off();
        }//end if
    }
    else if(usb_device_state == CONFIGURED_STATE)
    {
        if(led_count==0)
        {

```



```

//      mLED_1_Toggle();
//      mLED_2 = !mLED_1;    // parpadeo alternativo
    }//end if
  }//end if(...)
} //end if(UCONbits.SUSPND...)

} //end BlinkUSBStatus

BOOL Switch2IsPressed(void)
{
  if(sw2 != old_sw2)
  {
    old_sw2 = sw2;          // Guarda Nuevo valor
    if(sw2 == 0)           // Si presiono
      return TRUE;        // Fue presionado
  } //end if
  return FALSE;           // No fue presionado
} //end Switch2IsPressed

BOOL Switch3IsPressed(void)
{
  if(sw3 != old_sw3)
  {
    old_sw3 = sw3;          // Guarda Nuevo valor
    if(sw3 == 0)           // Si presiono
      return TRUE;        // Fue presionado
  } //end if
  return FALSE;           // No fue presionado
} //end Switch3IsPressed

void TXbyte(byte data)
{
  while(TXSTAbits.TRMT==0);
  TXREG = data;
} //end TXbyte

void PICDEMFSUSBDemoBoardTest(void)
{
  byte temp;

  //PICDEM FS USB Demo Board Test Procedimiento:
  if(Switch2IsPressed())
  {
    //LEDs and push buttons testing
    //  mLED_1_On();
    while(!Switch2IsPressed());
    //  mLED_1_Off();
    //  mLED_2_On();
    while(!Switch3IsPressed());
    //  mLED_2_Off();
  }
}

```

```

// mLED_3_On();
  while(!Switch3IsPressed());
// mLED_3_Off();
// mLED_4_On();
  while(!Switch3IsPressed());
// mLED_4_Off();

//RS-232 Setup
SSPCON1 = 0; // Asegura que SPI esta desactivado
TRISCbits.TRISC7=1; // RX
TRISCbits.TRISC6=0; // TX
SPBRG = 0x71;
SPBRGH = 0x02; // 0x0271 for 48MHz -> 19200 baud
TXSTA = 0x24; // TX habilita BRGH=1
RCSTA = 0x90; // continuou RX
BAUDCON = 0x08; // BRG16 = 1
temp = RCREG; // buffer vacio
temp = RCREG; // buffer vacio

//RS-232 Tx & Rx Tests
while(!Switch3IsPressed());
TXbyte('R');
TXbyte('S');
TXbyte('-');
TXbyte('2');
TXbyte('3');
TXbyte('2');
TXbyte(' ');
TXbyte('T');
TXbyte('X');
TXbyte(' ');
TXbyte('T');
TXbyte('e');
TXbyte('s');
TXbyte('t');
TXbyte(' ');
TXbyte('O');
TXbyte('K');
TXbyte(',');
TXbyte(' ');
TXbyte('P');
TXbyte('r');
TXbyte('e');
TXbyte('s');
TXbyte('s');
TXbyte(' ');
TXbyte('');
TXbyte('r');
TXbyte('');
TXbyte(',');

```

```
while(PIR1bits.RCIF==0);    //Espera por los datos desde RS232
if(RCREG == 'r')
{
    TXbyte(' ');
    TXbyte('R');
    TXbyte('X');
    TXbyte(' ');
    TXbyte('T');
    TXbyte('e');
    TXbyte('s');
    TXbyte('t');
    TXbyte(' ');
    TXbyte('O');
    TXbyte('K');
} //end if
UserInit();                // Re-inicializa por defecto user fw
//Test phase 1 done
} //end if
} //end PICDEMFSUSBDemoBoardTest()
```

```
/** EOF user.c
```

```
*****/
```

## ANEXO 4 CIN PARA CONEXIÓN ENTRE LABVIEW Y SISTEMA DE CONTROL

```
// CIN source file
```

```
/* CIN para control del sistema USB automático de iluminación. Este CIN monitorea el nivel de iluminación y ajusta el umbral de sensibilidad para activar la iluminación
```

Un CIN, llamado así por sus siglas "Nodo de interfaz de código" es una de las herramientas con que cuenta el LabView y que permite inserta código en C en una aplicación escrita en labview, para información detallada acerca de la programación y compilación de CINs remítase al manual "Using External Code in LabVIEW" que se distribuye con cualquier versión de labview\*/

```
#include "extcode.h"
```

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

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

```
#include <windows.h>
```

```
#include <stdlib.h>
```

```
#include "stdafx.h" // #include <stdio.h>
```

```
#include "windows.h"
```

```
#include "mpusbapi.h"
```

```
#define false 0
```

```
// Global Vars
```

```
char vid_pid[50]=""; //Cadena de caracteres para insertar la identificación del  
//dispositivo USB que se esta controlando, la identificación  
//del sistema de control de luz es "vid_04d8&pid_000b" el cual es  
//el mismo del "Demo Application Firmware"
```

```
char out_pipe[]= "\\MCHP_EP1";
```

```
char in_pipe[]= "\\MCHP_EP1";
```



DWORD temp;

/\*Para nuestra aplicación los datos se envían y reciben en forma de paquetes con un tamaño máximo de 64 bytes conservando el siguiente formato:

byte	Función
0	Comando
1	Longitud del paquete
2	Endpoint
3-64	Datos

\*/

```
BYTE send_buf[64];           //Buffer para el envío de paquetes
    BYTE receive_buf[64];    // Buffer para la recepción de paquetes
DWORD RecvLength;           //Puntero al número de datos a leer
    BYTE DataLen;            //Longitud de los datos
    BYTE EEAddress;          //Endpoint
    int i;                    //Variable para contador
    int di;                   //Variable para contador
    BYTE CurrentCMD;         //Comando a enviar al dispositivo USB
LVBoolean err;
char msg[250];
long len;
    BYTE umbral_luz;         //Umbral para encender la luz 1-255
HINSTANCE libHandle;
HANDLE myOutPipe;
HANDLE myInPipe;
```

/\* A continuación aparecen las definiciones de las variables que conectan labview y el código en C, estas definiciones se generan automáticamente desde labview.

```
+++++
+ Nombre   E/S      Comentario
+++++
+ max_count
+ id       E        :Identificación del dispositivo
+ arg1     S        :Arreglo de bytes de salida
+ EEP      E        :Endpoint
+ Error    S        :Bandera de Error, se pone en 1 si hay un error
+ string   S        :Mensaje de texto acerca del status de la comunicación
+ Umbral   E        :Byte 0-255 que indica el umbral de sensibilidad
+++++
```

\*/

/\* Typedefs \*/

```
typedef struct {
    int32 dimSize;
    uInt8 elt[1];
} TD1;
typedef TD1 **TD1Hdl;
```

```
MgErr CINRun(uInt32 *max_count, LStrHandle *id, TD1Hdl arg1, uInt8 *EEP,
             LVBoolean *Error, LStrHandle string,uInt8 *Umbral);
```

```
MgErr CINRun(uInt32 *max_count, LStrHandle *id, TD1Hdl arg1, uInt8 *EEP,
             LVBoolean *Error, LStrHandle string,uInt8 *Umbral)
{
```

//-----

```
// Prototypes
void ReadEEData(void);
```

```
void WriteEEData(void);
void ClearEEData(void);
uint32 GetSummary(void);
void LoadDLL(void);
void GetUSBDemoFWVersion(void);
void ReadPot(void);
void SetUmbral(void);
```

```
//La función SendReceivePacket() envía un paquete de datos al micro y recibe otro
DWORD SendReceivePacket(BYTE *SendData, DWORD SendLength, BYTE
*ReceiveData, DWORD *ReceiveLength, UINT SendDelay, UINT ReceiveDelay);
void CheckInvalidHandle(void);
```

```
    BOOLEAN bQuit;
    DWORD selection;
    bQuit = false;
    err=0;
    umbral_luz=*Umbral;
    strcpy(msg,"All Ok!"); //Inicia los mensajes de estado de la comunicación a"OK"
```

```
    sprintf(vid_pid, "%H", id); //Copia la cadena de caracteres "id" que entra
desde labview a la cadena vid_pid de C
```

```
LoadDLL(); // Carga la dll de comunicación suministrada por Microchip
```

```
// Always a good idea to initialize the handles
```

```
myOutPipe = myInPipe = INVALID_HANDLE_VALUE;
```

```
EEAddress=*EEP; //Inicia el endpoint
SetUmbral(); //Función para establecer el umbral y leer el nivel de luz
//Copiar los datos del paquete recibido al buffer de salida de labview
for(i=0;i<30;i++)
    (**arg1).elt[i]=receive_buf[i];
```

```

//----- Reporte -----
    //procedimiento para generar el reporte de la comunicación
    *Error=err;           //Se envía a labview el estado de errores

    len=StrLen(msg);
    /* actualize la longitud modificada de LV string */
    LStrLen(*string)=len;
    /* Convierte la cadena de reporte de comunicación a un "LabView string" */
    MoveBlock(&msg, LStrBuf(*string), len);
//----- Fin Reporte -----

// Siempre verificar cerrar todas las handles antes de terminar!
if (myOutPipe != INVALID_HANDLE_VALUE) MPUSBClose(myOutPipe);
if (myInPipe != INVALID_HANDLE_VALUE) MPUSBClose(myInPipe);
myOutPipe = myInPipe = INVALID_HANDLE_VALUE;

// Cerramos la dll
if (libHandle != NULL) FreeLibrary(libHandle);

    return 0;

    return noErr;
}

uInt32 GetSummary(void)
{
    HANDLE tempPipe = INVALID_HANDLE_VALUE;
    DWORD count = 0;
    uInt32 max_count = 0;
    int i;

    max_count = MPUSBGetDeviceCount(vid_pid);

    count = 0;

```



```

for(i = 0; i < MAX_NUM_MPUSB_DEV; i++)
{
    tempPipe = MPUSBOpen(i,vid_pid,NULL,MP_READ,0);
    if(tempPipe != INVALID_HANDLE_VALUE)
    {
        printf("Instance Index # %d\r\n",i);
        MPUSBClose(tempPipe);
        count++;
    }
    if(count == max_count) break;
} //end for
printf("\r\n");
return max_count;
}

```

//Función para cargar la dll "mpusbapi.dll" suministrada por Microchip

void LoadDLL(void)

```

{
    libHandle = NULL;
    libHandle = LoadLibrary("mpusbapi");
    if(libHandle == NULL)
    {
        strcpy(msg,"Error loading mpusbapi.dll");
        err=1;
    }
    else
    {
        MPUSBGetDLLVersion=(DWORD(*)(void))\
            GetProcAddress(libHandle,"_MPUSBGetDLLVersion");
        MPUSBGetDeviceCount=(DWORD(*)(PCHAR))\
            GetProcAddress(libHandle,"_MPUSBGetDeviceCount");
        MPUSBOpen=(HANDLE*)(DWORD,PCHAR,PCHAR,DWORD,DWORD)\
            GetProcAddress(libHandle,"_MPUSBOpen");
    }
}

```

```

MPUSBWrite=(DWORD*)(HANDLE,PVOID,DWORD,PDWORD,DWORD)\
    GetProcAddress(libHandle,"_MPUSBWrite");
MPUSBRead=(DWORD*)(HANDLE,PVOID,DWORD,PDWORD,DWORD)\
    GetProcAddress(libHandle,"_MPUSBRead");

MPUSBReadInt=(DWORD*)(HANDLE,PVOID,DWORD,PDWORD,DWORD)\
    GetProcAddress(libHandle,"_MPUSBReadInt");

MPUSBClose=(BOOL*)(HANDLE))GetProcAddress(libHandle,"_MPUSBClose");

    if((MPUSBGetDeviceCount == NULL) || (MPUSBOpen == NULL) ||
        (MPUSBWrite == NULL) || (MPUSBRead == NULL) ||
        (MPUSBClose == NULL) || (MPUSBGetDLLVersion == NULL) ||
        (MPUSBReadInt == NULL))
        printf("GetProcAddress Error\r\n");
    }//end if else
} //end LoadDLL
//-----
//
// Una aplicación típica envía datos al dispositivo y espera una respuesta
// SendReceivePacket es una función que facilita esta operación
// SendData - puntero a los datos para enviarse
// SendLength - longitud de los datos para enviarse
// ReceiveData - Puntero al buffer que recibe los datos
// ReceiveLength - Puntero al numero de datos a leer
// SendDelay - time-out valor de MPUSBWrite operación en milliseconds
// ReceiveDelay - time-out valor para MPUSBRead operación en milliseconds
//
DWORD SendReceivePacket(BYTE *SendData, DWORD SendLength, BYTE
*ReceiveData,
    DWORD *ReceiveLength, UINT SendDelay, UINT ReceiveDelay)
{

```

```

DWORD SentDataLength;
DWORD ExpectedReceiveLength = *ReceiveLength;

if(myOutPipe != INVALID_HANDLE_VALUE && myInPipe !=
INVALID_HANDLE_VALUE)
{

if(MPUSBWrite(myOutPipe,SendData,SendLength,&SentDataLength,SendDelay))
{

if(MPUSBRead(myInPipe,ReceiveData, ExpectedReceiveLength,
ReceiveLength,ReceiveDelay))
{
if(*ReceiveLength == ExpectedReceiveLength)
{
return 1; // Success!
}
else if(*ReceiveLength < ExpectedReceiveLength)
{
return 2; // Partially failed, incorrect receive length
} //end if else
}
else
CheckInvalidHandle();
}
else
CheckInvalidHandle();
} //end if

return 0; // Operation Failed
} //end SendReceivePacket

```

```

//-----

void CheckInvalidHandle(void)
{
    if(GetLastError() == ERROR_INVALID_HANDLE)
    {
        // lo más probable que causa el error es que la board estaba desconectada..
        MPUSBClose(myOutPipe);
        MPUSBClose(myInPipe);
        myOutPipe = myInPipe = INVALID_HANDLE_VALUE;
    }//end if
    else
        strcpy(msg, "Error");

} //end CheckInvalidHandle
//-----

void SetUmbral(void)
{
    // primero se abren las data pipes...
    DWORD selection=0; // solo el índice cero existe

    myOutPipe = MPUSBOpen(selection,vid_pid,out_pipe,MP_WRITE,0);
    myInPipe = MPUSBOpen(selection,vid_pid,out_pipe,MP_READ,0);
    if(myOutPipe == INVALID_HANDLE_VALUE || myInPipe ==
INVALID_HANDLE_VALUE)
    {
        strcpy(msg, "Failed to open data pipes.");
        err=1;
        return;
    } //end if
}

```



```

/*
El paquete usado en esta aplicación se ensambla a continuación
*/

RecvLength;
    CurrentCMD=SET_UMBRAL; //Comando que se envía al sistema
    send_buf[0] = CurrentCMD; //El comando se inserta en el byte 0 de paquete
    send_buf[1] = 0x07; //Longitud de los datos que se esperan (Excepto
// 2 bytes de cabecera).
    send_buf[6] = umbral_luz; //Byte para ajustar el umbral de la luz
    RecvLength=2+send_buf[1];
    if(SendReceivePacket(send_buf,send_buf[1],receive_buf,&RecvLength,1000,1000)
== 1)
    {
        if(receive_buf[0] == CurrentCMD)
            strcpy(msg, "All Ok");
        else
            strcpy(msg, "upper cerita is error");
    }

// Cierra data pipes.
MPUSBClose(myOutPipe);
MPUSBClose(myInPipe);
myOutPipe = myInPipe = INVALID_HANDLE_VALUE;

} //end SetUmbral

```

## **ANEXO 5. PROGRAMA EN LABVIEW PARA EL SISTEMA DOMOTICO**



set\_umbral.vi

C:\WINDOWS\Escritorio\listo 1 de nov\set\_umbral.vi

Last modified on 02/11/05 at 07:37 a.m.

Printed on 02/11/05 at 07:38 a.m.

Connector Pane



set\_umbral.vi

Front Panel

Principal | Ajustes

Nivel de Iluminacion

255  
200  
150  
100  
50  
0  
0 100  
Time

Luz

Umbral

Umbral

0

Modo

LOCAL

Comunicacion

Error

Movimiento

Iluminacion Automática

Tab Control





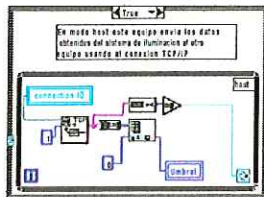
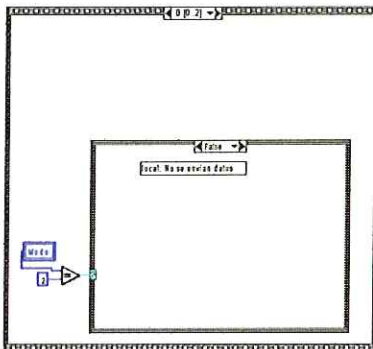
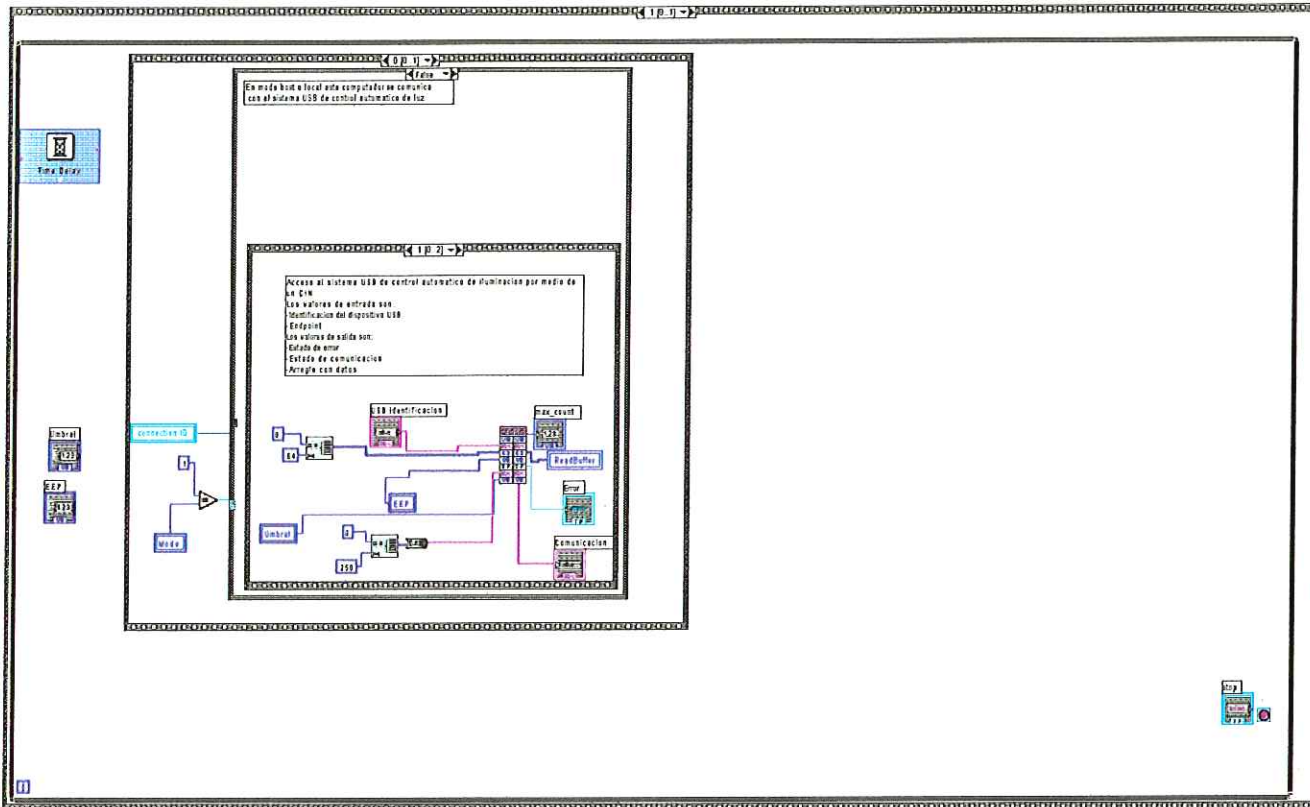
set\_umbral.vi

C:\WINDOWS\Escritorio\listo 1 de nov\set\_umbral.vi

Last modified on 02/11/05 at 07:37 a.m.

Printed on 02/11/05 at 07:38 a.m.

Block Diagram





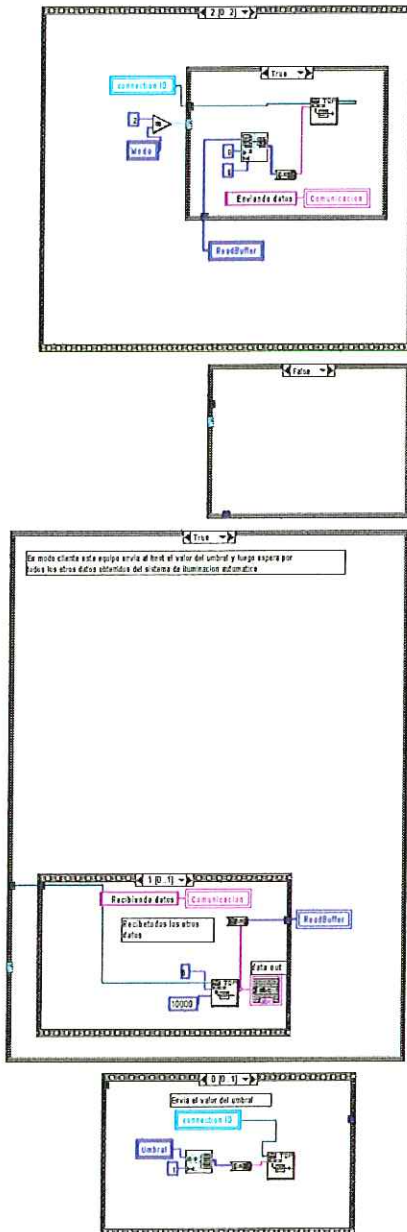


set\_umbral.vi

C:\WINDOWS\Escritorio\listo 1 de nov\set\_umbral.vi

Last modified on 02/11/05 at 07:37 a.m.

Printed on 02/11/05 at 07:38 a.m.



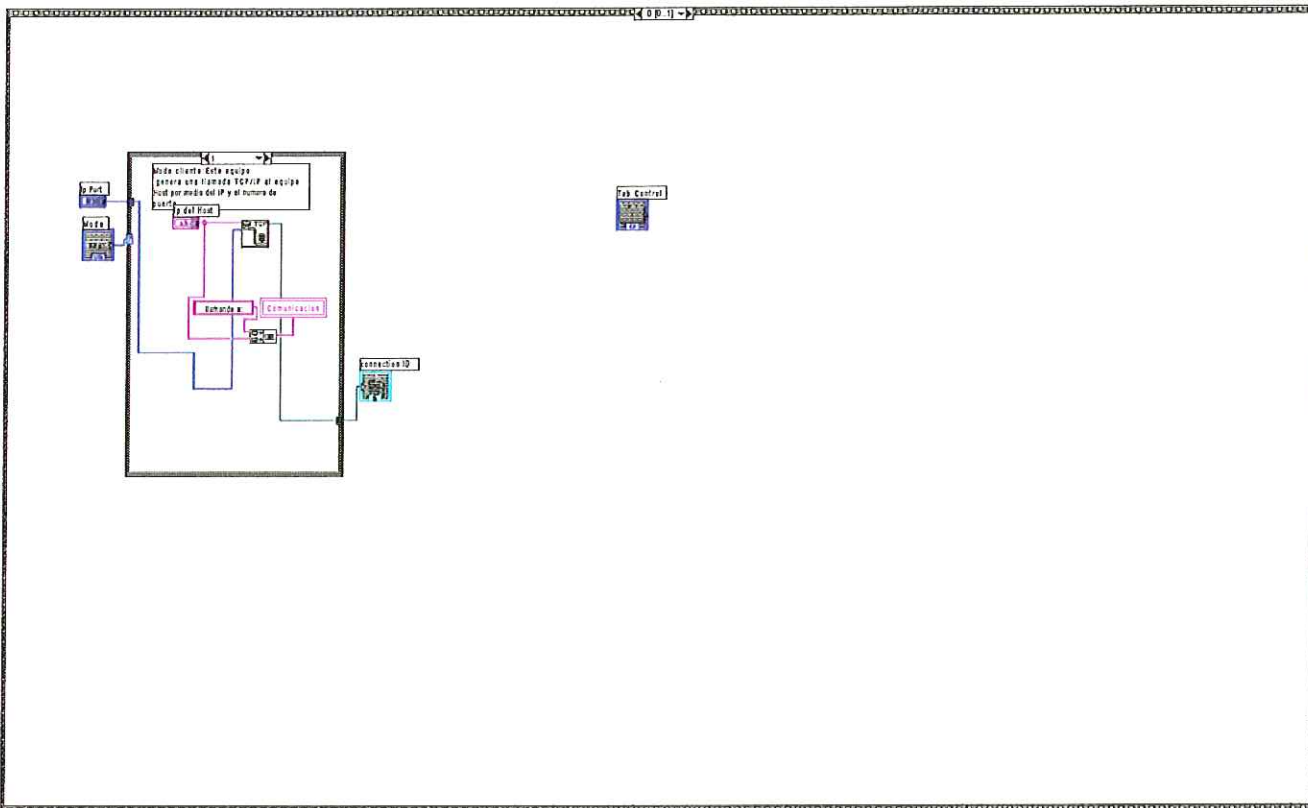
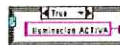
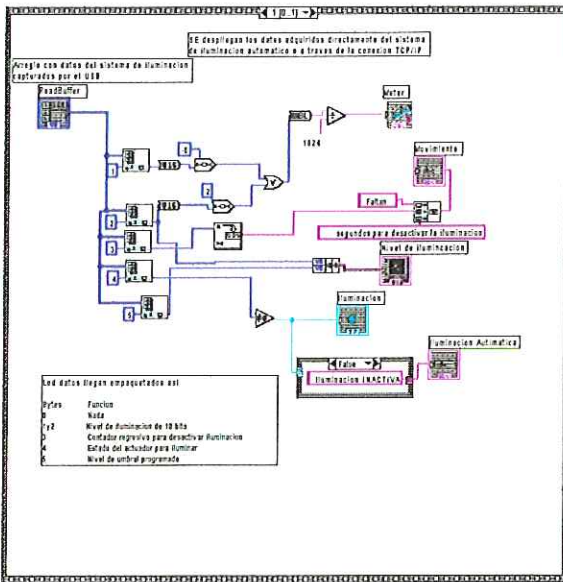


set\_umbral.vi

C:\WINDOWS\Escritorio\listo 1 de novilset\_umbral.vi

Last modified on 02/11/05 at 07:37 a.m.

Printed on 02/11/05 at 07:38 a.m.



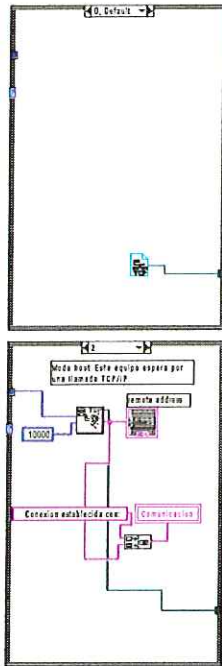


set\_umbral.vi

C:\WINDOWS\Escritorio\listo 1 de nov\set\_umbral.vi

Last modified on 02/11/05 at 07:37 a.m.

Printed on 02/11/05 at 07:38 a.m.



Express VI Configuration Information



**Time Delay**

Time Delay

Inserts a time delay in the Express VI.

-----

This Express VI is configured as follows:

Delay Time: 2 s

## ANEXO 6. PROGRAMA PARA EL MICROCONTROLADOR - CONTROL DE SEGURIDAD

```

;*****
;
;
;
; CONTROL DE SEGURIDAD
;
;
;
;
;*****
;

```

```

T_RELE      EQU    20H    ; CONTROLA EL TIEMPO QUE EL RELE
                ; ACCIONA EL MARCO DE LA PUERTA

PCL          EQU    02H
STATUS      EQU    03H
PORTA       EQU    05H
DATO        EQU    06H    ; PUERTO B
KEYB        EQU    07H
TRISA       EQU    05H
TRISB       EQU    06H
TRISC       EQU    07H
PUNTAB      EQU    20H
INTENTO     EQU    21H
TECLA       EQU    22H
CUENTA1     EQU    23H
CUENTA2     EQU    24H
CUENTA3     EQU    25H
COD1        EQU    26H
COD2        EQU    27H
COD3        EQU    28H
COD4        EQU    29H
CUENTA4     EQU    2AH
COD1A       EQU    2BH
COD2A       EQU    2CH
COD3A       EQU    2DH
COD4A       EQU    2EH
CODWRITE    EQU    2FH
EEADR       EQU    10DH
EECON1      EQU    18CH
EECON2      EQU    18DH
EEDATA      EQU    10CH
INTCON      EQU    0BH
PIR2        EQU    0DH
PIE2        EQU    8DH
F           EQU    1
W           EQU    0

#DEFINE     EN      05H,2
#DEFINE     RW      05H,1

```



```
#DEFINE RS 05H,0
#DEFINE ALA 05H,3
#DEFINE PORTERO 05H,5
```

```
ORG 00H
GOTO INICIO
ORG 04H
BCF PIR2,4
RETFIE
TABLA MOVWF PC ;SE CARGAN LAS LETRAS PARA VIZUALIZAR LCD
MEN_1 RETLW 'I' ; INTRODUCZA CLAVE
RETLW 'N'
RETLW 'T'
RETLW 'R'
RETLW 'O'
RETLW 'D'
RETLW 'U'
RETLW 'Z'
RETLW 'C'
RETLW 'A'
MEN_5 RETLW ''
RETLW 'C'
RETLW 'L'
RETLW 'A'
RETLW 'V'
RETLW 'E'
RETLW 00H
MEN_2 RETLW 'P' ; PUEDE PASAR
RETLW 'U'
RETLW 'E'
RETLW 'D'
RETLW 'E'
RETLW ''
RETLW 'P'
RETLW 'A'
RETLW 'S'
RETLW 'A'
RETLW 'R'
RETLW 00H
MEN_3 RETLW 'C' ; CLAVE INCORRECTA
RETLW 'L'
RETLW 'A'
RETLW 'V'
RETLW 'E'
RETLW ''
RETLW 'I'
```

```
RETLW      'N'  
RETLW      'C'  
RETLW      'O'  
RETLW      'R'  
RETLW      'R'  
RETLW      'E'  
RETLW      'C'  
RETLW      'T'  
RETLW      'A'  
RETLW      00H
```

```
MEN_4 RETLW      'C' ; CAMBIO DE  
RETLW      'A'  
RETLW      'M'  
RETLW      'B'  
RETLW      'T'  
RETLW      'O'  
RETLW      ''  
RETLW      'D'  
RETLW      'E'  
RETLW      00H
```

```
MEN_6 RETLW      '' ; ANTIGUA  
RETLW      'A'  
RETLW      'N'  
RETLW      'T'  
RETLW      'T'  
RETLW      'G'  
RETLW      'U'  
RETLW      'A'  
RETLW      00H
```

```
MEN_7 RETLW      'N' ; NUEVA  
RETLW      'U'  
RETLW      'E'  
RETLW      'V'  
RETLW      'A'  
RETLW      00H
```

```
MEN_8 RETLW      'V' ; VERIFIQUE  
RETLW      'E'  
RETLW      'R'  
RETLW      'T'  
RETLW      'F'  
RETLW      'T'  
RETLW      'Q'  
RETLW      'U'  
RETLW      'E'  
RETLW      00H
```

```
MEN_9 RETLW      ''      ; CAMBIADA
      RETLW      'C'
      RETLW      'A'
      RETLW      'M'
      RETLW      'B'
      RETLW      'T'
      RETLW      'A'
      RETLW      'D'
      RETLW      'A'
      RETLW      00H
```

```
MEN_10 RETLW     'L'      ; LA CLAVE DEBE TENER 4 DIGITOS
      RETLW     'A'
      RETLW     ''
      RETLW     'C'
      RETLW     'L'
      RETLW     'A'
      RETLW     'V'
      RETLW     'E'
      RETLW     ''
      RETLW     'D'
      RETLW     'E'
      RETLW     'B'
      RETLW     'E'
      RETLW     00H
```

```
MEN_11 RETLW     'T'
      RETLW     'E'
      RETLW     'N'
      RETLW     'E'
      RETLW     'R'
      RETLW     ''
      RETLW     '4'
      RETLW     ''
      RETLW     'D'
      RETLW     'T'
      RETLW     'G'
      RETLW     'T'
      RETLW     'T'
      RETLW     'O'
      RETLW     'S'
      RETLW     00H
```

```
MEN_12 RETLW     'I'      ; INTENTELO DE NUEVO
      RETLW     'N'
      RETLW     'T'
      RETLW     'E'
      RETLW     'N'
      RETLW     'T'
      RETLW     'E'
```





## ;ACUMULADOR

ECOD12 CALL	SUELKEY ;SE ESPERA A QUE SE ;SUELTE LA TECLA PULSADA ;INTRODUCE EL CÓDIGO 1 VEZ
CALL	ESPEKEY
MOVWF	COD1
SUBLW	'B'
BTFSC	STATUS,2
GOTO	ECOD12
MOVF	COD1,W
SUBLW	'A'
BTFSC	STATUS,2
GOTO	INCOR
MOVF	COD1,W
SUBLW	'C'
BTFSC	STATUS,2
GOTO	CAMBIOC
MOVF	COD1,W
SUBLW	'D'
BTFSC	STATUS,2
GOTO	ECOD12
MOVLW	!*'
CALL	WRITEL
ECOD22 CALL	SUELKEY
CALL	ESPEKEY ; INTRODUCE EL CÓDIGO 2 VEZ
MOVWF	COD2
SUBLW	'B'
BTFSC	STATUS,2
GOTO	ECOD1
MOVF	COD2,W
SUBLW	'A'
BTFSC	STATUS,2
GOTO	INCOR
MOVF	COD2,W
SUBLW	'C'
BTFSC	STATUS,2
GOTO	CAMBIOC
MOVF	COD2,W
SUBLW	'D'
BTFSC	STATUS,2
GOTO	ECOD22
MOVLW	!*'
CALL	WRITEL
ECOD32 CALL	SUELKEY
CALL	ESPEKEY ; INTRODUCE EL CÓDIGO 3 VEZ
MOVWF	COD3

SUBLW	'B'
BTFSC	STATUS,2
GOTO	ECOD2
MOVF	COD3,W
SUBLW	'A'
BTFSC	STATUS,2
GOTO	INCOR
MOVF	COD3,W
SUBLW	'C'
BTFSC	STATUS,2
GOTO	CAMBIOC
MOVF	COD3,W
SUBLW	'D'
BTFSC	STATUS,2
GOTO	ECOD32
MOVLW	'*'
CALL	WRITEL
ECOD42 CALL	SUELKEY

CALL	ESPEKEY	<b>; INTRODUCER EL C3DIGO 4 VEZ</b>
MOVWF	COD4	
SUBLW	'B'	
BTFSC	STATUS,2	
GOTO	ECOD3	
MOVF	COD4,W	
SUBLW	'A'	
BTFSC	STATUS,2	
GOTO	INCOR	
MOVF	COD4,W	
SUBLW	'C'	
BTFSC	STATUS,2	
GOTO	CAMBIOC	
MOVF	COD4,W	
SUBLW	'D'	
BTFSC	STATUS,2	
GOTO	ECOD42	
MOVLW	'*'	
CALL	WRITEL	
CALL	SUELKEY	

FINCOD CALL	ESPEKEY	<b>; FIN DE CODIGO</b>
MOVWF	TECLA	
SUBLW	'A'	
BTFSC	STATUS,2	
GOTO	FINCOD2	
MOVF	TECLA,W	
SUBLW	'B'	
BTFSC	STATUS,2	
GOTO	ECOD4	

```
MOVF    TECLA,W
SUBLW   'C'
BTFSC   STATUS,2
GOTO    CAMBIOC
GOTO    FINCOD
```

```
FINCOD2MOVLW 00H
CALL    LEECOD
SUBWF   COD1
BTFSS   STATUS,2
GOTO    INCOR
```

```
MOVLW   01H
CALL    LEECOD
SUBWF   COD2
BTFSS   STATUS,2
GOTO    INCOR
```

```
MOVLW   02H
CALL    LEECOD
SUBWF   COD3
BTFSS   STATUS,2
GOTO    INCOR
```

```
MOVLW   03H
CALL    LEECOD
SUBWF   COD4
BTFSS   STATUS,2
GOTO    INCOR
```

```
CALL    CLEARL                ;CLAVE CORRECTA
MOVLW   02H
CALL    SITUCU
MOVLW   MEN_2
CALL    ESCMEN
MOVLW   03H
MOVWF   INTENTO
BSF     PORTERO
MOVLW   T_RELE
CALL    DELAY
BCF     PORTERO
MOVLW   10H
CALL    DELAY
GOTO    PRINCI
```

```
INCOR MOVLW 0CH                ; CLAVE INCORRECTA
CALL  COMANDO
CALL  CLEARL
MOVLW MEN_3
```

	CALL	ESCMEN
	DECFSZ	INTENTO,F
	GOTO	INCOR2
	GOTO	ALARMA
INCOR2	MOVLW	20H
	CALL	DELAY
	GOTO	PRINCI
ECOD1	MOVLW	44H
	CALL	SITUCU
	MOVLW	' '
	CALL	WRITEL
	MOVLW	44H
	CALL	SITUCU
	GOTO	ECOD12
ECOD2	MOVLW	45H
	CALL	SITUCU
	MOVLW	' '
	CALL	WRITEL
	MOVLW	45H
	CALL	SITUCU
	GOTO	ECOD22
ECOD3	MOVLW	46H
	CALL	SITUCU
	MOVLW	' '
	CALL	WRITEL
	MOVLW	46H
	CALL	SITUCU
	GOTO	ECOD32
ECOD4	MOVLW	47H
	CALL	SITUCU
	MOVLW	' '
	CALL	WRITEL
	MOVLW	47H
	CALL	SITUCU
	GOTO	ECOD42
ALARMA	MOVLW	1EH ; ALARMAS PARA CODIGO INCORRECTO
	MOVWF	TECLA
ALARMA4	MOVLW	03H
	MOVWF	INTENTO
	MOVLW	00FH
	MOVWF	CUENTA1
	MOVLW	030H
	MOVWF	CUENTA2
ALARMA1	MOVLW	20H
	MOVWF	CUENTA4



DECF	CUENTA1,F
BTFSC	STATUS,2
GOTO	ALARMA2
ALARMA3 BSF	ALA
CALL	FRECU
BCF	ALA
CALL	FRECU
DECFSZ	CUENTA4,F
GOTO	ALARMA3
GOTO	ALARMA1

ALARMA2MOVLW	01H
CALL	DELAY
DECFSZ	TECLA,F
GOTO	ALARMA4
GOTO	PRINCI

FRECU MOVF	CUENTA1,W
MOVWF	CUENTA3
FRECU2 DECFSZ	CUENTA3,F
GOTO	FRECU2
DECFSZ	CUENTA2,F
GOTO	FRECU
MOVLW	030H
MOVWF	CUENTA2
RETURN	

CAMBIOCALL CLEARL

MOVLW	MEN_4
CALL	ESCMEN
MOVLW	MEN_5
CALL	ESCMEN
MOVLW	20H
CALL	DELAY
CALL	CLEARL
MOVLW	MEN_5
CALL	ESCMEN
MOVLW	MEN_6
CALL	ESCMEN
CALL	SUELKEY
MOVLW	44H
CALL	SITUCU

CALL	COGECOD
MOVLW	00H
CALL	LEECOD
SUBWF	COD1
BTFSS	STATUS,2
GOTO	INCORE

; CAMBIO DE CLAVE

MOVLW 01H  
CALL LEECOD  
SUBWF COD2  
BTFSS STATUS,2  
GOTO INCORE

MOVLW 02H  
CALLV LEECOD  
SUBWF COD3  
BTFSS STATUS,2  
GOTO INCORE

MOVLW 03H  
CALL LEECOD  
SUBWF COD4  
BTFSS STATUS,2  
GOTO INCORE

MOVLW 03H  
MOVWF INTENTO

NUEVACL CALL CLEARL ; NUEVA CLAVE  
MOVLW 02H  
CALL SITUCU  
MOVLW MEN\_7  
CALL ESCMEN  
MOVLW MEN\_5  
CALL ESCMEN  
MOVLW 44H  
CALL SITUCU  
CALL SUELKEY  
CALL COGECOD  
MOVF COD1,W  
BTFSC STATUS,2  
GOTO INCOREC  
MOVWF COD1A  
MOVF COD2,W  
BTFSC STATUS,2  
GOTO INCOREC  
MOVWF COD2A  
MOVF COD3,W  
BTFSC STATUS,2  
GOTO INCOREC  
MOVWF COD3A  
MOVF COD4,W  
BTFSC STATUS,2  
GOTO INCOREC  
MOVWF COD4A

CALL CLEARL ;VERIFIQUE LA NUEVA CLAVE

```

MOVLW    MEN_8
CALL     ESCMEN
MOVLW    MEN_5
CALL     ESCMEN
MOVLW    44H
CALL     SITUCU
CALL     SUELKEY
CALL     COGECOD
MOVF     COD1,W
SUBWF    COD1A,W
BTFSS    STATUS,2
GOTO     NOVERI
MOVF     COD2,W
SUBWF    COD2A,W
BTFSS    STATUS,2
GOTO     NOVERI
MOVF     COD3,W
SUBWF    COD3A,W
BTFSS    STATUS,2
GOTO     NOVERI
MOVF     COD4,W
SUBWF    COD4A,W
BTFSS    STATUS,2
GOTO     NOVERI

```

```

CALL     CLEARL           ;LA CLAVE SE HA CAMBIADO
MOVLW    MEN_5
CALL     ESCMEN
MOVLW    MEN_9
CALL     ESCMEN

```

```

MOVF     COD1,W           ; ESCRIBE CODIGO 1
MOVWF    CODWRITE
MOVLW    00H
CALL     EEWRITE

```

```

MOVF     COD2,W           ; ESCRIBE CODIGO 2
MOVWF    CODWRITE
MOVLW    01H
CALL     EEWRITE

```

```

MOVF     COD3,W           ; ESCRIBE CODIGO 3
MOVWF    CODWRITE
MOVLW    02H
CALL     EEWRITE

```

```

MOVF     COD4,W           ; ESCRIBE CODIGO 4
MOVWF    CODWRITE
MOVLW    03H
CALL     EEWRITE

```

MOVLW 20H  
CALL DELAY

GOTO PRINCI

NOVERICALL CLEARL  
MOVLW 03H  
CALL SITUCU  
MOVLW MEN\_12  
CALL ESCMEN  
MOVLW 43H  
CALL SITUCU  
MOVLW MEN\_13  
CALL ESCMEN  
MOVLW 20H  
CALL DELAY  
GOTO NUEVACL

INCORECCALL CLEARL ;LA CLAVE DEBE TENER 4 DIGITOS  
MOVLW MEN\_10  
CALL ESCMEN  
MOVLW 40H  
CALL SITUCU  
MOVLW MEN\_11  
CALL ESCMEN  
MOVLW 20H  
CALL DELAY  
GOTO PRINCI

INCOREDEC F  
BTFS STATUS,2  
GOTO ALARMA  
CALL CLEARL  
MOVLW MEN\_3  
CALL ESCMEN  
MOVLW 20H  
CALL DELAY  
CALL SUELKEY  
GOTO PRINCI

-----  
COGECOD CLR F COD1 ;RECIBE HASTA 4 CODOIGOS INVALIDOS  
CLR F COD2  
CLR F COD3  
CLR F COD4  
ECOD12C CALL SUELKEY ; PRIMER CODIGO  
CALL ESPEKEY  
MOVWF COD1



```

SUBLW      'B'
BTFSC     STATUS,2
GOTO      ECOD12C
MOVF      COD1,W
SUBLW 'A'
BTFSC     STATUS,2
RETURN
MOVF      COD1,W
SUBLW     'C'
BTFSC     STATUS,2
GOTO      ECOD12C
MOVF      COD1,W
SUBLW     'D'
BTFSC     STATUS,2
GOTO      ECOD12C
MOVLW    '*'
CALL      WRITEL

```

```

ECOD22C CALL      SUELKEY      ; SEGUNDO CODIGO
CALL      ESPEKEY
MOVWF     COD2
SUBLW     'B'
BTFSC     STATUS,2
GOTO      ECOD1C
MOVF      COD2,W
SUBLW     'A'
BTFSC     STATUS,2
RETURN
MOVF      COD2,W
SUBLW     'C'
BTFSC     STATUS,2
GOTO      ECOD22C
MOVF      COD2,W
SUBLW     'D'
BTFSC     STATUS,2
GOTO      ECOD22C
MOVLW    '*'
CALL      WRITEL

```

```

ECOD32C CALL      SUELKEY      ; TERCER CODIGO
CALL      ESPEKEY
MOVWF     COD3
SUBLW     'B'
BTFSC     STATUS,2
GOTO      ECOD2C
MOVF      COD3,W
SUBLW     'A'
BTFSC     STATUS,2
RETURN
MOVF      COD3,W

```

SUBLW	'C'
BTFSC	STATUS,2
GOTO	ECOD32C
MOVF	COD3,W
SUBLW	'D'
BTFSC	STATUS,2
GOTO	ECOD32C
MOVLW	'*'
CALL	WRITEL

ECOD42C CALL	SUELKEY	<b>; CUARTO CODIGO</b>
CALL	ESPEKEY	
MOVWF	COD4	
SUBLW	'B'	
BTFSC	STATUS,2	
GOTO	ECOD3C	
MOVF	COD4,W	
SUBLW	'A'	
BTFSC	STATUS,2	
GOTO	RETU4	
MOVF	COD4,W	
SUBLW	'C'	
BTFSC	STATUS,2	
GOTO	ECOD42C	
MOVF	COD4,W	
SUBLW	'D'	
BTFSC	STATUS,2	
GOTO	ECOD42C	
MOVLW	'*'	
CALL	WRITEL	
CALL	SUELKEY	

FINCODC CALL	ESPEKEY	
MOVWF	TECLA	
SUBLW	'A'	
BTFSC	STATUS,2	
RETURN		<b>;YA TENGO EL CÓDIGO</b>
MOVF	TECLA,W	
SUBLW	'B'	
BTFSC	STATUS,2	
GOTO	ECOD4C	
GOTO	FINCODC	

ECOD1C MOVLW	44H	
CALL	SITUCU	<b>SUBROUTINA QUE SITUA EL CURSOR EN LA DIRECCION INVADIDA POR EL ACUMULADOR</b>

```

        MOVLW      ''
        CALL      WRITEL ;SUBROUTINA QUE ESCRIBE EN EL LCD
                                ; EL CARÁCTER DEL ACUMULADOR

        MOVLW      44H
        CALL      SITUCU
        CLRF      COD2
        GOTO      ECOD12C

ECOD2C MOVLW      45H
        CALL      SITUCU
        MOVLW      ''
        CALL      WRITEL
        MOVLW      45H
        CALL      SITUCU
        CLRF      COD3
        GOTO      ECOD22C

ECOD3C MOVLW      46H
        CALL      SITUCU
        MOVLW      ''
        CALL      WRITEL
        MOVLW      46H
        CALL      SITUCU
        CLRF      COD4
        GOTO      ECOD32C

ECOD4C MOVLW      47H
        CALL      SITUCU
        MOVLW      ''
        CALL      WRITEL
        MOVLW      47H
        CALL      SITUCU
        GOTO      ECOD42C

RETU4  CLRF      COD4
        RETURN

```

---

```

;-----
;      LEECOD:    SUBROUTINA QUE LEE EL REGISTRO DE LA
;                  EEPROM INDICADO POR EL ACUMULADOR
;                  Y LO DEVUELVE EN ACUMULADOR
;

```

```

LEECOD BSF      STATUS,6
        BCF      STATUS,5
        MOVWF   EEADR
        BSF      STATUS,5
        BCF      EECON1,7
        BSF      EECON1,0
        BCF      STATUS,5

```

```
MOVF      EEDATA,W
BCF       STATUS,6
RETURN
```

-----  
; **EEWRITE: SUBROUTINA QUE ESCRIBE EN LA EEPRON DE DATOS  
PASAR LA DIRECCION DE LA EEPRON A ESCRIBIR  
EN W Y EL DATO A ESCRIBIR EN CODWRITE**

```
EEWRITEBSF STATUS,6
BCF        STATUS,5
MOVWF     EEADR
BCF       STATUS,6
MOVF     CODWRITE,W
BSF      STATUS,6
MOVWF   EEDATA
BSF     STATUS,5
BCF    EECON1,7
BSF    EECON1,2
BCF    INTCON,7
MOVLW 55H
MOVWF EECON2
MOVLW 0AAH
MOVWF EECON2
BSF   EECON1,1
BSF   INTCON,7
BCF   STATUS,5
BCF   STATUS,6
EEWRITBTFSS PIR2,4
GOTO  EEWRIT
BSF   STATUS,5
BSF   STATUS,6
BCF   EECON1,2
BCF   STATUS,5
BCF   STATUS,6
BCF   PIR2,4

RETURN
```

-----  
; **GETKEY: SUBROUTINA QUE DEVUELVE EL VALOR PULSADO EN  
UN TECLADO HEXADECIMAL EN CÓDIGO ASCII EN W**  
;COL1: 0 COL2: 1 COL3: 2 COL4: 3  
;ROW1: 4 ROW2: 5 ROW3: 6 ROW4: 7

```
GETKEY MOVLW 0FH
```



	MOVWF	KEYB
	NOP	
	MOVF	KEYB,W
	MOVWF	TECLA
	CALL	REBOTE
	MOVF	KEYB,W
	SUBWF	TECLA,F
	BTFSS	STATUS,2
	GOTO	GETKEY
	SUBLW	0FH
	BTFSS	STATUS,2
	GOTO	HAYTECL
	RETLW	0FH
HAYTECL	BTFSC	KEYB,4
	GOTO	ROW1
	BTFSC	KEYB,5
	GOTO	ROW2
	BTFSC	KEYB,6
	GOTO	ROW3
	BTFSC	KEYB,7
	GOTO	ROW4
	RETLW	0FH
ROW1	MOVLW	01H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,4
	RETLW	31H ;TECLA=1
	MOVLW	02H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,4
	RETLW	32H ;TECLA=2
	MOVLW	04H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,4
	RETLW	33H ;TECLA=3
	MOVLW	08H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,4
	RETLW	'A' ;TECLA=A
	RETLW	0FH
ROW2	MOVLW	01H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,5
	RETLW	34H ;TECLA=4
	MOVLW	02H

	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,5
	RETLW	35H ;TECLA=5
	MOVLW	04H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,5
	RETLW	36H ;TECLA=6
	MOVLW	08H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,5
	RETLW	'B' ;TECLA=B
	RETLW	0FH
ROW3	MOVLW	01H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,6
	RETLW	37H ;TECLA=7
	MOVLW	02H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,6
	RETLW	38H ;TECLA=8
	MOVLW	04H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,6
	RETLW	39H ;TECLA=9
	MOVLW	08H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,6
	RETLW	'C' ;TECLA=C
	RETLW	0FH
ROW4	MOVLW	01H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,7
	RETLW	'*' ;TECLA=*
	MOVLW	02H
	MOVWF	KEYB
	NOP	
	BTFSC	KEYB,7
	RETLW	30H ;TECLA=0
	MOVLW	04H
	MOVWF	KEYB

```

NOP
BTFSC      KEYB,7
RETLW     #'      ;TECLA=#
MOVLW     08H
MOVWF     KEYB
NOP
BTFSC      KEYB,7
RETLW     'D'      ;TECLA=D
RETLW     0FH

```

```

-----
;
;
;   SUELKEY:      SUBROUTINA QUE ESPERA A QUE SE SUELTE
;                 LA TECLA PULSADA
;

```

```

SUELKEY CALL  GETKEY
SUBLW        0FH
BTFSC        STATUS,2
RETURN
GOTO         SUELKEY

```

```

-----
;
;
;   ESPEKEY:     SUBROUTINA QUE ESPERA A QUE SE PULSE UNA TECLA
;                 DEVUELVE EL VALOR EN W Y TECLA
;

```

```

ESPEKEY CALL  GETKEY
MOVWF        TECLA
SUBLW        0FH
BTFSC        STATUS,2
GOTO         ESPEKEY
MOVF         TECLA,W
RETURN

```

```

-----
;
;
;   ESCMEN:     SUBROUTINA QUE ESCRIBE UN MENSAJE EN EL LCD,
;                 PASAR INICIO DE TABLA EN ACUMULADOR
;

```

```

ESCMEN      MOVWF      PUNTAB
ESCMEN2     CALL       TABLA
            ADDLW      00H
            BTFSC      STATUS,2
            RETURN
            CALL       WRITEL
            INCF       PUNTAB,F
            MOVF       PUNTAB,W
            GOTO       ESCMEN2

```

-----

;  
;     **WAITLCD:            SUBROUTINA DE ESPERA AL LCD**

```
WAITLCD   BSF            STATUS,5
          MOVLW         0FFH
          MOVWF         DATO
          BCF            STATUS,5
          BSF            EN
          BCF            RS
          BSF            RW
          MOVLW         0FFH
          MOVWF         DATO
WAITLC    BTFSC         DATO,7
          GOTO          WAITLC
          BCF            EN
          BCF            RW
          BCF            DATO,7
          BSF            STATUS,5
          CLRF          DATO
          BCF            STATUS,5
          RETURN
```

-----

-----

;  
;     **INITLCD:            SUBROUTINA DE INICIALIZACIÓN DEL LCD**

```
INITLCD   MOVLW     38H            ;00111000
          CALL      COMANDO       ;INTERFACE 8 BIT, 2 LINEAS
          MOVLW     0CH            ;00001100
          CALL      COMANDO       ;DISPLAY A ON, NO MUESTRA
                                  ;CURSOR NO PARPADEO
          MOVLW     06H            ;00000110
          CALL      COMANDO       ;MUEVE CURSOR A LA DCHA
                                  ;DESPUES DE ESCRIBIR, NO MUEVE
                                  ;LA VENTANA
          RETURN
```

-----

-----

;  
;     **CLEARL:            SUBROUTINA QUE BORRA EL LCD Y LLEVA EL**  
;  
;                        **CURSOR AL LADO IZQUIERDO DEL DISPLAY**

```
CLEARL    MOVLW         01H
          CALL          COMANDO            ;BORRADO
          RETURN
```





DECFSZ CUENTA1,F  
GOTO DELAY3  
RETURN

REBOTE

MOVLW  
MOVWF  
DECFSZ  
GOTO  
RETURN

0FFH  
CUENTA1  
CUENTA1,F  
REBO

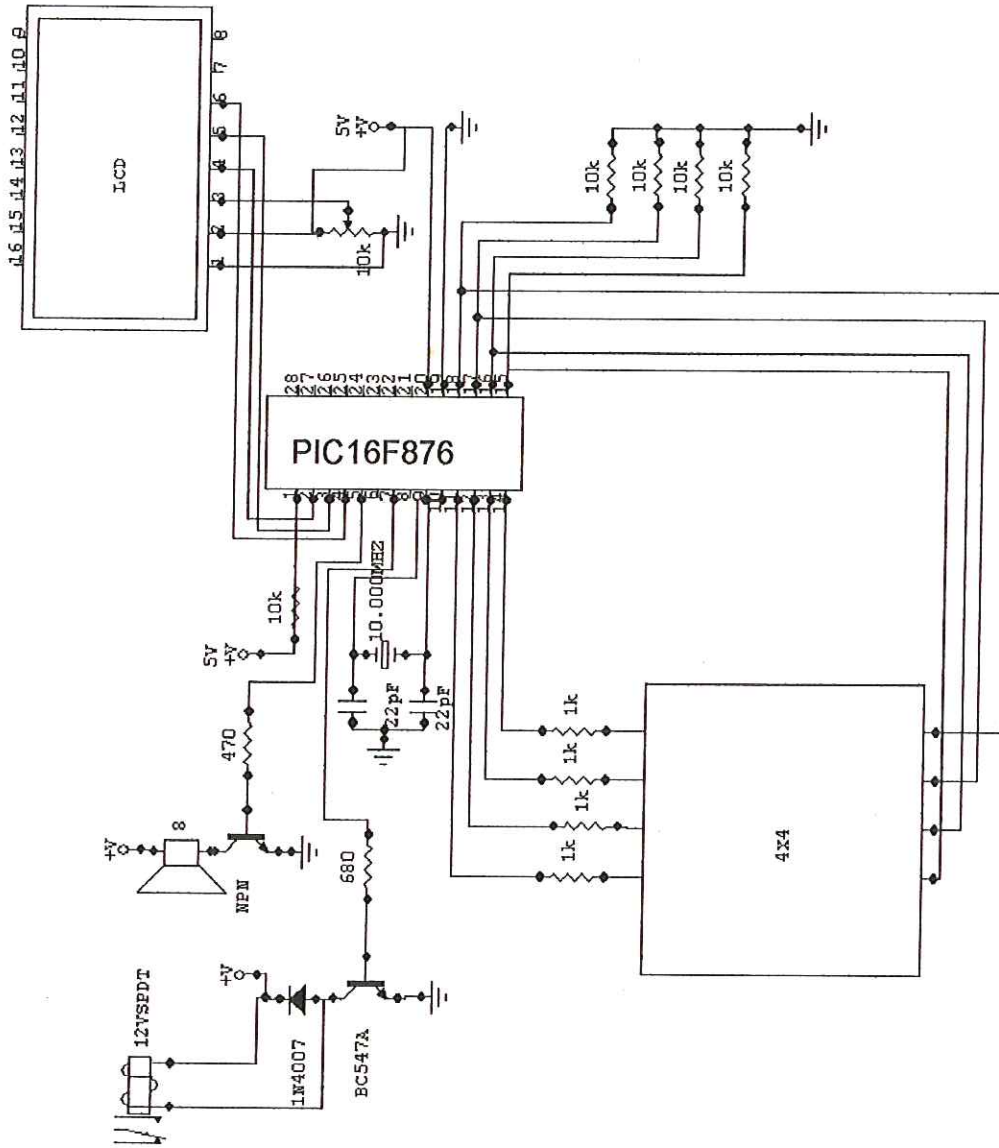
REBO

ORG  
DE

2100H  
30H,30H,30H,30H

END

# ANEXO 7. CIRCUITO ELECTRONICO CONTROL DE INGRESO



**ANEXO 8. FOLLETOS, CATALOGOS, ESPECIFICACIONES TECNICAS**



### DS940 Detector de Intrusión

Código de Modelo	Código de Precio	Descripción
DS940	4 998 800 704	Cobertura: 12m x 12m
DS940T	4 998 800 705	Cobertura: 12m x 12m, antidesarme



El DS940T es un detector por infrarrojo pasivo para montaje sobre pared que utiliza lentes de Fresnel facetadas para lograr una capacidad de detección claramente superior, a la facilidad de instalación y la flexibilidad de sus opciones de montaje proveen detección de avanzada en todo momento.

### Características

- Facilidad de instalación con opciones de montajes muy flexibles.
- Diseño pequeño y discreto.
- Cobertura volumétrica de 12 mts.
- Opción de Montaje con Soporte.
- Inmunidad contra Corrientes de Aire e Insectos.
- Antidesarme incorporado.
- Listado UL, CE.

### ■ Funcionamiento Básico

#### Procesamiento de Señales

El Procesamiento de Inicialización Patentado First Step (FSP) provee un ajuste automático durante la puesta en marcha, lográndose así una respuesta instantánea ante cualquier presencia humana sin sacrificar la inmunidad contra falsas alarmas. Su sensibilidad está basada en el ajuste de la amplitud, polaridad, pendiente y período de la señal, por lo que el FSP le elimina al instalador la necesidad de ajustar un nivel de sensibilidad en cada aplicación. Esta nueva tecnología garantiza el óptimo funcionamiento de captura a cualquier altura de instalación entre 2,2 y 2,7 metros.

#### Características de Prueba

LED de alarma de alta luminosidad que puede deshabilitarse después de la instalación.

#### Inmunidad contra Corrientes de Aire e Insectos

Sus ópticas selladas proveen inmunidad contra corrientes de aire e insectos.

### ■ Certificaciones

Listado UL	BP1448
CE-EMC	89/336

### ■ Patentes

4,764,755	BP1448
-----------	--------

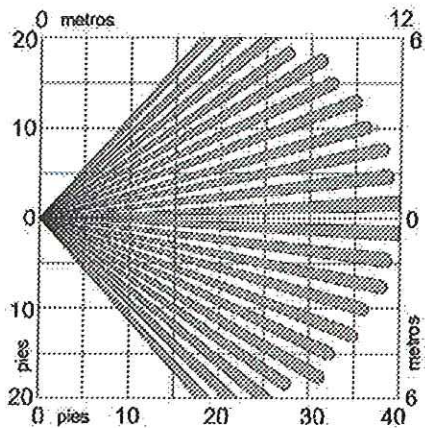
### ■ Cobertura

#### Cobertura Estándar

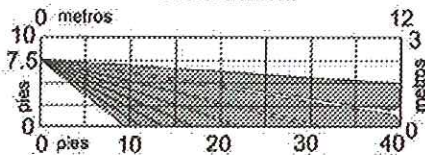
Volumétrico: 12 m x 12 m.



cobertura volumétrica - estándar  
vista superior



vista lateral



## ■ Especificaciones Técnicas

<b>Alimentación</b>	9 a 15 VCC, 20 mA @ 12 VCC.
<b>Salida de Alarma</b>	Contacto de relé Normal Cerrado. Capacidad de 18VAC de carga resistiva con 125 mA @ 28 VCC. El relé se abre ante una condición de alarma.
<b>Salida de Antidasarma</b>	Normal cerrado (con la cubierta colocada). Contactos preparados para un máximo de 125 mA @ 28 VCC. El interruptor se abre cuando se retira la cubierta.

**Interferencias de Radio (RFI)**

No se producen alarmas ni alteraciones en el rango de frecuencias críticas de 26 a 950 MHz a 30 v/m.

**Almacenamiento y Temperatura de Operación**

De -30°C a 49°C.  
Para instalaciones certificadas UL:  
De 0°C a 49°C.

**Diseño de Gabinete**

Gabinete de plástico ABS de alto impacto con marco interno para lentes. Medidas: (Al x An x F) 8,5 cm x 6,9 cm x 4,8 cm (3,35 pulg x 2,75 pulg x 1,9 pulg)

**Montaje Estándar**

Montaje superficial o en esquina. La altura de montaje recomendada es de 2,25 a 2,7 metros.

**Planos de Detección**

El diagrama de protección volumétrica consiste de 7 planos horizontales de 11 pares de zonas infrarrojas cada uno. Cada uno de estos pares está separado desde el centro a 0°, ±9°, ±18°, ±27°, ±36°, ±45° respectivamente, en el plano horizontal. La separación vertical entre planos y el alcance máximo de cada uno será:

**Plano 1:** Nivelado a -6,1° del plano horizontal y alcanza los 12,2 mts (40 pies).

**Plano 2:** Nivelado a -11,2° del plano horizontal y alcanza los 11,3 mts (37 pies).

**Plano 3:** Nivelado a -16,3° del plano horizontal y alcanza los 6,7 mts (22 pies).

**Plano 4:** Nivelado a -21,4° del plano horizontal y alcanza los 4,6 mts (15 pies).

**Plano 5:** Nivelado a -26,5° del plano horizontal y alcanza los 3,7 mts (12 pies).

**Plano 6:** Nivelado a -31,6° del plano horizontal y alcanza los 3 mts (10 pies).

**Plano 7:** Nivelado a -36,6° del plano horizontal y alcanza los 2,1 mts (7 pies).

## ■ Accesorios opcionales

### Soportes

Código de Modelo	Código de Pedido	Descripción
B335-3	4 998 800 014	Soporte de baja perfil



# DS940/DS940T

## Passive Infrared Detector

### Installation Instructions

#### 1.0 Description

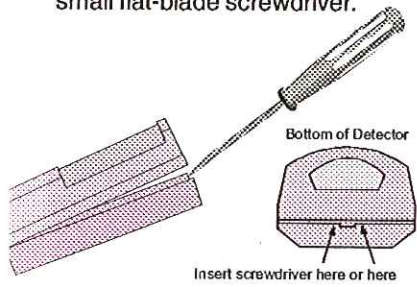
The DS940 is a high performance Passive Infrared (PIR) Motion Sensor which uses advanced signal processing to provide outstanding catch performance and unsurpassed false alarm immunity. It is designed to detect movement in the interior of a structure by sensing the Infrared energy emitted from the human body as it moves across the sensor's field of view. When motion is detected the unit sends an alarm signal to the Control Panel.

#### 2.0 Specifications

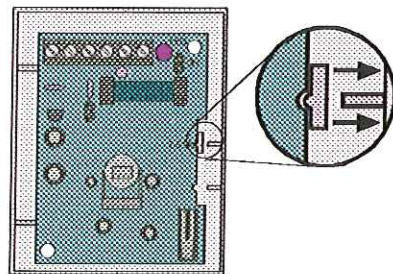
<p><b>Input Power:</b> 9-15 volts DC</p> <p><b>Current Draw:</b> 17 mA @ 12 VDC (standby and maximum)</p> <p><b>Standby Power:</b> No internal standby battery. <i>For UL Listed Product Installations, 4 hours (68 mAh) standby power must be provided by the control unit or by a Listed burglary power supply.</i></p> <p><b>Relay</b> Form "A" Normally Closed (NC) contact set rated for 125 mA @ 28 volts maximum DC, 18 volts maximum AC for resistive loads.</p> <p><b>Tamper:</b> (DS940T only) Normally Closed (with cover on). Contacts rated at 28 VDC, 125 mA max. Connect tamper circuit to a 24hr. protection circuit.</p>	<ul style="list-style-type: none"> <li>• <b>Temperature:</b> -20° to +120°F (-29° to +49°C). <i>For UL Listed installations, the temperature range is +32°F to +120°F (0°C to +49°C).</i></li> <li>• <b>Humidity:</b> 0 - 85% non-condensing.</li> <li>• <b>Dimensions:</b> 3 in. x 2.25 in. x 1.5 in. (HxWxD) (7.6 cm x 5.7 cm x 3.8 cm)</li> <li>• <b>Options:</b> B335 Swivel Mounting Bracket. Use of this bracket may decrease the PIR range and increase dead zones.</li> <li>• <b>Patents:</b> This product is covered by one or more of the following U.S. patents: #4764755, #5670943.</li> </ul>
---	--

#### 3.0 Installation

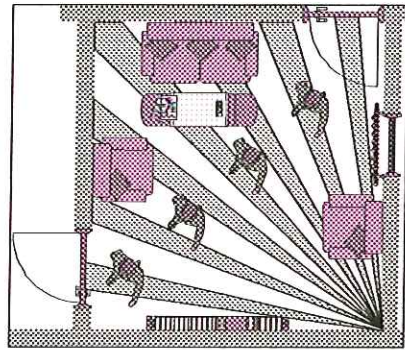
**1 Remove the cover** using a small flat-blade screwdriver.



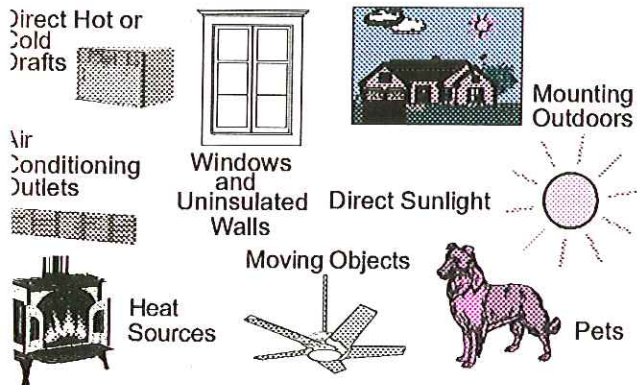
**3.2 Press the vertical adjust tab** toward the side of the case and lift out the board.



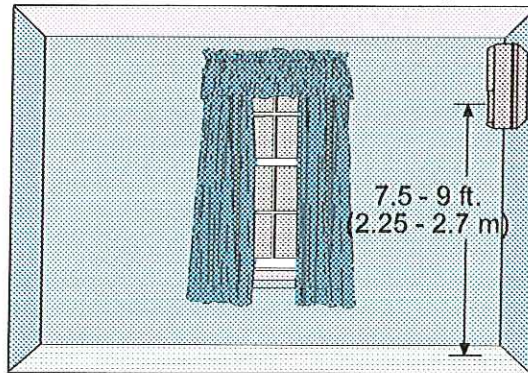
**3.3 Select a mounting location.** Mount the sensor where an intruder will most likely cross through the coverage pattern.



#### Avoid.



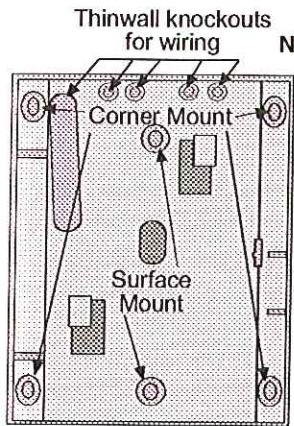
**3.5 Position the detector at the correct height.**



Mount the detector 7.5 - 9 feet (2.25 - 2.7 meters) above the floor



**3.6 Mount the unit,** using the appropriate mounting holes.

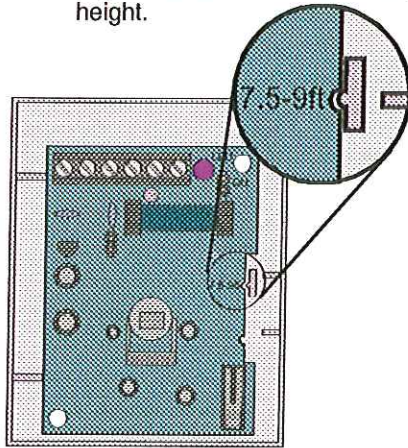


**Note:** To avoid possible circuit board damage, use **only** the mounting hardware provided in the appropriate punch-out mounting holes. Remove

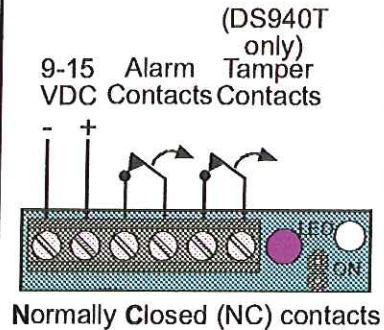
Areas if using the B335 Bracket

**Don't overtighten the mounting screws!**  
Cover may not attach correctly

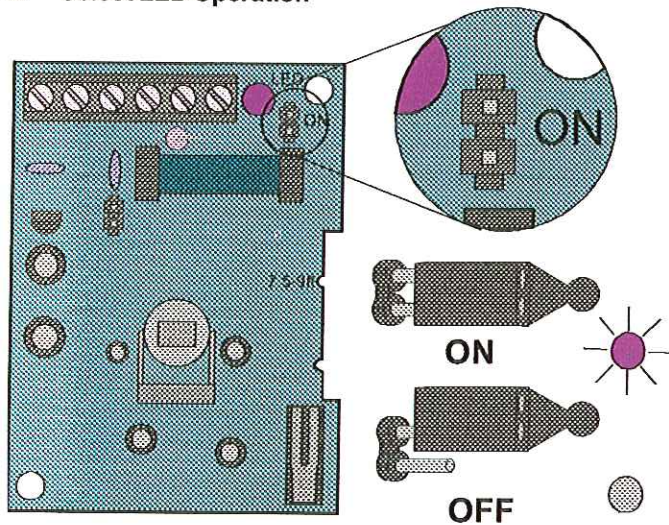
**3.7 Install the board into the case.** Be sure the vertical adjust tab aligns with the correct mounting height.



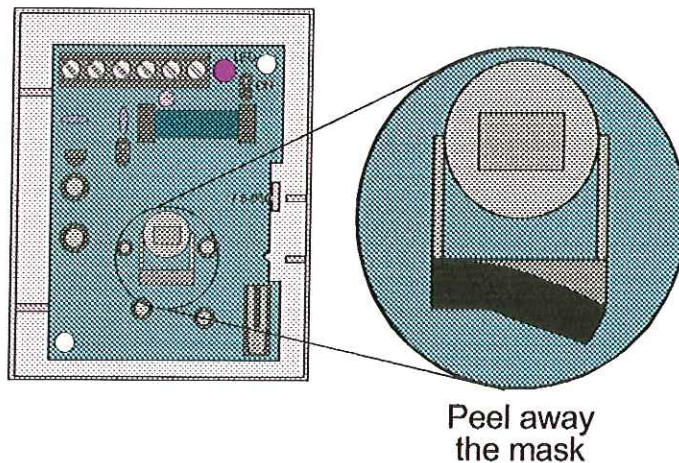
**3.8 Wire the Detector.**



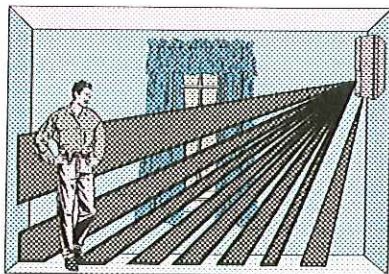
**3.9 Select LED Operation**



**3.10 If look-down is desired, peel away the look-down mask. Do not remove the clear plastic lens.**



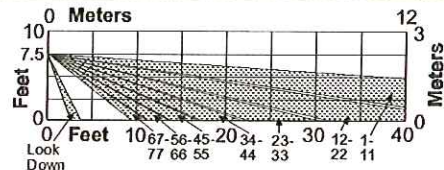
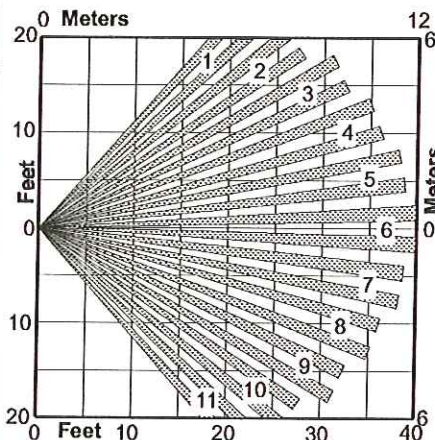
**4.0 Walk test the detector.** This test should be performed at the time of installation and annually thereafter.



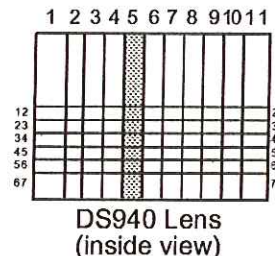
This detector contains an environmental stabilization circuit which requires approximately 2 minutes after initial power-up to warm up. During this time the detector will not respond to any movement.

**Please wait 2 minutes after initial power-up to perform any walk tests.**

**5.0 Coverage Patterns**



Although generally not required, if masking is desired, the lens diagram shows the appropriate areas to be masked (see shaded area on the DS940 lens diagram). Use an opaque material (such as, electrical tape) to mask the desired areas.







## TBW-101UB Adaptador USB Bluetooth

El Adaptador USB Bluetooth TBW-101UB de TRENDnet permite hacer conexiones inalámbricas de corto alcance entre su computadora y dispositivos permitidos Bluetooth tales como teléfonos celulares, PDAs y computadoras. Es compacto y portátil, eliminando así la necesidad de usar cables y conexiones físicas entre dispositivos electrónicos. La transmisión del Adaptador USB Bluetooth asegura protección contra de interferencia y transferencia de datos segura. Simplemente enchúfe el Adaptador USB al puerto USB y goce de la libertad y flexibilidad de la tecnología inalámbrica Bluetooth.



### Características

- Sumiso con la Especificación USB 1.1
- Usa la frecuencia de banda de 2.4Ghz ISM con la frecuencia estándar Hopping Spread Spectrum (FHSS)
- Transmisión Inalámbrica de alta velocidad, hasta 723Kbps de procesamiento de datos
- Compatible con Windows 98/ME/2000/XP
- Permite a cualquier computadora con conexiones USB convertirse en una computadora permitida Bluetooth
- Permite que computadoras se conecten a impresoras Bluetooth y dispositivos móviles que permitan la función Bluetooth (PDAs y teléfonos móviles)
- Seguridad del cifrado 128-bit integrado para comunicación entre los dispositivos Bluetooth
- Sincroniza automáticamente dispositivos en la red de area personal "Personal Area Network" (PAN) para mantener datos constantes
- Plug & Play e instalación sencilla
- Rango de conectividad de hasta 10 metros (33 pies)
- 3 año de garantía

### Ventajas

- **Integración Sencilla:**  
Drivers support Windows Operating Systems
- **Económico:**  
Save time, labor, and material for network cable installation
- **Flexible:**  
Connects PC, Laptop, Cellular Phone and PDA to share data, files and information in Personal Area Network quickly and easily
- **Funcionamiento:**  
Data Transfer Rate up to 723 Kbps





# TBW-101UB

## adaptador USB Bluetooth

### Especificaciones Técnicas

**Adapter Standards:**

- USB 1.1, Bluetooth 1.1

**Frequency:**

- 2.4 - 2.4835 GHz

**Module Technique:**

- Frequency Hopping Spread Spectrum (FHSS)

**Antenna:**

- 2dBi (Build-in)

**Receiving Sensitivity:**

- -80dBm (typical)

**Data Rate:**

- Up to 723 Kbps

**Data Range:**

- Up to 10m (33 ft.)

**Topology:**

- Point to MultiPoint (up to 8 devices totally)

**Unterstützte Betriebssysteme:**

- Windows 98/ME/2000/XP mit USB-Unterstützung

**LED-Anzeige:**

- Stromversorgung

**Betriebsspannung:**

- 5 V Gleichstrom

**Stromverbrauch:**

- 33 mA max. im Übertragungsmodus
- 38 mA max. im Empfängermodus
- 259 uA max. im Schlafmodus

**Abmessungen (BxTxH):**

- 70 x 25 x 7 mm

**Gewicht:**

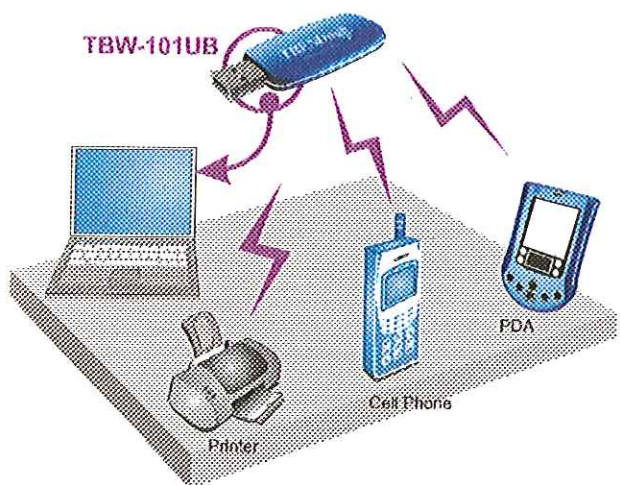
- 9 Gramm

**Zertifizierung:**

- FCC, CE



### Soluciones par Integrar



**Para ordenar por favor llame:**  
**1-888-326-6061**

### Productos Relacionados

→ **TBW-102UB :**  
Adaptador USB Bluetooth Wireless de Alto Poder

### Informacion de la orden

**TRENDware International, Inc.**  
3135 Kashiwa Street  
Torrance, CA. 90505. USA  
Tel: 310-891-1100  
Fax: 310-891-1111  
Web: [www.TRENDNET.com](http://www.TRENDNET.com)  
Email: [sales@trendware.com](mailto:sales@trendware.com)

**TRENDnet**  
TRENDware, USA