

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO Y CONSTRUCCIÓN DE UN CONTROL DOMÓTICO UTILIZANDO BLUETOOTH POR MEDIO DE UN PDA

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

ANDALUZ ORTIZ VÍCTOR HUGO

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

YÉPEZ RODRÍGUEZ JUAN FERNANDO

DIRECTOR: DR. ROBIN ÁLVAREZ RUEDA

Quito, febrero 2008

Ing. ANDALUZ ORTIZ VÍCTOR HUGO
Ing. YÉPEZ RODRÍGUEZ JUAN FERNANDO

Febrero, 2008

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO Y CONSTRUCCIÓN DE UN CONTROL DOMÓTICO UTILIZANDO BLUETOOTH POR MEDIO DE UN PDA

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

ANDALUZ ORTIZ VÍCTOR HUGO

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

YÉPEZ RODRÍGUEZ JUAN FERNANDO

DIRECTOR: DR. ROBIN ÁLVAREZ RUEDA

Quito, febrero 2008

DECLARACIÓN

Nosotros, ANDALUZ ORTIZ VÍCTOR HUGO y YÉPEZ RODRÍGUEZ JUAN FERNANDO, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

ANDALUZ O. VÍCTOR HUGO.

YÉPEZ R. JUAN FERNANDO.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por ANDALUZ ORTIZ VÍCTOR HUGO y YÉPEZ RODRÍGUEZ JUAN FERNANDO, bajo mi supervisión.

Dr. Robin Álvarez Rueda
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios por permitir cumplir uno de los objetivos más importantes de mi vida, además de manera profunda a todas las personas, familiares y amigos que han estado a mi lado a través de las diferentes etapas, buenas y malas respaldándome en todo momento para seguir adelante y poder culminar con éxito mis metas trazadas.

Un especial agradecimiento al Dr. Robín Álvarez quien me brindó su amistad y apoyo, a mi compañero de tesis quien fue una ayuda muy importante para lograr terminar con éxito este trabajo y a todos mis maestros que me supieron guiar a lo largo de mi vida estudiantil.

Victor Hugo

DEDICATORIA

Este trabajo esta dedicado a mis Padres.

A mi Padre por ser el ejemplo de un hombre recto, no conformista y que me ha inculcado el respeto y humildad.

A mi MAMI CHULA, a quien AMO con todo el corazón, ya que con su perseverante amor y cariño supo llenar de alegría mi vida en todo momento y pido a Dios que nunca me aleje de ella.

Victor Hugo

AGRADECIMIENTO

Agradezco a todas las personas que me apoyaron de alguna u otra manera, que hasta el día de hoy están conmigo pendientes de las cosas que hago y ayudándome a continuar y mirar siempre hacia adelante.

A todos mis amigos muchas gracias por estar cuando los necesitaba, a mis hermanos a los que extrañaba, a mis sobrinos que están en mi corazón, a mis tíos gracias por acogerme en su hogar y haberme tratado como un hijo más, a mi compañero de tesis que pese a las dificultades que se nos presentó estuvo conmigo en la lucha de no dejarnos vencer, a mi director gracias por haberse convertido en mi amigo, a mis profesores que me inculcaron su sabiduría para convertirme en un gran profesional.

Por todos ellos, he logrado alcanzar este gran objetivo de graduarme en la mejor universidad del país, mi querida Escuela Politécnica Nacional.

Juan Fernando.

DEDICATORIA

Dedico este proyecto de titulación a mis padres:

A mi papá que gracias a su apoyo en todo momento me permitió obtener este logro, dándome un gran ejemplo de perseverancia, enseñándome que los proyectos que se comienzan siempre hay que culminarlos, demostrándome que no importa las dificultades que se atraviesen en el camino siempre hay que encontrar la mejor manera de solucionar nuestros problemas.

A mi mamá que gracias a su cariño me ayudo a nunca sentirme solo, sabiendo que mi familia pase lo que pase siempre estará unida, enseñándome que hay que ayudar a los demás, porque en la vida alguna vez necesitaremos de alguien que nos extienda una mano.

Por ellos que son las personas que más quiero les dedico mi esfuerzo reflejado en este trabajo.

Juan Fernando

CONTENIDO

DECLARACIÓN.....	I
CERTIFICACIÓN.....	II
AGRADECIMIENTO.....	III
DEDICATORIA.....	V
CONTENIDO.....	VII
RESUMEN.....	XII
PRESENTACIÓN.....	XIV

CAPÍTULO 1: INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA.....	1
1.2. OBJETIVOS DEL PROYECTO.....	2

CAPÍTULO 2: ANTECEDENTES

2.1. CONCEPTOS UTILIZADOS EN ESTE PROYECTO.....	4
2.2. SISTEMAS DOMÓTICOS.....	8
2.2.1. TIPOS DE ARQUITECTURAS.....	9
2.2.2. CLASIFICACIÓN TECNOLÓGICA.....	10
2.2.3. SERVICIOS DE LA DOMÓTICA.....	12
2.3. ESTADO DEL ARTE.....	13
2.3.1. TECNOLOGÍAS APLICADAS A LA DOMÓTICA.....	13
2.3.1.1 Tecnología X-10.....	13

2.3.1.1.1	Aplicación de X-10.....	15
2.3.1.2	Tecnología Wi – Fi.....	16
2.3.1.2.1	Aplicación de Wi - Fi en la Domótica.....	17
2.3.1.3	Tecnología ZigBee.....	18
2.3.2.	CONTRIBUCIÓN A LA SOCIEDAD.....	19
2.4.	DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA INALÁMBRICA BLUETOOTH	
2.4.1.	PROTOCOLOS ESPECÍFICOS.....	23
2.4.2.	STACKS DE PROTOCOLO BLUETOOTH.....	24
2.4.3.	PERFILES DE BLUETOOTH.....	33
2.4.4.	ASPECTOS RELEVANTES DE UN DISPOSITIVO BLUETOOTH.....	37
2.4.4.1.	Seguridad de Bluetooth.....	38
2.4.5.	DISPOSITIVO KCWIREFREE KC -11.....	38
2.4.5.1.	Características generales del módulo KC-11.....	39
2.4.5.2.	Arquitectura de Software.....	40
2.4.5.2.1.	Pila de capas inferiores.....	41
2.4.5.2.2.	Pila de capas superiores.....	41
2.4.5.2.3.	Interface HCI.....	41
2.5.	DESCRIPCIÓN DE UN PDA.....	41
2.5.1.	SISTEMAS OPERATIVOS PARA PDAs.....	43
2.5.1.1.	PDA - PALM Versus PDA - POCKET PC.....	44
2.5.1.2.	Componentes de un PDA.....	44
2.5.2.	POCKET PC HP IPAQ H6315.....	46
2.5.2.1.	Especificaciones técnicas de la IPAQ h6315.....	47
2.6.	DESCRIPCIÓN DEL MICROCONTROLADOR ATMEGA 64.....	48
2.6.1.	ARQUITECTURA AVANZADA DEL RISC.....	49
2.6.2.	PROGRAMA Y DATOS DE MEMORIA.....	49

2.6.3. CARACTERÍSTICAS PERIFÉRICAS	49
2.6.4. Características especiales del microcontrolador.....	50
2.7. BASCOM-AVR.....	51
2.7.1. CARACTERÍSTICAS DE BASCOM-AVR.....	51
2.7.2. COMANDOS E INSTRUCCIONES.....	52

CAPÍTULO 3: DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DOMÓTICO

3.1. ESQUEMA GLOBAL DEL SISTEMA DOMÓTICO.....	54
3.2. ESPECIFICACIONES TÉCNICAS DEL EQUIPO.....	55
3.3. ELEMENTOS DEL CIRCUITO DE LA PASARELA DOMÓTICA.....	56
3.3.1. REGULADOR DE VOLTAJE DE 5 V DC.....	58
3.3.2. REGULADOR DE VOLTAJE DE 3.3 V DC.....	59
3.3.3. REGULADOR DE VOLTAJE DE 12V DC.....	59
3.3.4. OPTO - ACOPLADOR PC817.....	60
3.4. HARDWARE DE CONTROL.....	61

3.5. SOFTWARE DEL SISTEMA DOMÓTICO.....	63
3.5.1. PRESENTACIÓN DEL SISTEMA DOMÓTICO.....	65
3.5.2. INGRESO DE LA CONTRASEÑA.....	67
3.5.3. DISPOSITIVOS INALÁMBRICOS DEL PDA.....	72
3.5.4. PESTAÑA BLUETOOTH.....	74
3.5.4.1. Carga de Imágenes y botones del Formulario Menú.....	76
3.5.4.2. Conexión entre el PDA y la Pasarela Domótica.....	80
3.5.4.3. Proceso de Sincronización.....	82
3.5.5. PESTÑA ILUMINACIÓN.....	86

3.5.5.1. Forma de envío de los comandos.....	87
3.5.5.2. Cambio de estado de las cargas.....	91
3.5.6. PESTAÑA SEGURIDAD.....	93
3.5.6.1. Forma de envío de los comandos.....	94
3.5.7. PESTAÑA HORARIO.....	97
3.5.7.1. Forma de envío de los comandos.....	98
3.5.7.2. Modificación del horario programado de una determinada carga.....	102
3.5.7.3. Formato del proceso guardar.....	103

CAPITULO 4: PRUEBAS Y RESULTADOS

4.1. PRUEBAS CON REFERENCIA AL ALCANCE.....	109
4.1.1. CONDICIÓN DE PRUEBAS.....	109
4.1.2. ESTABLECIMIENTO DE LA CONEXIÓN.....	110
4.1.3. MANTENCIÓN DE LA CONEXIÓN.....	112
4.2. PRUEBAS DEL SISTEMA DOMÓTICO	114
4.2.1. ILUMINACIÓN.....	114
4.2.1.1. Estado de la carga.....	114
4.2.1.2. Control de las cargas eléctricas.....	115
4.2.2. SEGURIDAD.....	117
4.2.2.1. Muestreo de sensores.....	117

4.2.2.2. Habilitación de la alarma.....	118
4.2.3. HORARIO.....	119
4.2.3.1 Almacenamiento del Horario.....	119
4.2.3.2 Proceso de Simulación.....	120

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES	121
5.2. RECOMENDACIONES.....	124
5.3. TRABAJOS FUTUROS.....	125
5.4. APLICACIONES.....	126

REFERENCIAS BIBLIOGRÁFICAS.....	127
--	------------

ANEXO A: DIAGRAMAS DE FLUJOS.....	A -1
--	-------------

ANEXO B: DIAGRAMAS DE FLUJOS.....	B - 1
--	--------------

ANEXO C: HERRAMIENTA PARA PROGRAMACIÓN.....	C - 1
--	--------------

ANEXO D: CÓDIGOS DE PROGRAMACIÓN.....	D - 1
--	--------------

RESUMEN

El presente trabajo hace referencia al diseño y construcción de un Sistema Domótico de control de diferentes cargas eléctricas y dispositivos de seguridad. Para ello se ha desarrollado componentes, tanto de hardware como de software.

El componente de hardware es un sistema sencillo que mediante un PDA, controla distintas cargas eléctricas y la activación de un sistema de seguridad de manera inalámbrica. Asimismo, se puede tener acceso de modo manual desde los distintos puntos de trabajo correspondientes a dichas cargas eléctricas.

El proceso de comunicación inalámbrica utiliza tecnología Bluetooth, con la cual se crea una interfaz virtual de comunicación entre el usuario, por medio del PDA, y el hardware de control físicamente conectado con las distintas cargas eléctricas.

A continuación se detalla la estructura del documento:

Se inicia con una introducción sobre Domótica, el planteamiento del problema, a continuación se describe: los objetivos, el funcionamiento y las posibles soluciones del Sistema Domótico. Posteriormente se hace una descripción general de la tecnología inalámbrica "*Bluetooth*" con su stacks de protocolos.

En segundo lugar, se trata sobre los sistemas operativos del PDA, enfocándonos en la Pocket PC, a continuación se expone el estado del arte en cuanto a sistemas de control inalámbrico por medio de PDA con lo cual se tendrá una visión más clara de los avances tecnológicos existentes que se puede aprovechar, en base a esto, entregaremos nuestro aporte en beneficio del campo de la Domótica.

Posteriormente se explica el diseño de cada una de las partes del Sistema Domótico, la construcción de las mismas y el desarrollo del software tanto a nivel del microcontrolador como del programa de alto nivel que está instalado en el PDA.

Una vez listo y en funcionamiento, se muestran los resultados de las mediciones realizadas, la caracterización del dispositivo y finalmente se presentan las conclusiones, recomendaciones y trabajos futuros.

PRESENTACIÓN

En la actualidad una vivienda tiende a ser un sistema dinámico, debido a que sus requerimientos han ido evolucionando, a tal punto que en ella se puede encontrar sistemas de climatización, juegos de luces, redes telefónicas, de seguridad e incluso sistemas informáticos. Todos estos nuevos conceptos alejan a las viviendas actuales de la vieja concepción de un simple lugar de refugio.

El control inteligente de los grandes edificios que incluyen cámaras, megáfonos, redes informáticas, etc., nos obliga a desterrar la antigua idea estática.

Estas tecnologías actuales están siendo aplicadas a viviendas normales, siendo éste el objetivo del presente proyecto: *diseñar un Sistema Domótico* que permita controlar las cargas eléctricas del hogar desde un PDA, por medio de la tecnología de comunicación inalámbrica Bluetooth.

Mediante este sistema se puede realizar el control de diferentes tipos de cargas eléctricas en tiempo real, sin la necesidad de que el usuario esté físicamente presente en el lugar donde se encuentra la carga a controlar, ya que la tecnología utilizada permite un radio de cobertura de señal para el control de hasta cuarenta metros sin tener una línea de vista directa. Además, se puede programar el horario de encendido y apagado de una determinada carga eléctrica, con lo cual se consigue la simulación de presencia, evitando así robos futuros. Adicionalmente se cuenta con un sistema de seguridad, el cual puede o no ser habilitado por medio del PDA cuando el usuario lo necesite.

A medida que la tecnología avanza, la Domótica se inclina al control inalámbrico, por lo cual las empresas están aplicando dicha tecnología.

Con el presente trabajo queremos dar nuestro aporte al desarrollo de la Domótica en nuestro país.

CAPÍTULO 1

INTRODUCCIÓN

Domótica es un concepto que se refiere a la integración de las distintas tecnologías en el hogar mediante el uso simultáneo de la electricidad, la electrónica, la informática y las telecomunicaciones.

Su fin es mejorar la seguridad, el confort, la flexibilidad, las comunicaciones, el ahorro energético, que pueden estar conectados por medio de redes interiores y exteriores de comunicación, para facilitar el control general de los usuarios de los sistemas y ofrecer nuevos servicios.

1.1. PLANTEAMIENTO DEL PROBLEMA

Hoy en día, con el avance de la tecnología, la sociedad busca constantemente un sistema que le brinde comodidad en todos los aspectos de su vida cotidiana.

El hogar de una persona es siempre su refugio preferido y por ello pretende continuamente que éste sea un lugar confortable y seguro para quienes lo habitan.

Uno de los sectores favorecidos por la Domótica es el de los discapacitados físicos. El no poder movilizarse con facilidad, hace muy difícil realizar tareas que son elementales para un sujeto con capacidades normales. La domótica puede ayudar al usuario a encender o apagar una luminaria, a abrir o cerrar puertas y ventanas, todo esto comandado desde un PDA, a través de un sistema de comunicación inalámbrico.

Por todo lo anterior, en el presente trabajo se plantea implementar un sistema en el que el usuario tenga el control, de encender y apagar luminarias de manera inalámbrica por medio de un PDA. Además de tener un sistema de seguridad básico el cual debe ser habilitado cuando el usuario lo desee, el mismo que

deberá poseer sensores que detecten el movimiento y la apertura de puertas, esté control se espera que se lo pueda realizar en cualquier parte del domicilio.

En el caso de las personas discapacitadas, este sistema facilitará el control de luminaria y monitoreo de sensores de un sistema de seguridad, puesto que no es indispensable que se encuentren físicamente en el lugar ya que podrán utilizar su PDA para comandar cualquiera de ellas.

1.2. OBJETIVOS DEL SISTEMA DOMÓTICO

El objetivo general del proyecto está orientado a cumplir el siguiente ítem.

- Diseñar e implementar un sistema que, por medio de un PDA y mediante tecnología Bluetooth, permita realizar el control de luminarias y de sensores de seguridad en una vivienda.

Para cumplir este objetivo general, será necesario abordar los siguientes objetivos específicos:

- Que la distancias de comunicación inalámbrica sea por lo menos de diez metros, entre el PDA y la pasarela domótica.
- Se deberá colocar una clave para poder ingresar a la aplicación del Sistema Domótico, que se encontrará instalada en el PDA
- Que se pueda controlar desde el PDA, diez cargas eléctricas ON/OFF las mismas que deben soportar una corriente como máximo de diez amperios cada una de ellas. Además dichas cargas eléctricas (focos) deben poder ser controladas desde cada uno de los interruptores respectivos.
- Se deberá poder programar el horario de encendido y apagado de cada unas de las cargas eléctrica anteriormente mencionadas.

- El Sistema Domótico deberá tener un sistema de seguridad el cual debe ser habilitado o deshabilitado desde el PDA por medio del usuario.
- En el sistema de seguridad se debe tener la posibilidad de conectar cuatro sensores normalmente cerrados, un sensor normalmente abierto y una sirena la cual sonará cuando se active algún sensor.
- Implementar un software de alto nivel para un PDA, que permitirá al usuario controlar desde un ambiente visual cada una de las cargas anteriormente mencionadas y del sistema de seguridad.
- Construir el hardware necesario que permita realizar el control indicado anteriormente de las cargas eléctricas y del sistema de seguridad.

1.3 FUNCIONAMIENTO DEL SISTEMA DOMÓTICO

El Sistema Domótico deberá constar de una carátula en la que se indique el tema del proyecto, y sus actores. Para ingreso al sistema se presentará una pantalla en la cual se debe colocar una clave de ingreso.

El Sistema Domótico básicamente deberá tener: control de iluminación y control de seguridad, los cuales deberán funcionar de la siguiente manera:

a) Control de iluminación

Deberá constar de dos partes: La primera permitirá el encendido y apagado de luminarias de manera inalámbrica utilizando un PDA, desde distintos lugares del domicilio; para lo cual en el PDA se debe indicar el estado actual de la carga eléctrica a controlar, además dichas cargas también deben ser controladas desde los interruptores de cada uno de ellas.

La segunda parte dará la opción de simular presencia en la vivienda, al encender y apagar luminarias de manera automática en diferentes intervalos de tiempo, estos intervalos de tiempos deben ser ingresados por

el usuario a su conveniencia. Así mismo se deberá poder conocer el horario que se encuentra programada cada carga eléctrica.

b) Control de seguridad

Para el control de seguridad se tendrá en consideración los siguientes aspectos: el sistema de seguridad deberá ser habilitado o deshabilitado por medio del usuario desde el PDA. En una de las pantallas de la aplicación del Sistema Domótico se podrá visualizar el estado de los sensores.

Se considerará el conectar cuatro sensores normalmente cerrados y un sensor normalmente abierto, ya sean estos magnéticos o de movimiento.

1.4 SOLUCIONES AL SISTEMA DOMÓTICO

Se puede plantear varias alternativas para la solución del problema anteriormente señalado por ejemplo:

- Para la comunicación entre el PDA y la pasarela se lo puede realizar por medio de las tecnologías inalámbricas tales como: infrarrojo, Wi-Fi, y Bluetooth que son tecnologías que actualmente se encuentran incorporadas en un PDA.
- Mientras que para el control desde la pasarela domótica a las cargas eléctricas y a los sensores del sistema de seguridad se lo puede realizar de la siguiente manera: por medio de cableado desde la pasarela hasta cada uno de los sensores y de las cargas que se deseen controlar. O se lo puede realizar de manera inalámbrica utilizando la tecnología: Bluetooth, Wi-Fi, radio frecuencia y X-10 (utiliza la red eléctrica para la transmisión de datos).

En conclusión para la solución del problema se puede hacer combinación entre las diferentes opciones anteriormente mencionadas.

Para la solución del problema se ha optado por que la comunicación entre la pasarela domótica y el PDA, se lo realizará por medio de la tecnología Bluetooth, ya que provee ventajas significativas sobre las otras tecnologías inalámbricas; debido a que funciona con RF (radio frecuencia) y no está sujeto a limitaciones propias de transmitir en estas frecuencias. La distancia de conexión en Bluetooth es de 10 metros o más, dependiendo del incremento de la potencia de su transmisor; además, los dispositivos no necesitan estar en línea de vista, ya que las señales de RF pueden atravesar paredes y otros objetos no metálicos sin ningún problema.

Mientras que para el control desde la pasarela domótica hacia las diferentes cargas y sensores de seguridad se lo realizará por medio de cableado, ya que esta opción resulta ser la más económica ver figura 1.1.

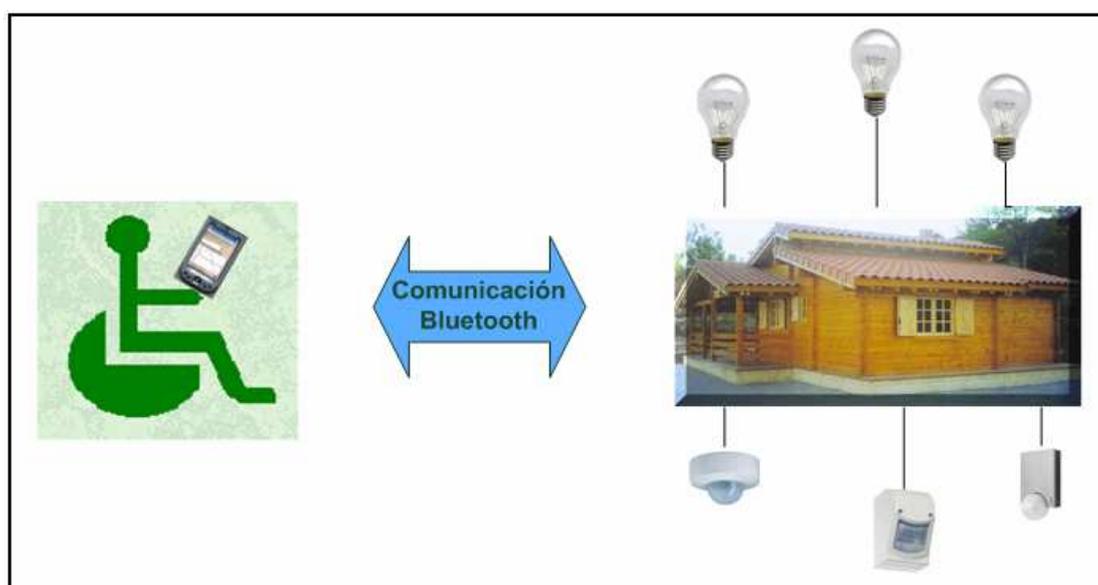


Figura 1.1. Solución al planteamiento del problema

La figura 1.2, muestra un diagrama de bloque en el cual se indica de manera global el Sistema Domótico, el cual debe constar de una fuente DC la misma que alimentará a todos los dispositivos a utilizar, un microcontrolador el mismo que

debe realizar la función de controlar las diferentes cargas eléctricas y los dispositivos de seguridad (sensores y bocina), asimismo debe realizar la comunicación bidireccional con un dispositivo Bluetooth mediante el puerto de comunicación serial. El dispositivo Bluetooth debe permitir la comunicación bidireccional entre el microcontrolador y el PDA, y por último un PDA el mismo que debe poseer la aplicación del Sistema Domótico el mismo que debe ser amigable para el usuario ya que por medio de el se debe controlar las cargas eléctricas y el sistema de seguridad.

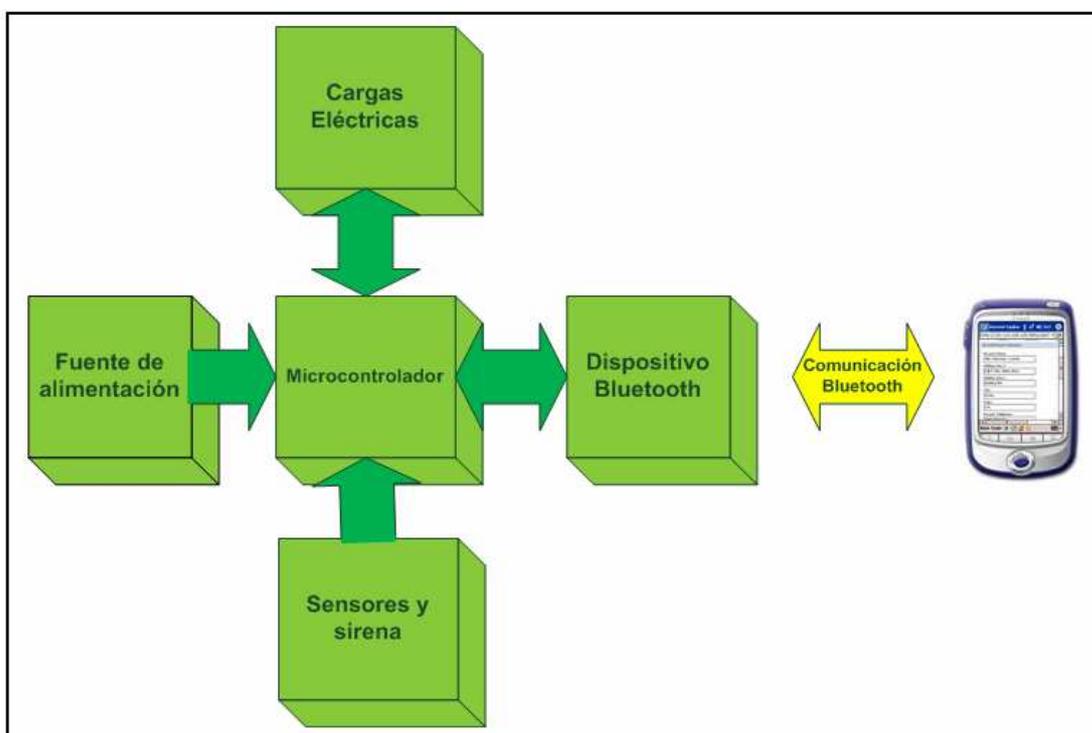


Figura 1.2. Diagrama de bloque del Sistema Domótico

CAPÍTULO 2

ANTECEDENTES

2.1. SISTEMAS DOMÓTICOS

Los sistemas domóticos consisten de uno o varios elementos. Se puede hacer la siguiente clasificación de los dispositivos de un sistema:

- **Controlador:** en instalaciones centralizadas, es la central que gestiona el sistema. En éste, reside toda la inteligencia del sistema y suele tener las interfaces de usuario necesarias para presentar la información a este (pantalla, teclado, monitor, etc.).
- **Actuador:** es el dispositivo de salida capaz de recibir una orden del controlador y realizar una acción (encendido/apagado, subida/bajada de persiana, apertura/cierre de electroválvula, etc.).
- **Sensor:** es el dispositivo que está, de forma permanente, monitorizando el entorno, con el propósito de generar un evento que será procesado por el controlador. Algunos ejemplos de estos sensores son de temperatura, de viento, de humedad, de humo, de escape de agua o gas, etc.

2.1.1. TIPOS DE ARQUITECTURAS DEL SISTEMA DOMÓTICO

Desde el punto de vista de donde reside la inteligencia del sistema domótico, hay dos arquitecturas diferentes:

- **Arquitectura Centralizada:** un controlador centralizado recibe información de múltiples sensores y, una vez procesada, genera las órdenes oportunas para los actuadores (Figura 2.1).

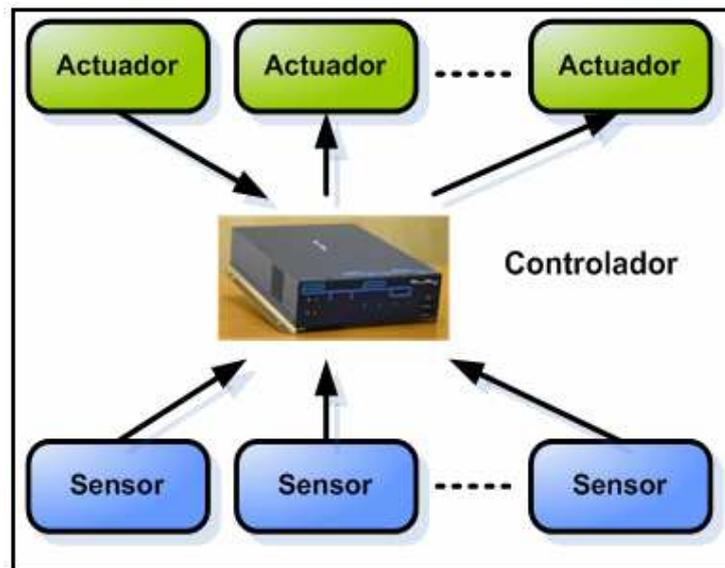


Figura 2.1. Arquitectura Centralizada

- **Arquitectura Distribuida:** en este caso, no existe la figura del controlador centralizado sino que toda la inteligencia del sistema está distribuida por todos los módulos, sean sensores o actuadores. Suele ser típico de los sistemas de cableado en bus (Figura 2.2).

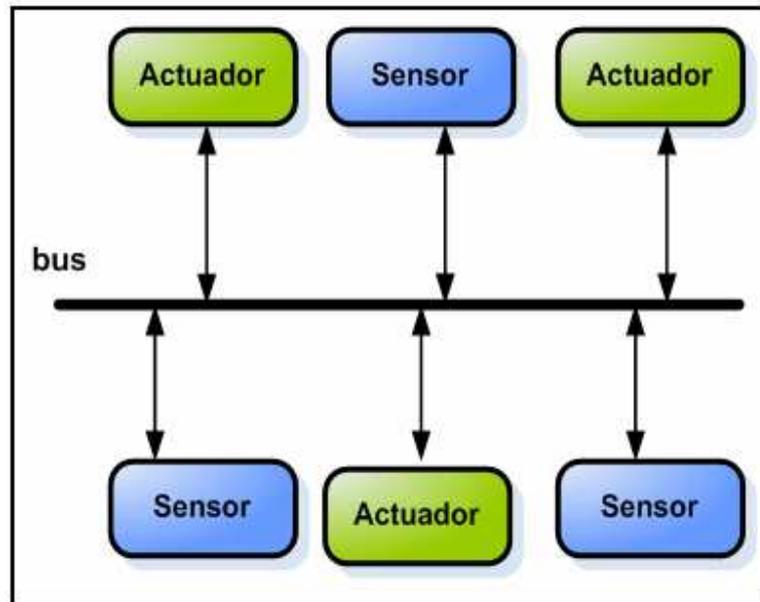


Figura 2.2. Arquitectura Distribuida

Hay que destacar que algunos sistemas usan un enfoque mixto, es decir, son sistemas con arquitectura descentralizada, que disponen de varios pequeños dispositivos capaces de adquirir y procesar la información de múltiples sensores y transmitirlos al resto de dispositivos distribuidos en la vivienda.

2.1.2. CLASIFICACIÓN TECNOLÓGICA UTILIZADA EN SISTEMA DOMÓTICO

Se pueden clasificar las centrales en dos tipos a nivel tecnológico:

- **Centrales cableadas:** todos los sensores y actuadores están unidos a la central mediante cable. Esta central tiene normalmente una batería de respaldo, en caso de fallo del suministro eléctrico normal. (Figura 2.3).

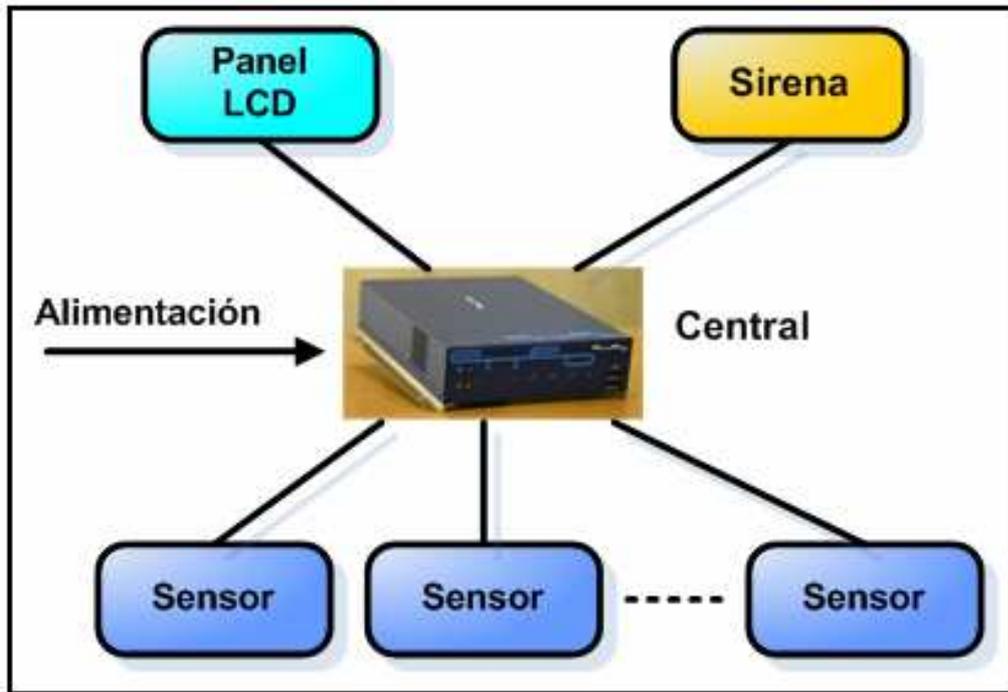


Figura 2.3. Sistema Cableado

- **Centrales inalámbricas:** en este caso usan sensores inalámbricos, alimentados por pilas o baterías, que transmiten vía radio la información de los eventos a la central, la cual está alimentada por la red eléctrica y también tiene sus baterías de respaldo (Figura 2.4).

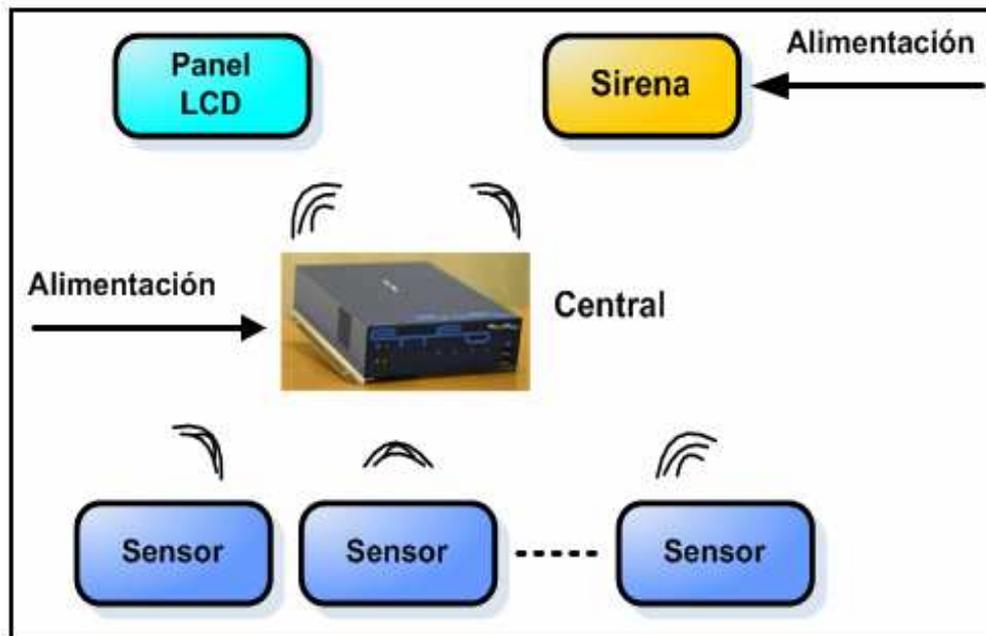


Figura 2.4. Sistema Inalámbrico

- **Centrales mixtas:** combinan el sistema cableado con el inalámbrico.

2.1.3. SERVICIOS DE LA DOMÓTICA

Los servicios que ofrece la domótica se pueden agrupar según cuatro aspectos principales:

a) En el ámbito del ahorro energético

- Climatización: programación y zonificación
 - La forma de controlar la climatización de una vivienda es la conexión o desconexión de todo el sistema de climatización. Se puede realizar esto según una programación horaria, según presencia de personas en el hogar o de forma manual. Con estos modos de funcionamiento el sistema sólo garantiza el establecimiento de una temperatura de

consigna única para toda la vivienda, de forma parecida a la existencia de un termostato de ambiente convencional.

- Gestión eléctrica
 - Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado. Reduce la potencia contratada.
 - Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
- Uso de energías renovables
 - Control de paneles solares con sus respectivas baterías.

b) En el ámbito del nivel de confort

- Iluminación.
 - Apagado general de todas las luces de la vivienda.
 - Automatización del encendido/apagado en cada punto de luz.
 - Regulación de la iluminación según el nivel de luminosidad ambiente.
- Automatización de todos los distintos sistemas/ instalaciones / equipos dotándolos de control eficiente y fácil manejo.
 - Motorizar puertas y ventanas para que pueden ser integradas en el sistema de domótica.
 - Las persianas y techos pueden ser controlados según la temperatura interior o la situación climatológica del exterior
- Integración del portero al teléfono, o del video portero al televisor.
 - Monitoreo de las personas que desean ingresar al domicilio mediante una cámara de vigilancia.
- Control vía Internet.
 - Control y monitoreo de toda la casa mediante el uso de cámaras de video.
- Gestión Multimedia y del ocio electrónico.

- Generación de macros y programas de forma sencilla por parte del usuario.

c) En el ámbito de la protección patrimonial

- Simulación de presencia.
 - Encendido y apagado de luminarias y diversos electrodomésticos, en intervalos de tiempos aleatorios o programados.
- Detección de incendio, fugas de gas, escapes de agua.
 - Mediante sensores de gas o de presión, humedad, etc., se puede sensor fuga de gas o de agua acontecimientos no normales, con lo que se puede realizar alguna subrutina de control.
- Alerta médica, tele-asistencia.
- Cerramiento de persianas puntual y seguro.
 - Motorizar persianas y ventanas para incorporar las en el sistemas domótico.
- Acceso a Cámaras IP.
 - Monitoreo de todo la vivienda.

d) En el ámbito de las comunicaciones

- Universalidad en el control tanto externo como interno, control remoto desde Internet, PC, mandos inalámbricos (p.ej. PDA con Wi - Fi).
 - Tener un control en el que el usuario se pueda movilizarse con facilidad y no un panel de control fijo o estático
- Transmisión de alarmas.
 - Al surgir algún inconveniente o evento fuera de lo normal se activen diversos tipos de alarma ya sean sonoras o audibles.
- Intercomunicaciones.
 - Tener sistemas de respaldo, repetidores o medios de comhnicación hacia el exterior de la vivienda.

2.2. ESTADO DEL ARTE

La domótica es un concepto que se refiere a la aplicación de las distintas tecnologías en el hogar, mediante el uso simultáneo de la electricidad, la electrónica, la informática y las telecomunicaciones. Su fin es mejorar la seguridad, el confort, la flexibilidad, las comunicaciones, el ahorro energético, facilitar el control integral de los sistemas para los usuarios y ofrecer nuevos servicios.

2.2.1. TECNOLOGÍAS APLICADAS A LA DOMÓTICA

Entre las tecnologías aplicadas a la domótica se tiene las siguientes:

- a) X-10; Sistema de control por corrientes portadoras
- b) Wi – Fi; Sistema de control inalámbrico
- c) Tecnología ZigBee
- d) Bluetooth

2.2.1.1 Tecnología X-10

Desde hace veinte años X – 10 es líder en el mercado de automatización del hogar, ofreciendo gran variedad de productos y accesorios. Es un sistema de control que utiliza la red eléctrica del lugar para poder transmitir órdenes y comandos a los receptores, cada punto de interruptor usa a un receptor para recibir y ejecutar dichas órdenes. Como todos los módulos tienen una dirección específica y única, ellos pueden ser controlados por separado.

Entre los dispositivos de la tecnología X – 10 se tiene:

- **Transmisores** estos transmisores envían una señal especialmente codificada de bajo voltaje que es superpuesta sobre el voltaje del cableado. Un transmisor es capaz de enviar información hasta 256 dispositivos sobre el

cableado eléctrico. Múltiples transmisores pueden enviar señales al mismo módulo.

- **Receptores** los receptores pueden comunicarse con 256 direcciones distintas. Cuando se usan con algunos controladores de computadoras, estos dispositivos pueden reportar su estado actual.
- **Bidireccionales** estos dispositivos toman la señal enviada por los dispositivos transmisores. Una vez que la señal es recibida el dispositivo responde encendiéndose (ON) o apagándose (OFF). Los receptores generalmente tienen un código establecido por el usuario para indicar la dirección del dispositivo. Múltiples dispositivos con el mismo código pueden coexistir y responder al mismo tiempo dentro de una misma casa.
Los dispositivos bidireccionales tienen la capacidad de responder y confirmar la correcta realización de una orden, lo cual puede ser muy útil cuando el sistema X-10 está conectado a un programa de ordenador que muestre los estados en que se encuentra la instalación domótica de la vivienda. Éste es el caso del programador para PC.
- **Inalámbricos** unidades que permiten conectarse a través de una antena y enviar señales de radio desde una unidad inalámbrica e inyectar la señal X-10 en el cableado eléctrico (como los controles remotos para abrir los portones de los garajes). Estas unidades no están habilitadas para controlar directamente a un receptor X-10, debe utilizarse un módulo transmisor.

2.2.1.1.1 Aplicación de X-10

El aporte de X-10 en la Domótica se podrá resumir en los diferentes tipos de control que se logra obtener:

- **Por un mando local**

Para el control de luces u otras cargas eléctricas, siempre se necesita al menos de un transmisor y un receptor.

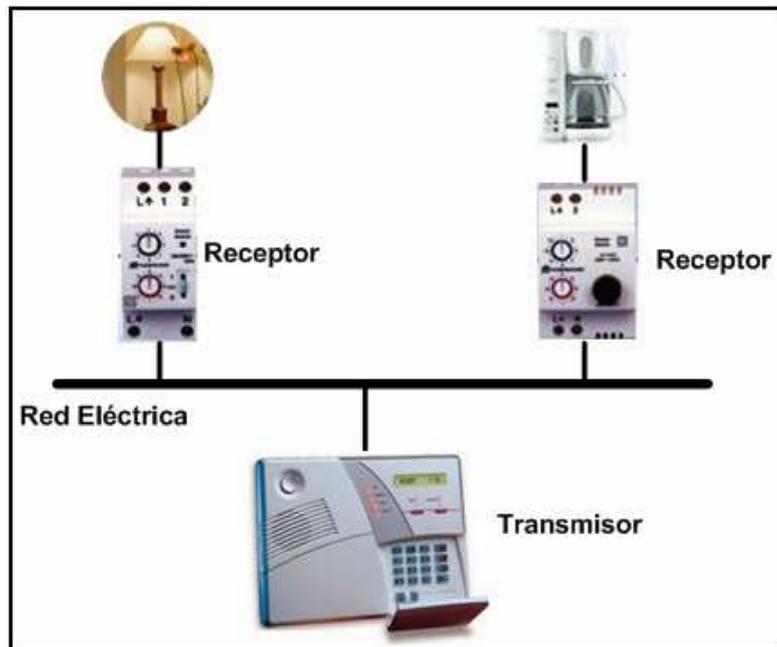


Figura 2.5. Mando local

- **Por mando a distancia**

La figura 2.6, representa el esquema de control de distintas cargas eléctricas vía un mando a distancia. El mando (control remoto), puede ser usado para controlar cualquier carga o dispositivo de la casa. Para el mando a distancia intervienen los siguientes dispositivos: un control remoto, un transmisor, un receptor y la carga que se desee controlar.

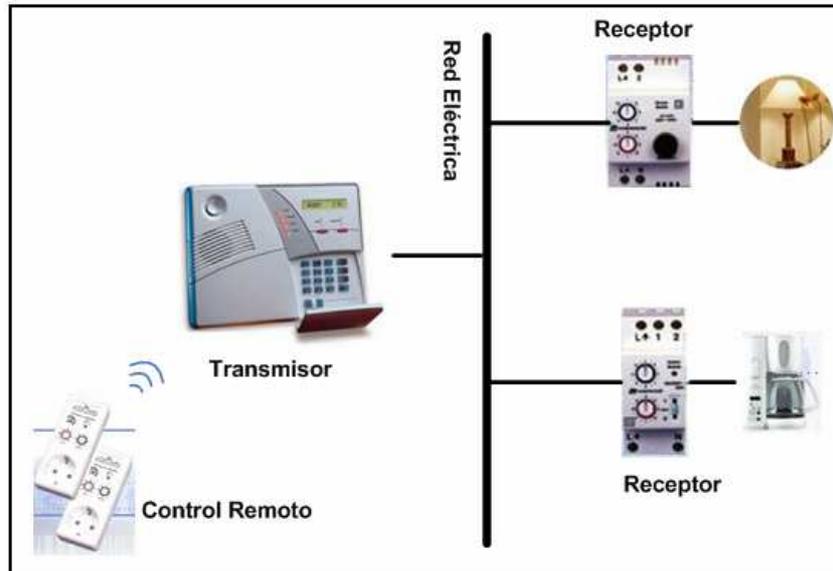


Figura 2.6. Mando a distancia

2.2.1.2 Tecnología Wi - Fi

Las tecnologías para redes inalámbricas WLAN (*wireless lan*) cumplen con los estándares de la serie IEEE 802.11, Wi - Fi es el estándar IEEE 802.11b, que hoy en día es una de las tecnologías de comunicación inalámbrica más utilizada, Wi - Fi es una abreviatura de Wireless Fidelity.

En la actualidad se puede encontrar en el mercado productos con los estándares:

- 802.11b, que puede transmitir hasta 11 Mb/seg, y
- 802.11g, que puede transmitir hasta 54 Mb/seg.

De hecho, con su velocidad y alcance (Posee un radio de acción de hasta 100 metros en espacios abiertos y de unos 45 metros en recintos cerrados con paredes)¹ lo que lo convierten en una fórmula perfecta para el control de diferentes cargas eléctricas y el acceso a Internet sin cables.

¹ www.34.com/unique/WiFiWan.asp

Para tener una red inalámbrica se precisa de un punto de acceso, el mismo que se conecta al módem, y un dispositivo Wi - Fi que se conectaría en algún otro dispositivo con el que se desee realizar la comunicación (PDA, impresora, etc.).

Es aconsejable mantener el punto de acceso en un lugar apropiado para que la recepción/emisión tenga mayor fluidez. Incluso si se encuentra que la velocidad no es tan alta como debería, quizás se deba a que los dispositivos no se encuentren situados adecuadamente o quizás existan barreras entre ellos como: paredes, metal o puertas.

Al tratarse de conexiones inalámbricas, no es difícil que alguien intercepte la comunicación y tenga acceso al flujo de información. Por esto, es recomendable la encriptación de la transmisión para emitirla en un entorno seguro. En Wi - Fi esto es posible gracias al WPA2 (estándar 802.11i), que es una mejora relativa a WPA. WPA es un sistema para proteger las redes inalámbricas (Wi-Fi); creado para corregir las deficiencias del sistema previo WEP.

2.2.1.2.1 Aplicación de Wi - Fi en la Domótica

En la actualidad hay muchos fabricantes los cuales han aplicado la tecnología Wi-Fi al servicio de la Domótica, entre uno de ellos es Samsung que ha creado la red Home VITA, la solución para lograr un control digital total dentro del hogar. Su tecnología cubre las necesidades de entretenimiento, trabajo, confort y seguridad del usuario.

Todos los elementos digitales que integran el hoga controlan desde el Home Pad (Figura 2.7), una pantalla táctil desde donde se los configura de forma sencilla. Incluso con un teléfono móvil WAP o desde Internet es posible efectuar los mandos de la casa.



Figura 2.7. Home Pad

El hogar digital de Samsung se basa en la tecnología Wi - Fi (sin cables), para las comunicaciones del Home Pad con el resto de la red Home del hogar. Mediante este dispositivo se puede controlar la televisión, las luces, las persianas, y prevenir fugas de agua o de gas.

2.2.1.3 Tecnología ZigBee

ZigBee es un protocolo de comunicaciones inalámbrico, similar al Bluetooth, y basado en el estándar para redes inalámbricas de área personal (WPANs) IEEE 802.15.4.

Surge como resultado de una alianza de más de 200 empresas sin fines lucro, la mayoría de ellas fabricantes de semiconductores, con el objetivo de conseguir el desarrollo e implantación de una tecnología inalámbrica a bajo costo.

ZigBee, también conocido como "HomeRF Lite", es una tecnología inalámbrica con velocidades comprendidas entre 20 kb/s - 250 kb/s y rangos de 10m a 75m, se puede usar las bandas libres ISM de 2,4 GHz, 868 MHz (Europa) y 915 MHz (EEUU).

Una red ZigBee puede estar formada por hasta 255 nodos los cuales tienen la mayor parte del tiempo dormido al transceiver ZigBee con el fin de consumir menos energía que otras tecnologías inalámbricas.

El objetivo es que un sensor equipado con un transceiver ZigBee pueda ser alimentado con dos pilas AA durante al menos 6 meses y hasta 2 años.

Destacan empresas como *Invensys*, *Mitsubishi*, *Honeywell*, *Philips* y *Motorola* que trabajan para crear un sistema estándar de comunicaciones, vía radio y bidireccional. Principalmente, el ámbito en el que se usará será la Domótica, debido a su bajo consumo, a su sistema de comunicaciones vía radio (con topología *MESH*) y a su fácil integración (se pueden fabricar nodos con muy poca electrónica).

Para la Domótica, la clase de comunicaciones que se producen en una red de sensores o actuadores es muy variada, se puede encontrar que muchas de estas comunicaciones se realizan con pequeños paquetes de datos: para enviar información de un sensor (por Ej., activado = detecta humo), o simplemente para controlar el estado de los mismos. Además de ser paquetes pequeños de información, la gran mayoría de los dispositivos pueden estar 'dormidos' hasta que envíen la información (porque no ocurre nada) y activarse al detectar algo, para el obtener el máximo ahorro de energía.

2.2.1.4 Tecnología Bluetooth

Bluetooth es una tecnología utilizada para conectividad inalámbrica de corto alcance entre dispositivos tales como PDAs, teléfonos celulares, teclados, computadoras de escritorio y portátiles, proyectores e impresoras; permitiendo transferir datos, voz e imágenes entre equipos que incorporen éste estándar.

La principal característica que lleva a la radiofrecuencia a ser superior sobre otras comunicaciones inalámbricas, es transmitir información con la ventaja de superar obstáculos entre el emisor y receptor. La tecnología Bluetooth es una especificación que presenta seguridad en el intercambio de datos y su principal objetivo es reemplazar los cables que conectan unos dispositivos con otros por medio de un enlace de radio universal y de corto alcance.

2.2.2. CONTRIBUCIÓN A LA SOCIEDAD ECUATORIANA

Todos los esfuerzos innovadores de las empresas, se centran en aplicar las nuevas tecnologías en aparatos que proporcionen al usuario los cuatro puntos en los que se basa todo sistema domótico: aumentar la seguridad, ahorrar energía, incrementar el confort, la comodidad y optimizar las comunicaciones.

En la última Feria de la Domótica y el Hogar Digital denominada Domogar 2007, celebrado en Feria Valencia durante los días 7, 8, 9 y 10 de Marzo de 2007, se promocionaron los avances de las tecnologías empleadas en cada uno de los sistemas domóticos e inmóticos disponibles en el mercado.

Es importante recalcar que no hubo un fabricante que exponga tecnología Bluetooth al servicio de la Domótica, ni tampoco que se utilice un PDA como medio de control del hogar.

Por lo acotado anteriormente, el presente trabajo contribuirá en el avance de la Domótica, ya que el mismo manejará tecnología inalámbrica Bluetooth como medio de comunicación entre la pasarela y el PDA, con lo cual se podrá controlar los diferentes dispositivos.

Puesto que el alcance de los dispositivos Bluetooth es de hasta 100 metros, se podrá realizar el control de las diferentes cargas, desde diferentes puntos de la vivienda.

2.3. DESCRIPCIÓN GENERAL DE LA TECNOLOGÍA INALÁMBRICA BLUETOOTH²

2.3.1 CARACTERÍSTICAS DE BLUETOOTH

² www.bluetooth.com/dev/specifications.asp

La tecnología inalámbrica Bluetooth, ofrece una forma de remplazar cables y enlaces infrarrojos por un enlace de radio universal de corto alcance, con capacidad de crear pequeñas redes PANs.

Bluetooth es un sistema de radio que opera en la banda de frecuencia libre de 2.4 GHz. Esta banda de frecuencia está disponible en la mayor parte del mundo. Las bandas, 902 - 924 MHz, 2 400 - 2 483,5 MHz y 5 725 - 5 850 MHz son libres para el uso de los sistemas de espectro ensanchado.

Bluetooth soporta un canal de datos asíncrono de hasta tres canales de voz simultáneos. El canal asíncrono soporta comunicación simétrica y asimétrica. En la comunicación asimétrica pueden ser enviados 723.3 Kb/s desde el servidor y 57.6 Kb/s hacia el servidor, mientras que en la comunicación simétrica pueden ser enviados 433 Kb/s en ambas direcciones.

2.3.2 CANALES DE TRANSMISIÓN BLUETOOTH

Bluetooth utiliza 79 frecuencias de radio diferentes con un ancho de banda de 1 MHz cada uno, las cuales son accedidas de acuerdo a una secuencia de saltos aleatoria a una velocidad de 1600 saltos/segundo. El canal está dividido en time-slots (ranuras de tiempo), cada ranura corresponde a una frecuencia de salto y tiene una duración de 625 μ s. Cada secuencia de salto en una Piconet está determinada por la dirección del maestro de la misma.

En una transmisión, cada paquete debe estar alineado con el inicio de una ranura y puede tener una duración de hasta cinco ranuras de tiempo. Durante la transmisión de un paquete la frecuencia es fija. Para evitar fallas en la transmisión, el maestro inicia enviando en las ranuras de tiempo pares y los esclavos en las ranuras de tiempo impares. En la Figura 2.8, se puede observar este esquema de transmisión.

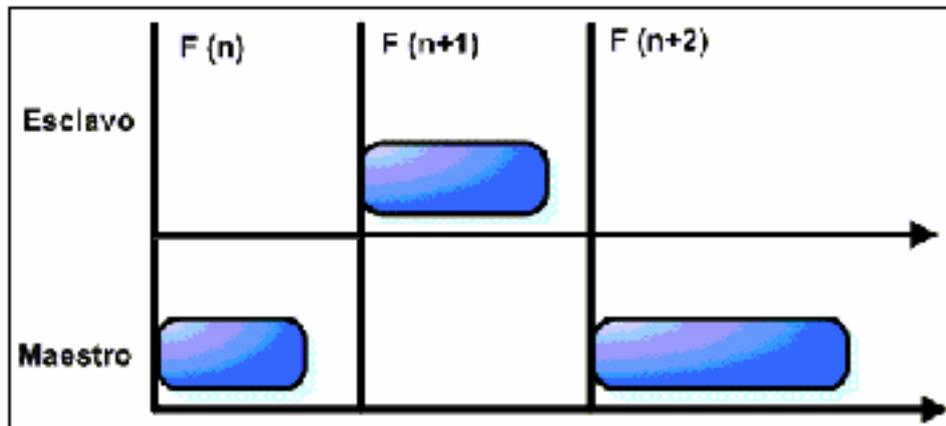


Figura 2.8. Transmisión en una Piconet

2.3.3 PICONETS Y SCATTERNETS

La comunicación sobre Bluetooth es buena para enlaces SCO o enlaces ACL. El enlace SCO es una conexión simétrica punto-a-punto entre el maestro y un esclavo específico. Para lograr la comunicación, el enlace SCO reserva ranuras en intervalos regulares en la iniciación, por esto el enlace puede ser considerado como una conexión de conmutación de circuitos. El enlace ACL es un enlace punto-a-multipunto entre el maestro y uno o más esclavos activos en la piconet. Este enlace de comunicación es un tipo de conexión de conmutación de paquetes. Todos los paquetes son retransmitidos para asegurar la integridad de los datos. El maestro puede enviar mensajes broadcast (de difusión) a todos los esclavos conectados dejando vacía la dirección del paquete, es así como todos los esclavos leerán el paquete.

La interconexión de varias Piconets forma una scatternet. En la Figura 2.9 se puede observar una Piconet donde el PC actúa como maestro y los otros dispositivos son conectados como esclavos.



Figura 2.9. Diagrama de una Piconet

Solamente el último esclavo direccionado está autorizado para enviar en el time slot de los esclavos, esto no causa problemas ya que el maestro siempre está inicializando todas las conexiones y transmisiones nuevas. Cada esclavo espera las oportunidades de conexión dadas por el maestro. Los paquetes pueden ser más grandes que un time slot, debido a esto el maestro puede continuar enviando en los time slots impares y viceversa. El sistema de reloj del maestro sincroniza a toda la Piconet.

El maestro nunca ajusta su sistema de reloj durante la existencia de una Piconet, son los esclavos quienes adaptan sus relojes con un offset de tiempo con el fin de igualarse con el reloj del maestro, éste offset es actualizado cada vez que se ha recibido un paquete desde el maestro.

2.3.4 ENVIO DE INFORMACIÓN

La información enviada sobre el canal de la Piconet es convertida en paquetes, éstos son enviados y el receptor los recibe iniciando por el bit menos significativo.

Como se observa en la Figura 2.10, el formato de paquete general consta de tres campos: código de acceso, cabecera y carga útil.



Figura 2.10. Formato de paquete general

• **Código de acceso.** Es usado para sincronización e identificación. Todos los paquetes comunes que son enviados sobre el canal de la Piconet están precedidos del mismo código de acceso al canal. Existen tres tipos diferentes de código de acceso:

- *Código de acceso al canal* - Para identificar los paquetes sobre el canal de la Piconet.
- *Código de acceso de dispositivo* – Para procedimientos de señalización especiales, paging (servicio para transferencia de señalización o información en un sentido), entre otros.
- *Código de Acceso de Búsqueda (IAC)* – llamado IAC general cuando se quiere descubrir a otras unidades Bluetooth dentro del rango, o IAC dedicado a descubrir unidades de un tipo específico.

• **Cabecera de paquete.** Como se observa en la Figura 2.11, la cabecera de paquete consta de seis campos:

- *Dirección* – Una dirección de dispositivo para distinguirlo de los demás dispositivos activos en la Piconet.
- *Tipo* – Define qué tipo de paquete es enviado.
- *Flujo* – El bit de control de flujo es usado para notificar al emisor cuándo el buffer del receptor está lleno.

- *ARQN* – Acknowledge Receive Data o reconocimiento de datos recibidos.
- *SEQN* – Sequential Numbering o numeración secuencial para ordenar los datos sobre el canal.
- *HEC* – Chequeo de redundancia cíclica de cabecera.



Figura 2.11. Formato de cabecera de paquete

- **Carga útil.** La carga útil de un paquete puede ser dividida en dos campos:
 - *Campo de Voz* – Consta de datos de voz de longitud fija y existe en paquetes de alta calidad de voz y paquetes combinados de datos-voz. No es necesaria ninguna cabecera de carga útil.
 - *Campo de Datos* – Consta de tres partes, cabecera de carga útil, datos de carga útil, y código CRC.

2.3.5 CORRECCIÓN DE ERRORES

En una comunicación Bluetooth existen varios esquemas diferentes de corrección de errores:

- En la cabecera, cada bit es repetido tres veces.
- En la carga útil se usa un esquema de código Hamming. Los bits de información son agrupados en secuencias de 10 bits, éstos son enviados como 15 bits y el algoritmo corrige todos los errores de un bit y detecta los errores de dos bits.

- Para garantizar una recepción correcta, todos los paquetes de datos son retransmitidos hasta que el emisor reciba una confirmación. La misma que es enviada en la cabecera de los paquetes retornados.
- Los paquetes broadcast son paquetes transmitidos desde el maestro a todos los esclavos. No hay posibilidad de usar confirmación para esta comunicación, sin embargo, para incrementar la posibilidad de recibir correctamente un paquete, cada bit en el paquete es repetido un número fijo de veces.
- El chequeo de redundancia cíclica (CRC) se usa para detectar errores en la cabecera. La suma de comprobación CRC está contenida en el campo HEC de la cabecera de paquete. Los chequeos de redundancia cíclica también se aplican sobre la carga útil en la mayoría de los paquetes.
- Para asegurar que no desaparezcan paquetes completos, Bluetooth usa números de secuencia. Actualmente sólo se usa un número de secuencia de un bit.

2.3.6 CONTROL DE CANAL

El control de canal describe cómo se establece el canal de una Piconet y cómo las unidades pueden ser adicionadas o liberadas en la misma. La dirección del maestro determina la secuencia de saltos y el código de acceso al canal.

La fase de la Piconet está determinada por el sistema de reloj del maestro, por definición, la unidad Bluetooth que inicia la conexión representa al maestro.

En Bluetooth, la capa de control de enlace se divide en dos estados principales: standby y conexión. Además existen siete sub-estados: page, page scan, inquiry (búsqueda), inquiry scan, respuesta de maestro, respuesta de esclavo y respuesta a inquiry. Los sub-estados son usados para agregar nuevos esclavos a una

Piconet. Para moverse de un estado a otro se usan comandos de capas más altas o señales internas.

En Bluetooth se define un procedimiento de búsqueda que se usa en aplicaciones donde la dirección del dispositivo de destino es desconocida para la fuente. Esto puede ser usado para descubrir qué otras unidades Bluetooth están dentro del rango. Durante un sub-estado de búsqueda, la unidad de descubrimiento recoge la dirección del dispositivo y el reloj de todas las unidades que respondan al mensaje de búsqueda, entonces la unidad puede iniciar una conexión con alguna de las unidades descubiertas. El mensaje de búsqueda difundido por la fuente no contiene información de ella, sin embargo, puede indicar qué clase de dispositivos deberían responder. Una unidad que permita ser descubierta, regularmente entra en un sub-estado de inquiry scan para responder a los mensajes de búsqueda.

Existen dos formas de detectar otras unidades. La primera, detecta todas las otras unidades en el rango de cobertura, y la segunda, detecta un tipo específico de unidades.

Los esclavos que se encuentran en el sub-estado de page scan, escuchan esperando su propio código de acceso de dispositivo. El maestro en el sub-estado page activa y conecta a un esclavo. El maestro trata de capturar al esclavo transmitiendo repetidamente el código de acceso de dispositivo en diferentes canales de salto.

Debido a que los relojes del maestro y del esclavo no están sincronizados, el maestro no sabe exactamente cuándo y en qué frecuencia de salto se activará el esclavo.

Después de haber recibido su propio código de acceso de dispositivo, el esclavo transmite un mensaje de respuesta, este mensaje de respuesta es simplemente el código de acceso de dispositivo del esclavo. Cuando el maestro ha recibido este paquete, envía un paquete de control con información acerca de su reloj, dirección, clase de dispositivo, etc. El esclavo responde con un nuevo mensaje

donde envía su dirección. Si el maestro no obtiene esta respuesta en un determinado tiempo, él reenvía el paquete de control. Si el esclavo excede el tiempo de espera, entonces retorna al sub-estado de page scan, si es el maestro quien lo excede, entonces retorna al sub-estado de page e informa a las capas superiores.

Cuando se establece la conexión, la comunicación inicia con un paquete de sondeo desde el maestro hacia el esclavo. Como respuesta se envía un nuevo paquete de sondeo y de esta forma se verifica que la secuencia de salto y la sincronización sean correctas. La Figura 2.12, muestra la inicialización de la comunicación sobre el nivel banda base.

Cada transceiver (receptor-transmisor) Bluetooth tiene una única dirección de dispositivo de 48 bits asignada, la cual está dividida en tres campos: campo LAP, campo UAP y campo NAP.

Los campos LAP y UAP forman la parte significativa del código de acceso. En la Figura 2.13 se puede observar el formato de la dirección para un dispositivo Bluetooth. La dirección del dispositivo es conocida públicamente y puede ser obtenida a través de una rutina inquiry.

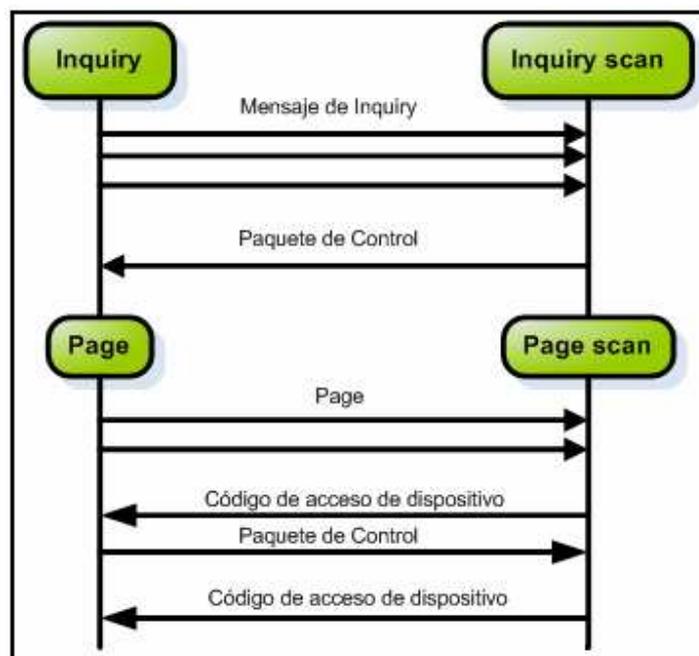


Figura 2.12. Iniciación de comunicación sobre el nivel banda base



Figura 2.13. Dirección de dispositivo Bluetooth

2.3.7 SEGURIDAD EN BLUETOOTH

Con el fin de brindar protección y confidencialidad a la información, el sistema debe ofrecer medidas de seguridad en las dos capas: aplicación y enlace. Todas las unidades Bluetooth tienen implementadas las mismas rutinas de autenticación y encriptación. En la capa de enlace, estas rutinas constan de cuatro entidades diferentes: una única dirección pública, dos llaves secretas y un número aleatorio el cual es diferente para cada transacción.

Solamente es encriptada la carga útil. El código de acceso y la cabecera de paquete nunca son encriptados. Cada tipo de unidad Bluetooth tiene una rutina de autenticación común. El maestro genera un número aleatorio y lo envía al esclavo; él usa este número y su propia identidad para calcular el número de autenticación; luego, es enviado al maestro, quien hace el mismo cálculo. Si los dos números generados son iguales, entonces la autenticación es compatible.

La encriptación, frecuentemente está restringida por leyes de varios países. Para evitar estas limitaciones, en Bluetooth la llave de encriptación no es estática, ésta es deducida de la llave de autenticación cada vez que se activa la encriptación

La seguridad de Bluetooth se basa en la autenticación del dispositivo, no en la autenticación del usuario. Los dispositivos de Bluetooth son identificados por los 48 bits usados como dirección MAC.

Bluetooth ofrece tres modos de seguridad que se presentan en la Tabla 2.1.

MODO	NOMBRE	DESCRIPCIÓN
1	No-seguro	No se pone en ejecución ninguna seguridad
2	Seguridad del Servicio-nivel	Tener acceso se concede a los servicios individuales
3	Seguridad Link-level	La seguridad se hace cumplir en un nivel común para todos los usos al principio de la conexión

Tabla 2.1. Modos de seguridad

Bluetooth ofrece tres niveles posibles de seguridad que se presentan en la Tabla 2.2.

MODO	DESCRIPCIÓN
3	No se requiere ninguna autenticación o autorización.
2	Se requiere la autenticación; la autorización no se requiere.
1	Se requiere la autorización y la autenticación.

Tabla 2.2. Niveles de seguridad

2.3.8 CLASES DE DISPOSITIVOS BLUETOOTH

El radio de cobertura de los dispositivos de Bluetooth depende de la potencia por lo que se les clasifica de la siguiente manera Tabla 2.3:

CLASE	ALCANCE (Aprox.)	POTENCIA DE SALIDA
1	100m	20dBm
2	10m	4dBm
3	10cm	0dBm

Tabla 2.3. Clase de los dispositivos Bluetooth

Se debe tomar en cuenta que las distancias que indican las especificaciones, son medidas tomando punto a punto dos dispositivos de la misma clase, instalados a campo abierto y sin ninguna interferencia.

La interfaz radio Bluetooth se basa en una antena de potencia nominal de 0dBm. Cada dispositivo puede variar su potencia de manera opcional, el equipamiento con control de potencia optimiza la potencia de salida con comandos procedentes del protocolo de enlace, esto se hace midiendo el RSSI (Receiver Signal Strength Indicator), retornando un mensaje indicando si la potencia debe ser incrementada o decrementada.

2.3.9 PILA DE PROTOCOLOS DE BLUETOOTH

En la figura 2.14 se muestra la pila completa de protocolos de Bluetooth, en la parte superior de la pila están las aplicaciones que soportan los modelos de uso de Bluetooth y en la parte inferior se encuentran las capas comunes, tal como la Banda Base. No todas las aplicaciones hacen uso de todos los protocolos que se muestran en la figura sino que funcionan dependiendo de las necesidades de la aplicación.

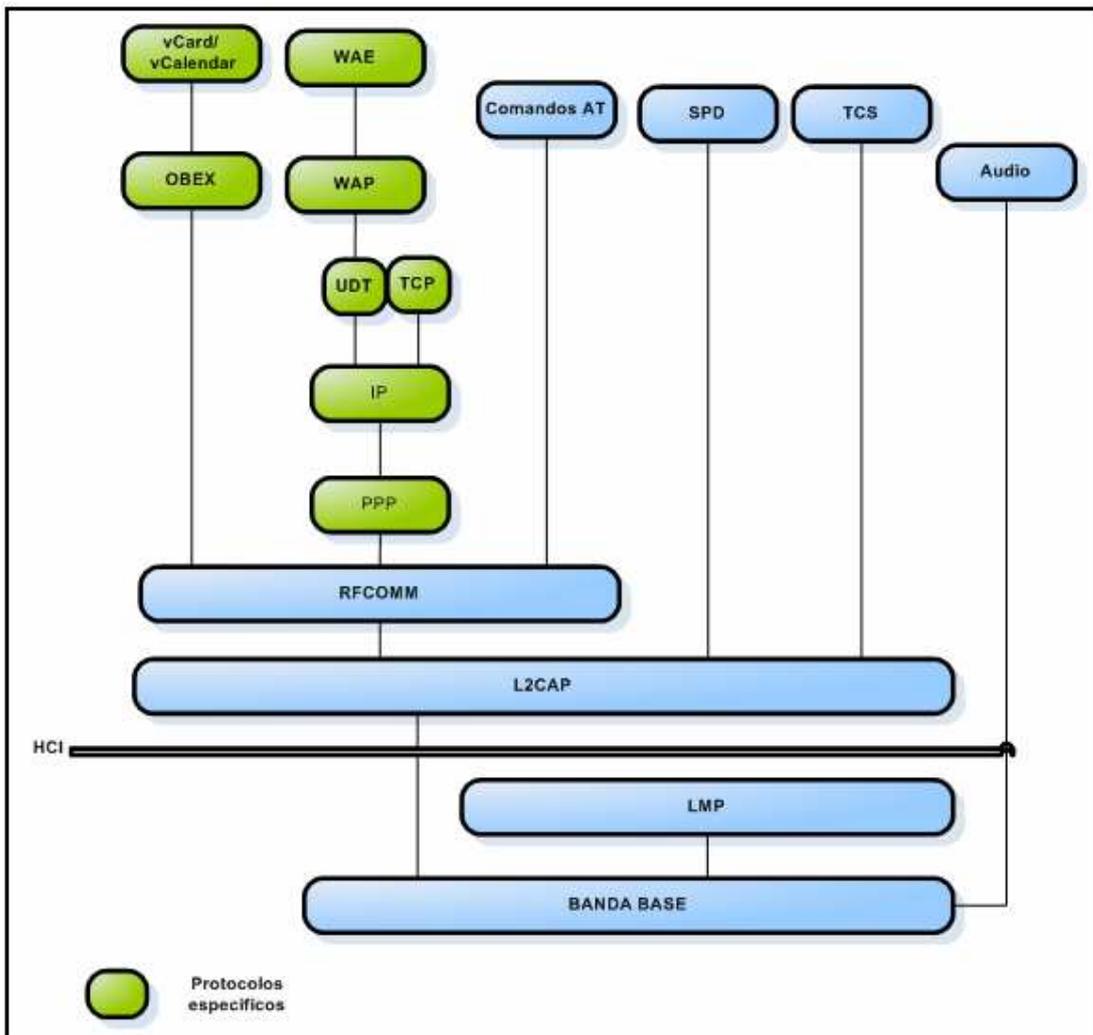


Figura 2.14. Stack de Protocolo Bluetooth

- Banda Base** es la capa de comunicación más baja que implementa el canal físico real y emplea una secuencia aleatoria de saltos a través de 79 frecuencias de radio diferentes. Los paquetes son enviados sobre el canal físico en una frecuencia de salto diferente. Controla la sincronización de las unidades Bluetooth y la secuencia de saltos en frecuencia, además es la responsable de la información para el control de enlace a bajo nivel como el reconocimiento, control de flujo y caracterización de carga útil y soporta dos tipos de enlace: síncrono orientado a la conexión (SCO), para datos y

asíncrono no orientado a la conexión (ACL), principalmente para audio. Los dos pueden ser multiplexados para usar el mismo enlace RF, usando un ancho de banda reservado, los enlaces SCO soportan tráfico de voz en tiempo real.

- **Link Manager Protocol (LMP)** o Protocolo de Gestión de Enlace es el responsable de la autenticación, encriptación, control y configuración del enlace. También se encarga del manejo de los modos y consumo de potencia, además soporta los procedimientos necesarios para establecer un enlace SCO.
- **Host Controller Interface (HCI)** o Interfaz del Controlador de Enlace brinda un método de interfaz uniforme para acceder a los recursos de hardware de Bluetooth. Éste contiene una interfaz de comando para el controlador Banda Base y la gestión de enlace e incluso para acceder al hardware.
- **Logical Link Control and Adaptation Protocol (L2CAP)** o Protocolo de Control y Adaptación de Enlace Lógico, corresponde a la capa de enlace de datos. Ésta brinda servicios de datos orientados y no orientados a la conexión a capas superiores. L2CAP multiplexa los protocolos de capas superiores con el fin de enviar varios protocolos sobre un canal Banda Base y manipular paquetes de capas superiores más grandes que el tamaño máximo del paquete Banda Base, L2CAP los segmenta en varios paquetes Banda Base. La capa L2CAP del receptor reensambla los paquetes Banda Base en paquetes más grandes para la capa superior. La conexión L2CAP permite el intercambio de información referente a la calidad de la conexión, inclusive maneja grupos de tal manera que varios dispositivos pueden comunicarse entre sí.
- **Service Discovery Protocol (SDP)** o Protocolo de Descubrimiento de Servicio define cómo actúa una aplicación de un cliente Bluetooth para

descubrir servicios disponibles de servidores Bluetooth, además de proporcionar un método para determinar las características de dichos servicios.

- **Protocolo RFCOMM** ofrece emulación de puertos seriales sobre el protocolo L2CAP. RFCOMM emula señales de control y datos RS-232 sobre la Banda Base Bluetooth. Éste ofrece capacidades de transporte a servicios de capas superiores (por ejemplo OBEX) que usan una línea serial como mecanismo de transporte.

RFCOMM soporta dos tipos de comunicación, directa entre dispositivos actuando como endpoints y dispositivo-modem-dispositivo, además tiene un esquema para emulación de null modem.

- **Control de telefonía binario (TCS binario)** es un protocolo que define la señalización de control de llamadas, para el establecimiento y liberación de una conversación o una llamada de datos entre unidades Bluetooth. Además, éste ofrece funcionalidad para intercambiar información de señalización no relacionada con el progreso de llamadas.
- **Audio** es una capa especial, usada sólo para enviar audio sobre Bluetooth. Las transmisiones de audio pueden ser ejecutadas entre una o más unidades usando muchos modelos diferentes. Los datos de audio no pasan a través de la capa L2CAP, pero sí directamente después de abrir un enlace y un establecimiento directo entre dos unidades Bluetooth.

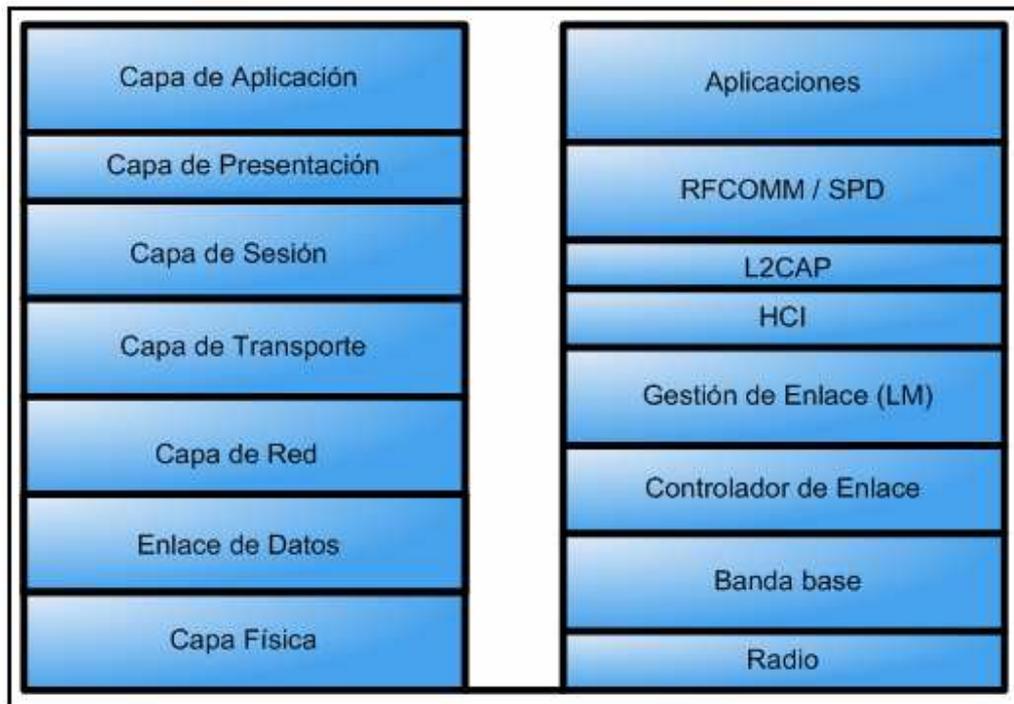
La pila completa de Bluetooth comprende protocolos específicos como:

- **Control de telefonía** Bluetooth soporta un número de comandos AT para el control de telefonía a través de la emulación de puerto serial (RFCOMM).

- **Protocolo Punto a Punto (PPP)** el PPP es un protocolo orientado a paquetes y por lo tanto debe usar su mecanismo serial para convertir una corriente de paquetes de datos en una corriente de datos seriales. Este protocolo corre sobre RFCOMM para lograr las conexiones punto a punto.
- **Protocolos UDP/TCP – IP** los estándares UDP/TCP e IP permiten a las unidades Bluetooth conectarse, por ejemplo a Internet, a través de otras unidades conectadas. Por lo tanto, la unidad puede actuar como un puente para Internet. La configuración TCP/IP/PPP está disponible como un transporte para WAP.
- **Wireless Application Protocol (WAP)** o Protocolo de Aplicación Inalámbrica, WAP es una especificación de un protocolo inalámbrico que trabaja con una amplia variedad de tecnologías de red inalámbricas conectando dispositivos móviles a Internet. Bluetooth puede ser usado como portador para ofrecer el transporte de datos entre el cliente WAP y su servidor de WAP adyacente. Además, las capacidades de red de Bluetooth dan a un cliente WAP posibilidades únicas en cuanto a lo que movilidad se refiere en comparación con otros portadores WAP.
- **Protocolo OBEX** el OBEX es un protocolo opcional de nivel de aplicación diseñado para permitir a las unidades Bluetooth soportar comunicación infrarroja para intercambiar una gran variedad de datos y comandos. Éste usa un modelo cliente-servidor y es independiente del mecanismo de transporte y del API (Application Program Interface) de transporte. OBEX usa RFCOMM como principal capa de transporte.

La Figura 2.15 muestra una comparación del stack Bluetooth con el modelo de referencia estándar Open Systems Interconnect, OSI, para stacks de protocolo de comunicaciones. A pesar de que Bluetooth no concuerda exactamente con el modelo, ésta comparación es muy útil para relacionar las diferentes partes del

stack Bluetooth con las capas del modelo OSI. Dado que el modelo de referencia es un stack ideal y bien particionado, la comparación sirve para resaltar la división de funciones en el stack Bluetooth.



a) Modelo de Referencia OSI

b) Bluetooth

Figura 2.15. Modelo de referencia OSI y Bluetooth

2.3.10 PERFILES DE BLUETOOTH

El estándar Bluetooth fue creado para ser usado por un gran número de fabricantes e implementado en áreas ilimitadas. Para asegurar que todos los dispositivos que usen Bluetooth sean compatibles entre sí son necesarios esquemas estándar de comunicación en las principales áreas.

Para evitar diferentes interpretaciones del estándar Bluetooth acerca de cómo un tipo específico de aplicación debería ser implementado, el Bluetooth Special Interest Group (SIG), ha definido modelos de usuario y perfiles de protocolo. Un perfil define una selección de mensajes y procedimientos de las especificaciones

Bluetooth y ofrece una descripción clara de la interfaz de aire para servicios específicos.

Un perfil puede ser descrito como una “rodaja” completa de la pila de protocolo. Existen cuatro perfiles generales definidos, en los cuales están basados directamente algunos de los modelos de usuario más importantes y sus perfiles.

Estos cuatro modelos son: Perfil Genérico de Acceso (GAP), Perfil de Puerto Serial, Perfil de Aplicación de Descubrimiento de Servicio (SDAP) y Perfil Genérico de Intercambio de Objetos (GOEP).

A continuación se hace una breve descripción de estos y algunos otros perfiles Bluetooth. La Figura 2.16 muestra el esquema de los perfiles Bluetooth. En ella se puede observar la jerarquía de los perfiles, como por ejemplo que todos los perfiles están contenidos en el Perfil Genérico de Acceso (GAP).

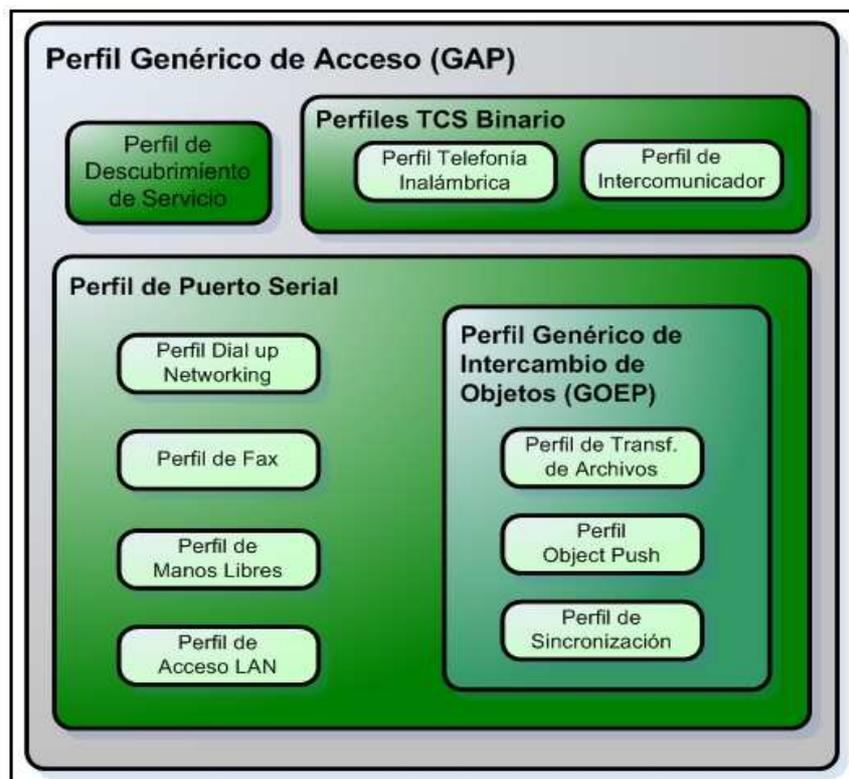


Figura 2.16. Perfiles de Bluetooth

- **Perfil Genérico de Acceso (GAP)** este perfil define los procedimientos generales para descubrir y establecer una conexión entre dispositivos Bluetooth. El GAP maneja el descubrimiento y establecimiento entre unidades que no están conectadas y asegura que cualquier par de unidades Bluetooth, cualquiera que sea su fabricante o aplicación, puedan intercambiar información a través de Bluetooth para descubrir qué tipo de aplicaciones soportan las unidades.
- **Perfil de Puerto Serial** define los requerimientos para dispositivos Bluetooth, necesarios para establecer una conexión de cable serial emulada usando RFCOMM entre dos dispositivos similares. Este perfil solamente requiere soporte para paquetes de un slot, esto significa que pueden ser usadas tasas de datos de hasta 128 Kb/s. El soporte para tasas más altas es opcional. RFCOMM es usado para transportar los datos de usuario, señales de control de modem y comandos de configuración. El perfil de puerto serial es dependiente del GAP.
- **Perfil de Aplicación de Descubrimiento de Servicio (SDAP)** define los protocolos y procedimientos para una aplicación en un dispositivo Bluetooth donde se desea descubrir y recuperar información relacionada con servicios localizados en otros dispositivos. El SDAP es dependiente del GAP.
- **Perfil Genérico de Intercambio de Objetos (GOEP)** define protocolos y procedimientos usados por aplicaciones para ofrecer características de intercambio de objetos. Los usos pueden ser, por ejemplo: sincronización, transferencia de archivos o modelo Object Push.
Los dispositivos más comunes que usan este modelo son: agendas electrónicas, PDAs, teléfonos celulares y teléfonos móviles. El GOEP es dependiente del perfil de puerto serial.

- **Perfil de Telefonía Inalámbrica** este perfil define cómo un teléfono móvil puede ser usado para acceder a un servicio de telefonía de red fija a través de una estación base. Es usado para telefonía inalámbrica de hogares u oficinas pequeñas. El perfil incluye llamadas a través de una estación base, haciendo llamadas de intercomunicación directa entre dos terminales y accediendo adicionalmente a redes externas. Es usado por dispositivos que implementan el llamado “teléfono 3 en 1”.
- **Perfil de Intercomunicador** define los usos de teléfonos móviles, los cuales establecen enlaces de conversación directa entre dos dispositivos. El enlace directo es establecido usando señalización de telefonía sobre Bluetooth. Los teléfonos móviles que usan enlaces directos funcionan como walkie-talkies.
- **Perfil de Manos Libres** este perfil define los requerimientos, para dispositivos Bluetooth, necesarios para soportar el uso de manos libres. En este caso el dispositivo puede ser usado como unidad de audio inalámbrico de entrada/salida. El perfil soporta comunicación segura y no segura.
- **Perfil Dial-up Networking** define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso del modelo llamado Puente Internet. Este perfil es aplicado cuando un teléfono celular o modem es usado como un modem inalámbrico.
- **Perfil de Fax** este perfil define los protocolos y procedimientos que deben ser usados por dispositivos que implementen el uso de fax. En el perfil un teléfono celular puede ser usado como un fax inalámbrico.
- **Perfil de Acceso LAN** este perfil define el acceso a una red de área local, LAN, usando el protocolo punto-a-punto, PPP, sobre RFCOMM. PPP es ampliamente usado para lograr acceder a redes soportando varios

protocolos de red. El perfil soporta acceso LAN para un dispositivo Bluetooth sencillo, acceso LAN para varios dispositivos Bluetooth y PC-a-PC (usando interconexión PPP con emulación de cable serial).

- **Perfil Object Push** define protocolos y procedimientos usados en el modelo object push. Este perfil usa el GOEP. En el modelo object push hay procedimientos para introducir en el inbox, sacar e intercambiar objetos con otro dispositivo Bluetooth.
- **Perfil de Transferencia de Archivos** este perfil define protocolos y procedimientos usados en el modelo de transferencia de archivos. El perfil usa el GOEP. En el modelo de transferencia de archivos hay procedimientos para chequear un grupo de objetos de otro dispositivo Bluetooth, transferir objetos entre dos dispositivos y manipular objetos de otro dispositivo. Los objetos podrían ser archivos o folders de un grupo de objetos tal como un sistema de archivos.
- **Perfil de Sincronización** este perfil define protocolos y procedimientos usados en el modelo de sincronización. Éste usa el GOEP. El modelo soporta intercambios de información, por ejemplo para sincronizar calendarios de diferentes dispositivos.

2.3.11. DISPOSITIVO KCWIREFREE KC -11³

El módulo a utilizar para la comunicación Bluetooth entre la pasarela y el PDA del Sistema Domótico es el KC-11.

El módulo de mayor alcance de Bluetooth disponible, es el módulo amplificado OEM de KC-11 Bluetooth que se diseñó para tener una potencia de salida de señal máxima de 20 dBm aproximadamente. Su gran alcance se basa en el

³ Datasheet KCWIREFREE KC11

amplificador integrado en el chip, el cual eleva las señales de la transmisión y las de recepción para lograr exceder las especificaciones mínimas para un dispositivo Bluetooth clasificado en clase 1.

El KC-11 es un módulo de montaje superficial que proporciona un tamaño pequeño (15mm x 43mm), el mismo que contiene una memoria flash reprogramable, proporciona soporte para el reemplazo del cable serial usando el perfil de Bluetooth SPP. Los perfiles más populares de Bluetooth están también disponibles.

Se puede modificar rápidamente las características del módulo para los requisitos particulares que se necesiten en cada aplicación, simplemente con comandos AT, para las optimizaciones tales como mínimo consumo de energía, respuesta de alta velocidad, características de audio, y los valores iniciales por defecto, entre otros.

Los módulos vienen previamente configurados en sus características óptimas, los cuales se cargan fácilmente en cualquier dispositivo de transmisión serial y están listos para usar sin procedimientos adicionales.

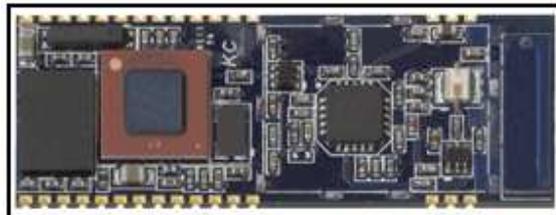


Figura 2.17. Dispositivo KC – 11

2.3.11.1 Características generales del módulo KC-11

Entre las características más generales del módulo KC-11 se tiene las siguientes:

- Cumple las normas de la FCC.
- RF configurado para Bluetooth.
- Perfil del puerto serial (SPP).
- Radio de clase 1.

- Bluetooth v1.2.
- Comunicación inalámbrica.
- Microprocesador ARM7 hasta 48MHz.
- Memoria flash de 8Mbytes.
- Velocidad de señal máxima de 921 K baudios
- Antena integrada con ganancia de 16 dBm.
- Seguridad del cifrado de 128 bits.
- Transmisión y recepción amplificada.
- Rango de 100m LoS.
- Interfaz de SPI, hasta 24MHz.
- 14 pines de entrada-salida de propósito general.
- Configurable con comandos AT.
- Capacidad multi – punto.

2.3.11.2. Arquitectura de Software

El módulo KC – 11 cumple con la siguiente pila de protocolos Bluetooth para la interacción de mensajes (comandos y eventos), entre el microprocesador ARM7 y la memoria flash.

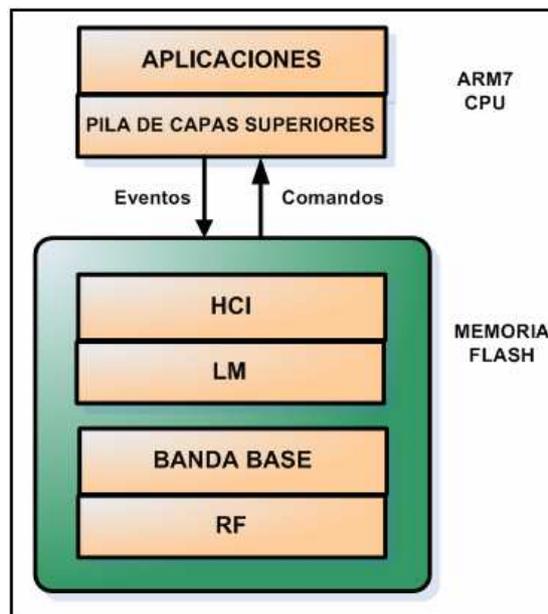


Figura 2.18. Pila de capas

2.4.11.2.1. Pila de capas inferiores

- Velocidad de transmisión máxima de 723.3kbps.
- Autenticación y encriptación.
- Encriptación de 8-bits a 128-bits como máximo.
- Memoria Flash para el almacenamiento de los parámetros de radio.
- En el enlace asíncrono no orientado a la conexión (ACL) actúa con tipos de paquetes DM1, DH1, DM3, DH3, DM5, DH5, AUX1.
- En el enlace síncrono orientado a la conexión (SCO) actúa con tipos de paquetes HV1, HV2, HV3.
- Para punto – multipunto y soporte scatternet el módulo puede interactuar con 3 másteres y 7 esclavos (máximo 10 enlaces activos).
- Co- existe con 802.11b, 802.11g, AWMA y AFH.
- SCO sobre UART, PCM o interface SPI.

2.3.11.2.2. Pila de capas superiores

- Funciona con los protocolos SPP, SDAP, GAP, y DUN.
- Soporte para RFCOMM, SDP y L2CAP.

2.3.11.2.3. Interface HCI

- Cumple con las especificaciones Bluetooth 1.2.
- HCI con USB en la capa de transporte (H2).
- HCI con UART en la capa de transporte (H4).
- Actualización de software a través de UART.

2.4. DESCRIPCIÓN DE UN PDA

PDA, son las iniciales en Inglés de Personal Digital Assistant, cuya traducción al español es Asistente Personal Digital.

En la actualidad son potentes computadores personales de bolsillo. Aunque originalmente se iniciaron como agendas electrónicas, hoy en día son capaces de realizar la mayoría de tareas que realiza un computador convencional: agenda electrónica, diccionario, conversor de divisas y medidas, calculadora, acceso a Internet, reproductor de MP3, grabadora de sonidos (que incluso permite recordatorios sonoros), y además leer libros electrónicos, etc.

Las PDAs funcionan con dos sistemas operativos muy usuales: el Palm Os, que ya va por su versión 5, y el Windows de Microsoft, que en la actualidad presenta la Versión Pocket PC Windows Mobile Versión 6.0.

Salvando las lógicas diferencias de rendimiento, un PDA es parecido a una PC ya que en la tarjeta madre posee un procesador, memoria RAM, memoria permanente (ROM). También hay un sistema operativo en el cual se ejecutan los programas que se requieran (procesadores de texto, hojas de cálculo, navegadores web, juegos, agendas, etc.). La forma de comunicarse con el PDA es la propia pantalla que en todos los modelos actuales es táctil. El lápiz que incorpora el aparato se utiliza a modo de ratón. Para escribir se puede utilizar un teclado virtual que se muestra en la pantalla o bien hacer uso de la característica de reconocimiento de escritura. En esta última existen dos sistemas: trazado aprendido (cada carácter tiene asociado un movimiento del lápiz) y trazado natural (el PDA aprende nuestra forma de escribir).

En resumen, los PDAs son computadores que tienen las siguientes características:

- *Tamaño:* caben en la palma de la mano y se pueden llevar en un bolsillo.
- *Touchscreen:* es una pantalla táctil en la que se puede escribir con un lápiz que incluye un punzón de plástico, siendo capaz de reconocer letras y números manuscritos. La salida se realiza a través de la misma pantalla.

- *Interconectividad:* los PDAs se conectan con otros PDAs, teléfonos móviles, etc., mediante una interconexión de infrarrojos, y con una PC mediante un soporte que se conecta a través de un puerto serie o USB.
- *Alimentación:* mediante pilas o baterías.

Hay muchas personas que asocian PDA con agenda electrónica, pero como se indicó en el párrafo anterior un PDA es mucho más que una agenda electrónica. Las últimas versiones son auténticos computadores de bolsillo.

2.4.1. Componentes de un PDA

- **Procesador**

Dos son los fabricantes que incorporan su tecnología a los procesadores de un PDA: Intel que acompaña a las PDA_S Pocket PC de Microsoft y Texas Instruments para los procesadores incorporados en las PDA_S de PALM OS, también están los ARM y los Hitachi.

Las velocidades mínimas, hoy en día ya obsoletas, eran de 16MHz, en la actualidad la velocidad máxima alcanza los 400MHz. Obviamente, dependiendo de las funciones que se requiera, se deberá elegir la que esté más acorde a nuestras necesidades.

- **Memoria**

La memoria estándar que utilizan los dispositivos Pocket PC es de 64 Mbytes, aunque se puede añadir más memoria con tarjetas de expansión.

Los modelos Palm más bajos suelen tener menor cantidad de RAM, usualmente 16 Mb, que sería la memoria mínima que se debería exigir.

El requerimiento de mayor cantidad de memoria es siempre deseable teniendo en cuenta las posibilidades de entorno multimedia, los archivos MP3 y los videos. Aún siendo el modelo menos vendido, La Zire 31 brinda

la posibilidad de escuchar archivos MP3 sin ningún tipo de dificultad y dispone de tarjeta de expansión si el cliente quiere.

La memoria en estos ordenadores de bolsillo almacena los programas que se están ejecutando, pero también los documentos del usuario (agenda, documentos de Word, canciones MP3, etc.).

La mayoría de los modelos poseen una tarjeta interna de memoria y espacio para tarjetas adicionales. Dichas tarjetas son de tipo Compact Flash, SmartMedia, Multimedia Card, es decir, las mismas que emplean los reproductores MP3 o las cámaras digitales. La cantidad de memoria instalada determinará el número de aplicaciones que se puede tener instaladas y la cantidad de archivos de datos que se pueden almacenar, en la actualidad existen en el mercado tarjetas de varias gigas de capacidad.

- **Puertos**

Desaparecido ya el puerto serial 232 (que adolecía de gran lentitud) las formas habituales de conexión son el puerto USB y el puerto de infrarrojos. Dos PDAs que utilicen el mismo sistema operativo pueden intercambiar datos de esta última forma con gran sencillez, lo cual amplía enormemente las posibilidades de estos pequeños computadores.

- **Comunicaciones**

Se puede conectar el PDA con un computador para intercambiar datos (por ejemplo, para mantener sincronizada una agenda, citas, tareas, para transferir documentos, música, etc.), esto se logra mediante un cable USB o por infrarrojos.

Opcionalmente, se logra conectar el PDA a una red inalámbrica (Wi – Fi o Bluetooth), o a una línea de teléfono (con un módem).

- **Bluetooth**

Bluetooth se ha convertido en la principal tecnología inalámbrica que actualmente incorpora cualquier dispositivo que se evalúe. Es indiscutible que continuamente se lanzan al mercado asistentes digitales con conexión mediante infrarrojos, de cualquier modo, ya existen dispositivos muy asequibles con tecnología Bluetooth, que se perfila como la tecnología de conectividad del futuro, por ejemplo con la última generación de teléfonos móviles.

- **Acceso a Internet**

Durante mucho tiempo el acceso a Internet se ha realizado exclusivamente a través de un PC fijo y se han sucedido algunas iniciativas de realizarlo a través de los móviles, siendo el más actual los módulos GRPS, pero a precios absurdos y con dificultades de navegación.

Los PDAs aparecen como una solución, pues a través de la interfaz con el PC es posible enviar y recibir e-mails, descargar e-books, periódicos, y con esta información trabajar y entretenerse todo el día.

- **Ranuras de expansión**

Para los usuarios que demandan mayores capacidades en movilidad es importante que el PDA disponga de una ranura de expansión para tarjetas Secure Digital, Compact Flash ó Multimedia Card, entre otras que existen en el mercado. Con ellas se solucionan posibles limitaciones iniciales de los dispositivos que pueden ser ampliadas paulatinamente.

2.5.1.1. PDA - PALM Versus PDA - POCKET PC

El PDA - PALM fue el pionero hasta el punto que los PDAs se conocían bajo el nombre genérico PALM, siendo en principio unas potentes agendas personales que cada vez iban incorporando más aplicaciones, y gracias a ello todavía hoy se mantiene como líder del mercado. Algunos gigantes como Sony incorporaron a

sus PDAs un sistema operativo, aunque hoy en día Microsoft ha abandonado el Windows CE cambiándolo por el PC Pocket.

Por lo general, las dos compañías ofrecen aplicaciones similares entre los diferentes productos que componen su gama de dispositivos, pero utilizan software diferente. Las últimas versiones de PDAs utilizan software de Microsoft, mientras que las PDAs con menores prestaciones tienen un software PALM que optimiza mucho mejor los recursos de RAM.

2.5.2. POCKET PC HP IPAQ H6315⁴

La Pocket PC utilizada para la aplicación del Sistema Domótico es la IPAQ H6315 ver Figura 2.19

⁴ Datasheet HP IPAQ H6315



Figura 2.19. IPAQ 6315

La hp IPAQ h6315 es un PDA teléfono que ofrece conectividad y funcionalidad total. Es una de las más completas del mercado al incorporar en un solo aparato: teléfono con banda libre para activar en cualquier telefonía móvil, Wi – Fi para poder conectarse a Internet en cualquier lugar donde se disponga del servicio, cámara de fotos, bluetooth. Además consta de un completo paquete de software para gestionar documentos, conectarse a Internet e incluso ver películas y escuchar música.

El PDA h6315 es un dispositivo especialmente diseñado para estar conectado en cualquier momento en cualquier lugar donde exista señal de telefonía móvil, porque combina las capacidades de un teléfono móvil con las funciones de una Pocket PC y conectividad GSM/GPRS, Wi - Fi y Bluetooth.

El h6315 integra un sistema de conexión inteligente que detecta las redes disponibles en cada momento, de esta forma, si el dispositivo detecta una red WLAN no se conectará a través de GPRS. Por el contrario, si no detecta ninguna red Wi - Fi, el dispositivo se conectará a la red GPRS configurada.

2.5.2.1. Especificaciones técnicas de la IPAQ h6315

- Teléfono cuatribanda GSM/GPRS integrado.
- Wi – Fi LAN 802.11b.
- Bluetooth® 1.1.
- Irda.
- Sistema operativo Microsoft® windows® mobile 2003 (phone edition).
- Versiones pocket de Microsoft software incluidas (Outlook, Word, Excel e Internet Explorer para Pocket PC).
- Procesador Texas Instruments™ OMAP 1510.
- Display de 3.5" (89 mm) transfective tft color con 65.536 colores (16bits).
- Cámara VGA, resolución 640 x 480.
- Memoria de 64mb SDRAM, 64mb flash rom, hasta 20mb archive de almacenamiento IPAQ.
- Dimensiones 4.68" x 2.95" x .73" (119 mm x 75 mm x 18.7 mm).
- Ranuras de expansión SD slot integrado (soporta SD/MMC y SDIO).
- Audio micrófono y altavoz integrado.
- Conexión para auricular estéreo de 3.5 mm.
- Mp3 estéreo.

2.6. DESCRIPCIÓN DEL MICROCONTROLADOR ATMEGA 64

Es un microcontrolador de 8 bits de memoria flash programable de 64K bytes, el ATmega64 (ver Figura 2.20), es un microcontrolador de baja potencia Cmos basado en la arquitectura AVR. Ejecutando instrucciones de gran alcance en un solo ciclo de reloj, el ATmega64 alcanza velocidades de procesamiento que se acercan a 1 Mbps por megaciclo permitiendo que el diseñador del sistema optimice el consumo de energía en base del proceso de velocidad.

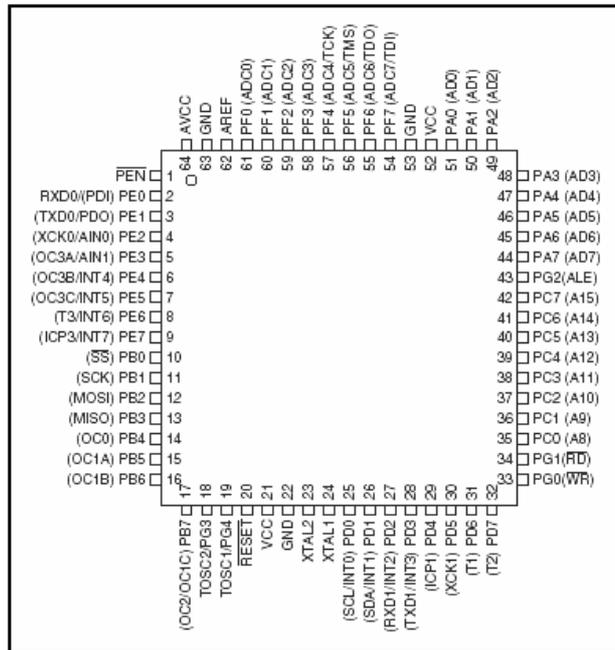


FIGURA 2.20. ATMEGA64

2.6.1. ARQUITECTURA AVANZADA DEL RISC

- Microcontrolador AVR de 8 bits con alto rendimiento y baja potencia.
- Rango de temperatura: -40°C a +85°C.
- Chip de 64 pines.
- 130 instrucciones de gran alcance - la mayoría de una sola ejecución con un ciclo de reloj.
- 32 x 8 registros de propósito general + registros de control periféricos.
- Operación completamente estática.
- Hasta 16 Mbps de velocidad de procesamiento en 16 MHz
- Duplicador interno de ciclo de máquina.
- 53 líneas programables de la entrada-salida.

2.6.2. PROGRAMA Y DATOS DE MEMORIA

- 64Kbytes de memoria flash reprogramable. Se puede usar 10.000 ciclos de escritura y borrado.
- Lectura de instrucciones mientras se encuentra en el proceso de escritura.

- Sección opcional de arranque con bits de bloqueo.
- 2Kbytes de memoria EEPROM (100.000 ciclos de escritura y borrado).
- 4Kbytes de SRAM interno.
- Memoria externa opcional de hasta 64Kbytes.
- Seguridad de software.
- Programación de interfaz SPI.

2.6.3. CARACTERÍSTICAS PERIFÉRICAS

- Dos temporizadores/contadores independientes de 8 bits con modo de comparación.
- Dos temporizadores/contadores extendidos de 16 bits con modo de comparación y modo de captura.
- Real Time con oscilador independiente.
- Dos canales PWM de 8 bits.
- Seis canales PWM con resolución programable que va de 1 a 16 bits.
- Ocho canales con ADC de 10 bits.
- 8 canales comunes.
- 7 canales diferenciales.
- 2 canales diferenciales con ganancia programable (1x, 10x, o 200x).
- Dos puertos de seriales UART.
- Interfaz SPI maestro/esclavo.
- Watchdog Timer programable con el oscilador del chip.
- Comparador Análogo interno.

2.6.4. CARACTERÍSTICAS ESPECIALES DEL MICROCONTROLADOR

- Permite resetear el sistema en caliente y la detección Brown-out.
- Oscilador RC con calibración interna.
- Interrupciones internas y externas.
- Seis modos de sleep: Idle, reducción del nivel de ruidos del ADC, ahorro de energía, Standby, Power-down, y Standby extendido.
- Frecuencia de reloj seleccionable por software.

- Modo de compatibilidad con el ATmega103 por fusibles.
- Deshabilitación global de Pull-up.

CAPÍTULO 3

DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DOMÓTICO

3.1. ESQUEMA GLOBAL DEL SISTEMA DOMÓTICO

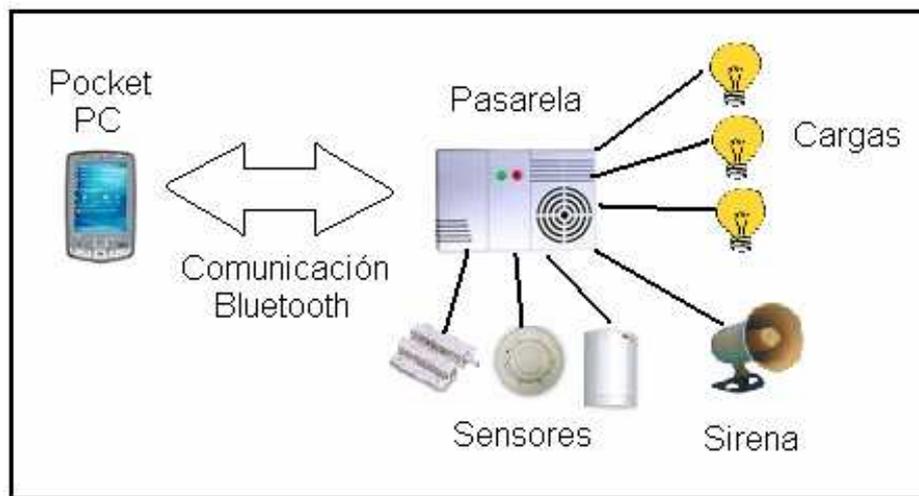


Figura 3.1. Esquema global del sistema

El sistema global se compone de una interfaz gráfica realizada en Embedded Visual Basic 3.0, programa que será instalado en la Pocket PC. Los comandos generados desde este programa serán enviados en forma inalámbrica, por medio de Bluetooth, hacia la pasarela Domótica. La pasarela realiza el control de todas las cargas y supervisa el estado de cada una de ellas. También efectúa las funciones de una alarma. La conexión entre la pasarela y las diferentes cargas y sensores que se van a controlar, está comandada por medio de cables.

3.2. CONDICIONES DE DISEÑO DEL SISTEMA DOMÓTICO

Las condiciones para el diseño del sistema domótico son las siguientes:

- Deberá ser un sistema comprendido entre un dispositivo móvil y un fijo los cuales deben comunicarse inalámbricamente a una distancia mínima de diez metros por medio de Bluetooth. El sistema debe poder realizar un control, mínimo para diez cargas eléctricas ON/OFF cuyas salidas deben soportar una corriente de diez amperios cada una de ellas, además se debe poder programar el horario de encendido y apagado de dichas cargas y también debe permitir el monitoreo de cuatro sensores normalmente cerrados (NC) y un sensor normalmente abierto (NA), el sistema de seguridad debe poder ser habilitado por el usuario y debe constar de una sirena.
- El dispositivo móvil tiene que ser un PDA, el cual debe poderse comunicar mediante Bluetooth con un radio de cobertura superior a los diez metros, debe tener un perfil de puerto serial SPP, y además que se tenga la posibilidad de instalar el Run Time Visual Basic.
- Para el control de la pasarela se deberá utilizar un microcontrolador AVR de tecnología superficial, el cual deberá tener un puerto de comunicación serial, debe disponer de un Reloj en Tiempo Real (Real Time), el mismo que proporcione la hora y la fecha actual, y el número de pines debe cubrir con todas las entradas y salidas para el control de las cargas eléctricas y del sistema de seguridad.
- Para la comunicación inalámbrica entre la pasarela domótica y el PDA se deberá utilizar un dispositivo Bluetooth el mismo que debe tener un radio de cobertura mayor a los diez metros, además debe tener perfil de puerto serial SPP, y tiene que ser barato.
- El sistema deberá tener una interfaz gráfica programada para el PDA el mismo que debe cumplir con las siguientes especificaciones:

- Una carátula en la cual consta el nombre de la aplicación y los autores.
- Una pantalla en la cual se ingrese una clave de cuatro dígitos los mismos que deben ser solo números, el usuario debe tener la posibilidad de cuatro intentos para ingresar la clave correcta caso contrario saldrá de la aplicación.
- Por medio de la aplicación se debe poder controlar las cargas eléctricas y el sistema de seguridad de la siguiente manera:
 - El usuario debe tener la posibilidad de habilitar y deshabilitar la comunicación inalámbrica entre el PDA y la pasarela domótica.
 - Se debe tener la posibilidad de controlar las cargas eléctricas desde el PDA, además de poder visualizar el estado de cada una de ellas (encendida o apagada).
 - En una nueva pantalla se deberá elegir una de las diez cargas eléctricas para observar y modificar su horario de encendido y apagado. Se debe poder programar las diez cargas eléctricas.
 - En otra pantalla se debe poder visualizar el estado de los cinco sensores los cuales deben estar siempre sensando. Además debe haber un botón para la habilitación y deshabilitación del sistema de seguridad.
- El sistema constará de una Pasarela Domótica, la cual permitirá realizar la comunicación, por medio de Bluetooth, entre los circuitos de control y el PDA.
- Para el sistema de seguridad se deberá contar con una sirena y los siguientes tipos de sensores:

- Sensor de movimiento.
- Sensor de humo.
- Sensor magnético.

3.3. ELEMENTOS DEL CIRCUITO ELECTRÓNICO DE LA PASARELA DOMÓTICA

El diagrama de bloques del módulo de la pasarela encargado de la comunicación entre el PDA y las diferentes cargas (iluminación, sensores, etc.) consta de:

- Fuente de 3.3 voltios de corriente continua.
- Fuente de 5 voltios de corriente continua.
- Fuente de 12 voltios de corriente continua.
- Un microcontrolador Atmega64.
- Un dispositivo Bluetooth KC-11.
- Opto-acopladores.
- Cargas eléctricas (focos,...).
- Sensores y sirena.

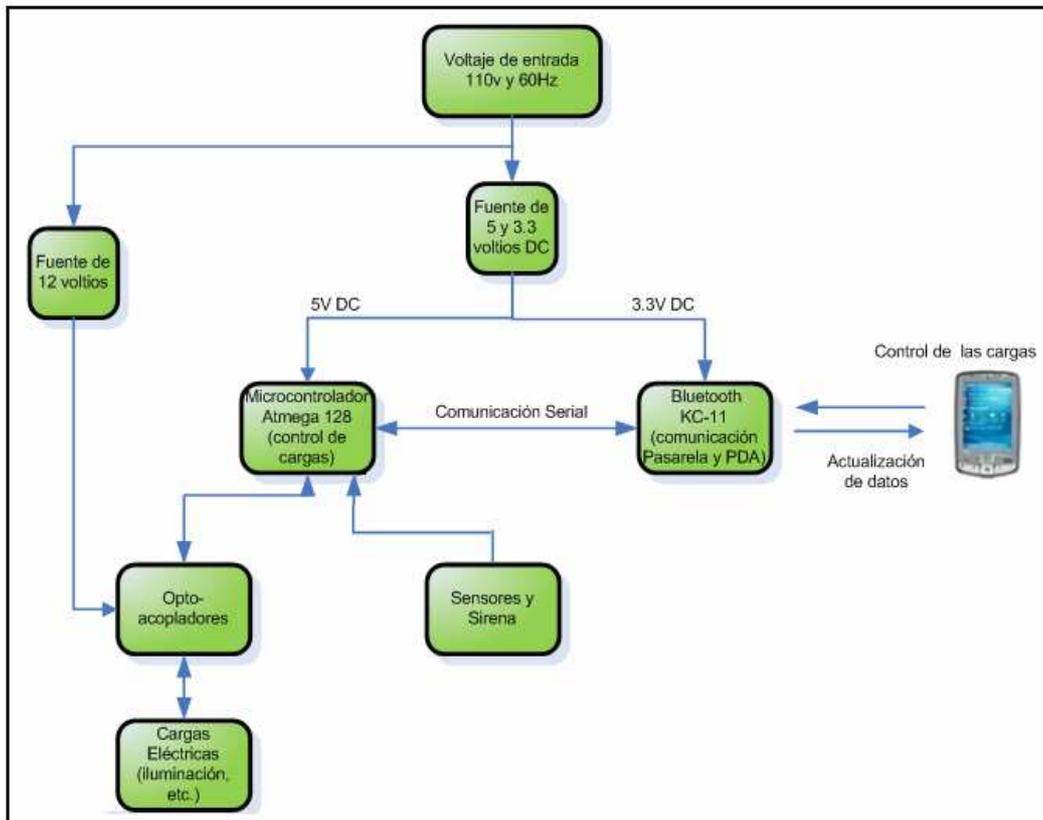


Figura 3.2. Diagrama-esquema del módulo de Pasarela Domótica

A continuación se detalla cada uno de los elementos de la figura 3.2:

Una vez que el voltaje de la red (110v, 60Hz) ingresa a la pasarela, pasa a dos fuentes fijas independientes. La primera proporciona 5V y 3.3V DC, mientras que la segunda proporciona 12V DC. La fuente de 3.3 voltios es utilizada solo para la alimentación del dispositivo Bluetooth KC-11, la de 5 voltios alimenta al microcontrolador y la de 12 voltios es para la alimentación de los opto-acopladores.

El dispositivo Bluetooth KC-11 es el encargado de realizar la comunicación inalámbrica bidireccional entre el microcontrolador y el PDA. Esta comunicación la realiza mediante comunicación serial.

El microcontrolador Atmega64, controla el sistema (controla las diferentes cargas y monitoriza los sensores de seguridad). El microcontrolador, al comunicarse con el PDA actualiza su información y a su vez los estados de las diferentes cargas.

El microcontrolador se encarga de las funciones de encendido y apagado de cualquier carga ya sea al momento en que el usuario esté utilizando el PDA o cuando dichas acciones hayan sido programadas.

Además, cuando verifique el estado de los diferentes sensores, en caso de existir algún inconveniente, este enviará una señal en la que se activará una sirena, con la cual se alertará sobre algún suceso extraño.

A continuación se presentan cada uno de los dispositivos que constituyen el circuito.

3.3.1. REGULADOR DE VOLTAJE DE 5 V DC

Se encarga de fijar a su salida un voltaje de 5 voltios DC. Éste alimentará a todos los dispositivos que requieran de este voltaje.

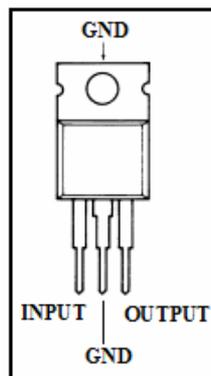


Figura 3.3. Regulador de voltaje 7805

Entre las principales especificaciones técnicas se tiene:

7805

$V_{in} = 7V \text{ a } 20V$, $T_j = 25 \text{ }^\circ\text{C}$

$0.5m \text{ A} \leq I_{out} \leq 1.0 \text{ A}$, $P_{out} \leq 15W$
 $4.8V \leq V_{in} \leq 5.2V$

3.3.2. REGULADOR DE VOLTAJE DE 3.3 V DC

Se encarga de fijar a su salida un voltaje de 3.3 voltios DC. Entre las especificaciones más importantes se tiene:

LM1117-3.3
$I_{out} = 100\text{mA}$, $V_{in} = 5\text{V}$, $T_j = 25\text{ }^\circ\text{C}$
$0 \leq I_{out} \leq 800\text{mA}$, $4.75\text{V} \leq V_{in} \leq 10\text{V}$

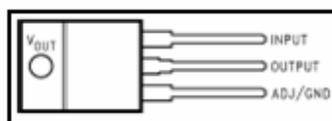


Figura 3.4. Regulador de voltaje LM1117-3.3

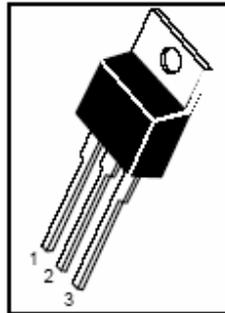
3.3.3. REGULADOR DE VOLTAJE DE 12V DC

Este circuito integrado es el encargado de fijar a su salida un voltaje fijo de 12 voltios de corriente continua.

Las características eléctricas del regulador de voltaje 7812 son las siguientes:

Condiciones generales ($V_{in} = 19\text{V}$, $I_o = 500\text{mA}$).

7812
$V_{in} = 14.5\text{V a } 27\text{V}$, $T_j = 25\text{ }^\circ\text{C}$
$0.5\text{mA} \leq I_{out} \leq 1.0\text{A}$, $P_{out} \leq 15\text{W}$
$11.4\text{V} \leq V_{in} \leq 12.6\text{V}$



- Pin: 1. Input.
2. Ground.
3. Output.

Figura 3.5. Regulador de voltaje 7812

3.3.4. OPTO - ACOPLADOR PC817

El opto-acoplador PC817 es un dispositivo de acoplamiento óptico. También se denominan opto-aisladores de acoplamiento óptico. Basan su funcionamiento en el empleo de un haz de radiación luminosa para pasar señales de un circuito a otro sin conexión eléctrica. La distribución de pines es la que se muestra en la Figura 3.6.

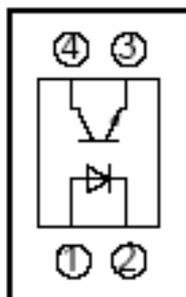


Figura 3.6. Opto-acoplador PC817

Las características eléctricas del opto-transistor PC817 son las siguientes:

PC817	
Input:	$I_F = 50\text{m A}$, $I_{FM} = 1.0\text{ A}$, $V_R = 6\text{ V}$, $P = 70\text{m W}$.
Output:	$V_{C-E} = 35\text{ V}$, $V_{EC} = 6\text{ V}$, $I_C = 50\text{m A}$, $P_{\text{disiCol}} = 150\text{m W}$, $P_{\text{TotalDisi}} = 200\text{m W}$

3.4. HARDWARE DE CONTROL

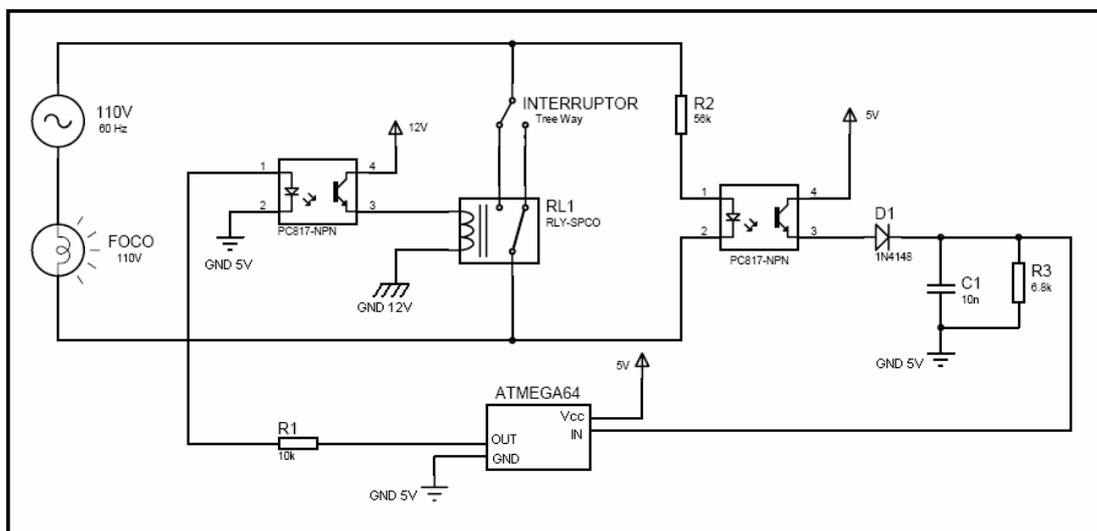


Figura 3.7. Control y detector del estado de una carga eléctrica (foco)

La función de encendido o apagado de un foco se puede efectuar de dos maneras diferentes: la primera es mediante de un conmutador, el cual está instalado en algún sitio de la habitación en el cual se desea controlar dicho foco. El segundo es mediante la Pocket PC (desde diferentes lugares de la casa). Cabe aclarar que dicha acción se realiza por medio del microcontrolador Atmega64 (Figura 3.7).

Para lo cual el microcontrolador debe conocer el estado actual en el que se encuentra el foco (ON/OFF), para esto se sensa el voltaje que cae entre los puntos **a** y **b** de la Figura 3.8.

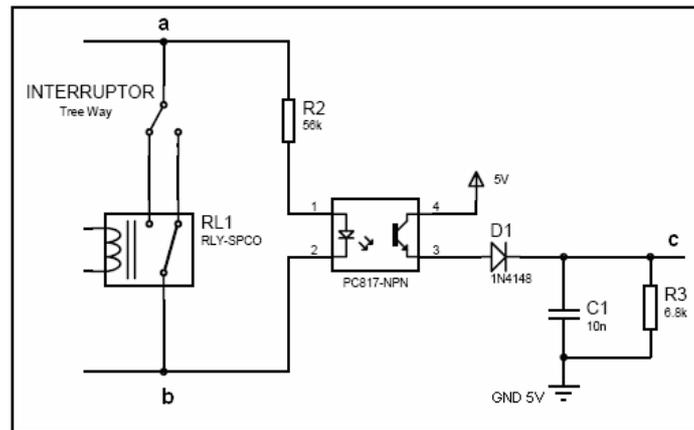


Figura 3.8. Detección del estado actual del foco.

Si el foco está apagado el voltaje que cae entre estos puntos es de aproximadamente 110 voltios, esto hará que el opto-transistor entre en saturación cuando pasa el semi ciclo positivo y en corte cuando pasa el semi ciclo negativo con lo que a la salida no se tiene un voltaje de corriente continua, para lo cual se realiza una rectificación de media onda lo que hace que en el punto **c** se obtenga un voltaje de 5V DC más un voltaje de rizado despreciable. Esta señal ingresa directamente al microcontrolador el cual, al leer un “1 Lógico” sabe que el foco está apagado.

Mientras que cuando el foco está encendido, el voltaje que cae entre los puntos **a** y **b** es de 0 voltios, con lo que el opto-transistor siempre va estar en corte generando que a la salida en el punto **c** se obtenga un voltaje de 0V, de igual manera este valor de voltaje entra directamente al microcontrolador en el cual al leer el “0 Lógico” sabrá que el foco se encuentra encendido.

Con esto, el microcontrolador; al saber el estado en el que se encuentra el foco puede apagarlo si éste se halla encendido o viceversa.

3.5. DISEÑO DEL SISTEMA DOMÓTICO

El sistema domótico planteado en este proyecto de titulación, se basa principalmente en la comunicación del PDA con el sistema de control, por medio de tecnología Bluetooth, la cual envía comandos al microcontrolador ATMEGA64, el mismo que valora y ejecuta las instrucciones necesarias para realizar dicho control por medio de hardware.

A continuación, se detalla mediante: las principales líneas de programación, diagramas de flujo que se encuentran en el Anexo A, y circuitos eléctricos necesarios para el correcto funcionamiento del sistema domótico.

3.5.1. PRESENTACIÓN DEL SISTEMA DOMÓTICO

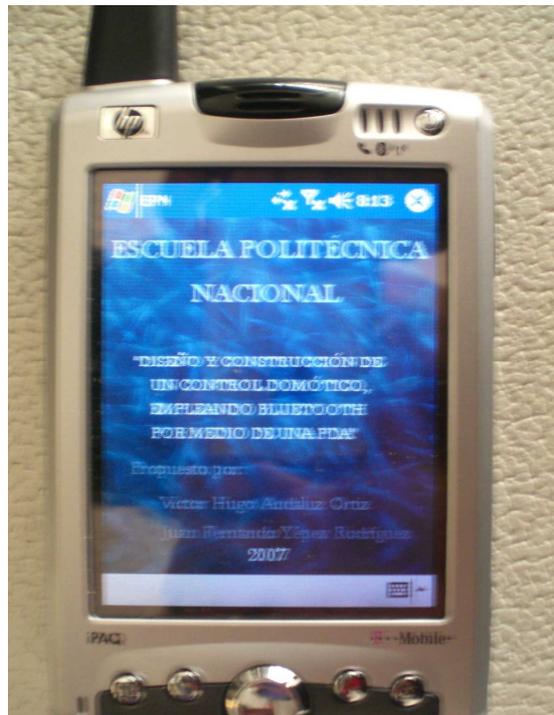


Figura 3.12. Presentación de la Carátula

Al ingresar se tiene una presentación en la cual se muestra por un tiempo determinado (7 segundos aproximadamente) una carátula (figura 3.12), con el

nombre de la aplicación y sus autores. Mientras transcurre dicho tiempo se escucha un mensaje de bienvenida: *"Bienvenido al Sistema Domótico"*.

Para obtener esta pantalla se ha seguido el siguiente procedimiento:

Se creó un formulario llamado *Carátula* (Figura 3.13), el cual consta de un cuadro de imagen que trae el gráfico de la ubicación "\btn\Caratula.bmp" y lo muestra como fondo en el momento de cargar el formulario (Figura 3.14). Además consta de un reloj (timer) que determina el tiempo durante el cual se presenta dicha carátula. Después de este tiempo (2 segundos) se envía un comando para reproducir un archivo de audio que está ubicado en "\sonidos\inicio.wav".

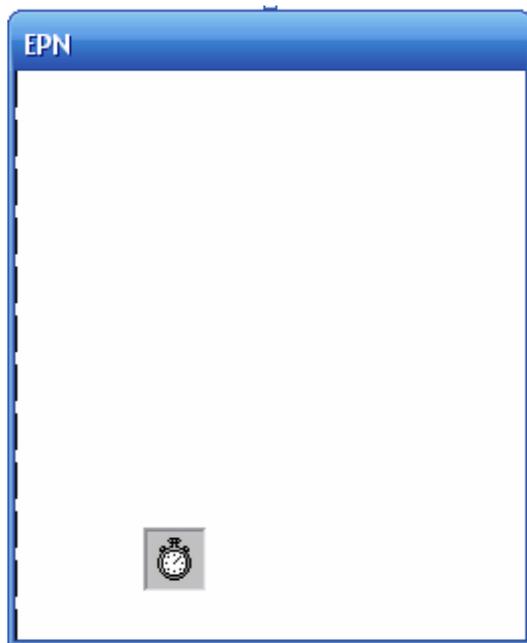


Figura 3.13. Formulario Carátula

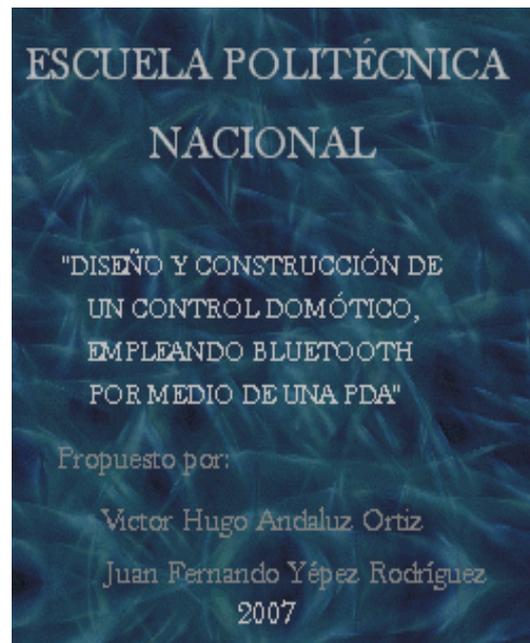


Figura 3.14. Imagen Carátula

//----- Código del formulario caratula ----- //

'Instrucciones que se ejecutará en el momento de cargar el formulario.

Private Sub Form_Load()

```

Img.Picture = App.Path + "\btn\Caratula.bmp" 'carga la imagen de presentación
Caratula.SIPVisible = False                'oculta el teclado
End Sub                                    'finaliza la subrutina

```

'Subrutina que controla el objeto Timer llamado Tmrbtn.

```

Private Sub Tmrbtn_Timer()
PlaySoundFile ("\sonidos\inicio.wav")    'reproduce el sonido de inicio
cont.Show                                 'carga el formulario de la
                                           contraseña llamado cont
Tmrbtn.Enabled = False                   'deshabilita el timer
End Sub                                    'finaliza la subrutina

```

3.5.2. INGRESO DE LA CONTRASEÑA

Seguidamente se presenta una nueva pantalla, en la cual el usuario debe ingresar una clave que consta de cuatro dígitos (Figura 3.15).



Figura 3.15. Presentación de Bienvenida

En caso de que la clave ingresada no sea la correcta el sistema mostrará el siguiente mensaje: “**¡La Clave Ingresada es Incorrecta!**” (Figura 3.16).



Figura 3.16. Mensaje: La clave es incorrecta



Figura 3.17. Ingreso correcto de la clave

Una vez ingresada correctamente la clave, el usuario continúa con el proceso de conexión (Figura 3.17).

El proceso de validación de clave se la realizó con el formulario *Contraseña* Figura 3.18.

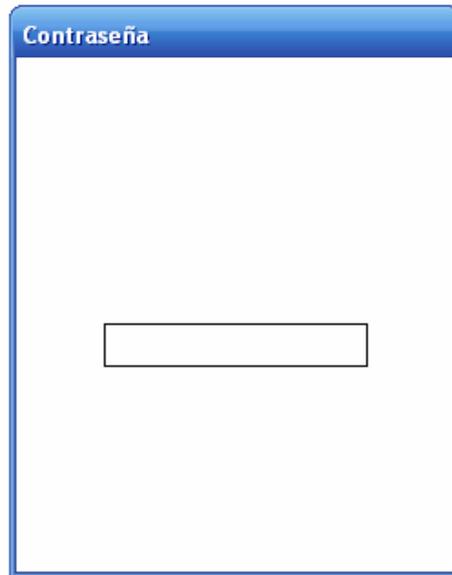


Figura 3.18. Formulario Contraseña

El formulario consta de una imagen (Figura 3.19), la misma que se encuentra en la ubicación "\btn\Clave.bmp" y se la carga como fondo al momento de iniciar esta pantalla.

Además consta de un cuadro de texto, que verifica si el carácter es un número cada vez que se digite. En caso de serlo se guarda este número en un registro, por cada número ingresado se imprime un "*" el cuadro de texto.

Si el carácter no es un número el sistema solo permite el ingreso de dos caracteres más los cuales son: BACKSPACE el mismo que borra el último carácter válido ingresado y ENTER el cual manda la cadena de números al proceso de verificación de clave. En este proceso se compara la clave ingresada con la clave guardada en la memoria de la Pocket PC. En caso de que la clave no sea la correcta mediante el comando MsgBox se muestra el mensaje "¡La Clave Ingresada es Incorrecta!", y se borra la información de los dígitos ingresados.

La clave consta de cuatro dígitos, los cuales deben ser solo números.



Figura 3.19. Imagen Clave

Para el ingreso de la contraseña se ejecuta el siguiente código.

//----- Código del formulario Contraseña ----- //

'Instrucciones que se ejecutará en el momento de cargar el formulario.

Private Sub Form_Load()

imgclave.Picture = App.Path + "\btn\Clave.bmp" *'muestra la imagen en el cuadro de
imagen llamado imgclave.*

Caratula.Hide *'oculta la pantalla anterior formulario
Caratula.*

End Sub *'finaliza la subrutina.*

'Instrucciones de la subrutina Continuar que permiten validar la clave.

'el registro en donde se guarda la clave está en la etiqueta Label4.

'el cuadro de texto donde el usuario ingresa la clave se llama Text1.

Private Sub Continuar()

If Label4.Caption = "1234" Then *'comprueba si la contraseña es correcta.*

```

cont.SIPVisible = False           'oculta el teclado del actual formulario.
Menu.SIPVisible = False           'oculta el teclado del siguiente formulario.
Menu.Show                          'muestra el siguiente formulario llamado Menú.
Else
    MsgBox "¡La Clave Ingresada es Incorrecta!", vbCritical, "Error"
                                    'muestra un msj de error.
    Text1.Text = ""                'limpia el texto de la contraseña.
    Label4.Caption = ""            'limpia el valor de la contraseña.
End If
End Sub                             'finaliza la subrutina.

```

'Subrutina que controla el ingreso de cada carácter en el cuadro de texto.

```

Private Sub text1_KeyPress(KeyAscii As Integer)
Dim a As Integer                   'define la variable de longitud.
Dim i As Integer                   'define un contador.
Dim Clave As String                'variable que almacena la contraseña.
Select Case KeyAscii               'proceso de validación de la contraseña.
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8, 13 'restringe el ingreso de letras.
a = Len(Text1.Text)                'calcula la longitud de la clave.
If KeyAscii = 13 Then              'si un Enter es ingresado se verifica que la
                                    contraseña es correcta.
    If Label4.Caption = "1234" Then 'compara la contraseña del registro.
        Menu.Show                  'si es correcta muestra el formulario Menú.
    Else                            'si no se cumple.
        Call Continuar              'se llama a la subrutina Continuar.
    End If                          'finaliza la condición.
End If                              'finaliza la condición.
If KeyAscii = 8 And a >= 0 Then    'Compara la longitud del texto y el carácter
                                    ingresado.
    If a = 0 Then                   'si la longitud es cero.
        Label4.Caption = ""         'se borra el registro.
    Else                            'si no.
        Text1.Text = ""            'se borra el cuadro de texto.
    End If
End If

```

```

For i = 1 To a - 1           'se realiza un lazo.
    Text1.Text = Text1.Text + "*"   'para cambiar el texto por asteriscos
Next I                       'finaliza el lazo
Clave = Label4.Caption       'se guarda la clave en la variable del mismo
                              nombre
End If                       'finaliza la condición.
Else
If KeyAscii = 13 Then       'si el carácter es el Enter
    Text1.Text = ""         'se limpia el cuadro de texto.
    Label4.Caption = ""     'se borra el registro.
    KeyAscii = 0           'se borra el carácter.
Else
If a < 6 Then               ' se debe ingresar una clave menor a 6 dígitos
Label4.Caption = Label4.Caption + Chr(KeyAscii) 'se almacena la clave
KeyAscii = 0               'se limpia el carácter
Text1.Text = Text1.Text + "*" 'se cambia el dígito por un carácter.
Else                       'si no
KeyAscii = 0               'se limpia el carácter y no se imprime nada.
End If                     'finaliza la condición.
End If                     'finaliza la condición.
End If                     'finaliza la condición.
End Sub                    'finaliza la subrutina.

```

' Timer llamado Tmrfoco que permite ubicar en un tiempo el cursor en el cuadro de texto.

```

Private Sub Tmrfoco_Timer()
Text1.SetFocus             'comando que ubica el cursor
Tmrfoco.Enabled = False   'deshabilita el Timer .
End Sub                   'finaliza la subrutina.

```

3.5.3. DISPOSITIVOS INALÁMBRICOS DEL PDA

Si la clave ingresada es la correcta, el sistema muestra la pantalla de los dispositivos inalámbricos que contiene la Pocket PC (Figura 3.20), y emite el siguiente mensaje de voz: "Verifique que la opción de Bluetooth se encuentre encendido".

El usuario debe verificar que el medio de conexión Bluetooth se encuentra encendido (el ícono debe tener el color verde).



Figura 3.20. Dispositivos inalámbricos de la Pocket PC

Para llamar a la pantalla de los dispositivos inalámbricos del PDA, se ejecuta el comando Shell el mismo que busca el archivo "iPAQWireless.exe" desde el directorio raíz.

```
IIResult = Shell("iPAQWireless.exe", App.Path & "\myapp.htm#main_contents")
```

En caso de que no se muestre la pantalla anteriormente mencionada se imprime el mensaje "Shell() error: "

```
If IIResult <> 0 Then MsgBox "Shell() error: " & IIResult
```

Para el mensaje de voz se reproduce un archivo de audio de la ubicación "\sonidos\encendido.wav".

```
PlaySoundFile ("\sonidos\encendido.wav")
```

Si todo el proceso se realizó correctamente, a continuación se muestra la pantalla principal, en la cual se encuentran las opciones necesarias para el manejo del sistema por parte del usuario.

Para mejorar la interfaz visual con el usuario se crearon cuatro pestañas llamadas: Bluetooth, Iluminación, Seguridad y Horarios; las mismas que asocian en sus propiedades los objetos usados para cada aplicación.

A continuación se indica el funcionamiento de cada una:

3.5.4. PESTAÑA BLUETOOTH

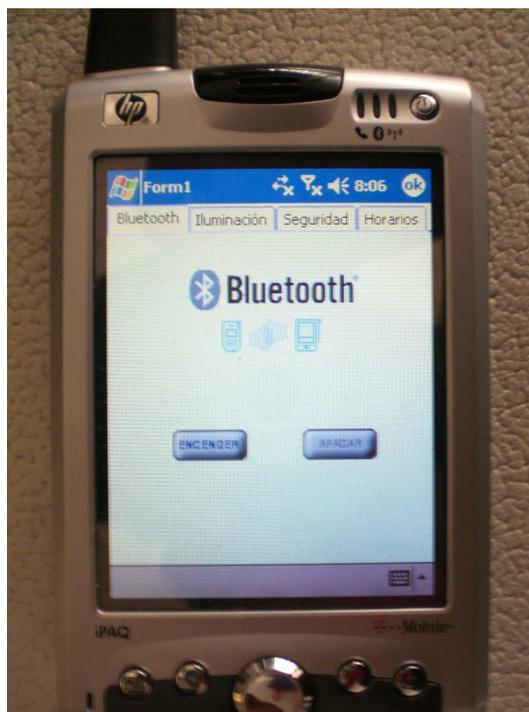


Figura 3.21. Dispositivo Bluetooth desactivado

La pestaña *Bluetooth* permite la conexión con la pasarela Domótica. Al iniciar la aplicación el icono de *Bluetooth* tiene un color gris (Figura 3.21).

Cuando se presiona el ícono *Encender* y se conecta correctamente los dispositivos Bluetooth, el icono cambia a color azul (Figura 3.22). Solo en este momento se podrá acceder a las otras pestañas del formulario.



Figura 3.22. Dispositivo Bluetooth activado

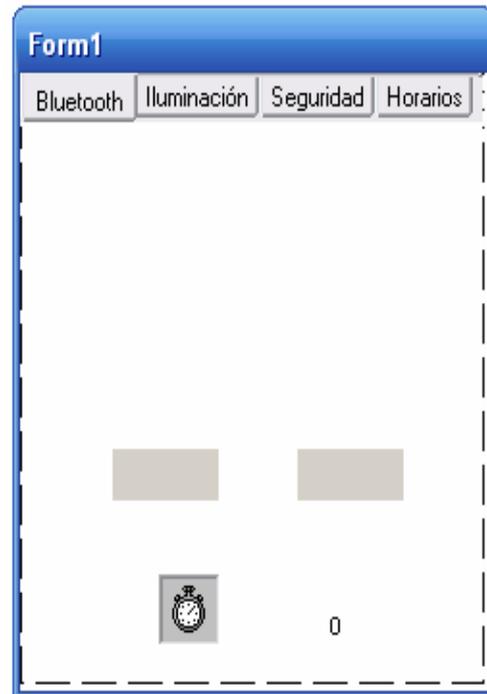


Figura 3.23. Formulario Menú

El formulario *Menú* (Figura 3.23), es el que permite la conexión inalámbrica por medio de Bluetooth entre el PDA y la pasarela para utilizar todos los servicios del Sistema Domótico: Iluminación, Seguridad y Horarios.

3.5.4.1. Carga de Imágenes y botones del Formulario Menú

Para ingresar a la pantalla principal Figura 3.21., se carga inicialmente todas las imágenes, botones e información necesaria de las cargas a controlar.

a) Para la Pestaña Bluetooth

Para la opción Bluetooth previamente se verifica si el modo Bluetooth de la PDA se encuentra activado o no.

En caso de que esté desactivado, se cargan:

- La imagen del fondo de pantalla (Figura 3.24) la cual se extrae de la ubicación "\btn\blueoff.bmp".
- La del botón de encendido (Figura 3.25), de la ubicación "\btn\btnencender.bmp".
- La del botón de apagado (Figura 3.26), de "\btn\btnapagar1o.bmp".

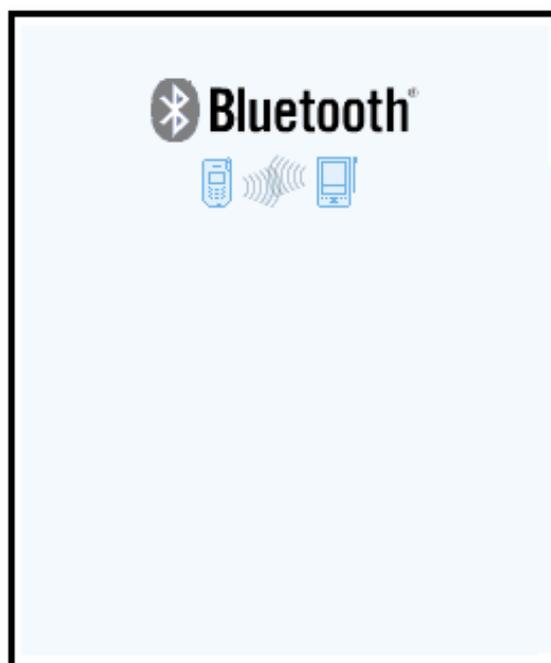


Figura 3.24. Blueoff



Figura 3.25. Botón encendido

Figura 3.26. Botón apagado

En el caso de que esté activado, se cargan:

- La imagen de fondo de pantalla (Figura 3.27), la cual se extrae de la ubicación "\btn\bluelogo.bmp".
- La del botón de encendido (Figura 3.28), de la ubicación "\btn\btnencender1o.bmp".
- La del botón de apagado (Figura 3.29), de "\btn\btnapagar1.bmp"

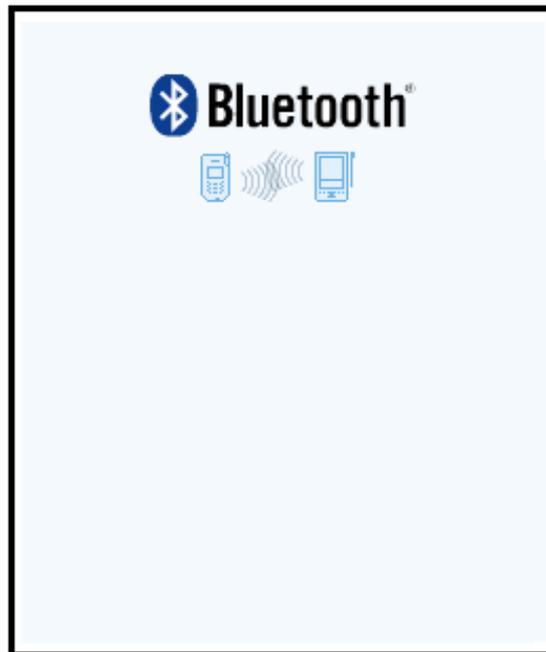


Figura 3.27. bluelogo



Figura 3.28. Botón encendido



Figura 3.29. Botón apagado

//----- Código de la pestaña Bluetooth ----- //

'Condición que compara la conexión del puerto de comunicaciones Comm1.

If Comm1.PortOpen = False Then	<i>'Si esta desconectado.</i>
Picbtn1.Picture = App.Path + "\btn\btnencender.bmp"	<i>'se cargan las imágenes.</i>
picbtn2.Picture = App.Path + "\btn\btnapagar1o.bmp"	<i>'de esta desconexión.</i>
Imgblue.Picture = App.Path + "\btn\blueoff.bmp"	
Else	<i>'Si no.</i>
Picbtn1.Picture = App.Path + "\btn\btnencender1o.bmp"	<i>'se muestra</i>
picbtn2.Picture = App.Path + "\btn\btnapagar1.bmp"	<i>'las imágenes de la</i>
Imgblue.Picture = App.Path + "\btn\bluelogo.bmp"	<i>'conexión</i>
End If	<i>'finaliza la condición.</i>

b) Para la Pestaña Iluminación

En la opción iluminación se cargan dos imágenes las cuales dependen del estado actual de la carga eléctrica (foco), si la carga está encendida se registra la imagen luzon (Figura 3.30), y si la carga esta desconectada se enciende la imagen luzoff (Figura 3.31).



Figura 3.30. luzon



Figura 3.31. luzoff

```
Luz0.Picture = App.Path + "\btn\luzoff.bmp"
```

```
Luz1.Picture = App.Path + "\btn\luzon.bmp"
```

c) Para la Pestaña Seguridad

En la opción Seguridad también se cargan dos imágenes las cuales dependen del estado actual del sensor. Si el sensor es activado se carga la imagen srojo (Figura 3.32), y si está desactivado se carga la imagen sverde (Figura 3.33).



Figura 3.32. srojo



Figura 3.33. sverde

```
imgsensor1.Picture = App.Path + "\btn\srojo.bmp"  
imgsensor2.Picture = App.Path + "\btn\sverde.bmp"
```

d) Para la Pestaña Horarios

En la opción Horarios, en el bloque *ComboBox* (Figura 3.43) se agregan los ítems para las diez cargas eléctricas a controlarse.

```
cmbhorario.AddItem "Carga1"  
cmbhorario.AddItem "Carga2"  
cmbhorario.AddItem "Carga3"  
cmbhorario.AddItem "Carga4"  
cmbhorario.AddItem "Carga5"  
cmbhorario.AddItem "Carga6"  
cmbhorario.AddItem "Carga7"  
cmbhorario.AddItem "Carga8"  
cmbhorario.AddItem "Carga9"  
cmbhorario.AddItem "Carga10"
```

3.5.4.2. Conexión entre el PDA y la Pasarela Domótica

La conexión entre los dispositivos Bluetooth del Sistema (PDA y Pasarela) se realiza de la siguiente manera:

a) **Comunicación del PDA**

Se selecciona el Puerto de Comunicaciones correspondiente al usado para la salida Serial de Bluetooth., en este caso el puerto número siete. La comunicación se realiza a 19200 baudios, con un bit de inicio, sin bit de paridad, 8 bits de datos, y un bit de parada.

'El control de comunicaciones usado es el Comm1, al cual se lo configura de esta manera.

Comm1.CommPort = 7	<i>'se escoge el puerto.</i>
Comm1.Settings = "19200, N, 8, 1"	<i>'se configura los parámetros de transmisión.</i>
Comm1.PortOpen = True	<i>'se abre el puerto.</i>
Comm1.RThreshold = 1	<i>'se genera el evento comm1. por cada carácter.</i>
Comm1.SThreshold = 1	<i>'se envía cada carácter.</i>

Si el proceso de conexión falla se muestra un mensaje de error: *"No se pudo conectar con el dispositivo, por favor vuelva a intentarlo"*, con lo que las imágenes de la pantalla de Bluetooth vuelven al modo de desconexión.

Si se conectó correctamente se llama al proceso de *Sincronización*, el cual actualiza los datos de fecha, hora y estado de las cargas del microcontrolador con la información que emite el PDA y las imágenes se muestran en modo de conexión.

'se verifica si el proceso de conexión se lo realizó correctamente.

If Comm1.PortOpen = False Then	<i>' si el puerto no se encuentra abierto.</i>
MsgBox "No se pudo conectar con el dispositivo, por favor vuelva a intentarlo", vbCritical, "Dispositivo no disponible"	<i>'se muestra un mensaje de error.</i>

```

Picbtn1.Picture = App.Path + "\btn\btnencender.bmp"      'se cambian las
picbtn2.Picture = App.Path + "\btn\btnapagar1o.bmp"      'imágenes.
Else                                                       'si no
  Imgblue.Picture = App.Path + "\btn\bluelogo.bmp"       'se muestra las
  picbtn2.Picture = App.Path + "\btn\btnapagar.bmp"      'imágenes de conexión.
  Call sincronizacion                                  'se llama a la subrutina sincronización.
End If                                                     'se finaliza la condición.

```

b) Comunicación con el microcontrolador

El microcontrolador está configurado para trabajar con una frecuencia de reloj interno de 8MHz con una velocidad de señal de 19200 baudios y un cristal externo 32768 Hz.

```

$regfile = "m64def.dat"                                  ' se especifica el micro a usar.
$crystal = 8000000                                     ' definir la frecuencia interna del micro.
$baud = 19200                                          ' velocidad de la señal.

```

El sistema se encuentra configurado para utilizar un LCD de 16 columnas y 2 filas

```

Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 = Portb.7 , E = Portb.3 ,
Rs = Portb.2                                           ' asignación de pines del lcd con los de lmicro.
Config Lcd = 16 * 2                                    ' tipo de lcd.

```

Para la comunicación con el dispositivo Bluetooth KC-11, se utiliza el comando "AT+ZV RESET" para resetear el módulo antes de comenzar la comunicación. En ese instante el dispositivo se pone en estado de esclavo esperando que otro dispositivo Bluetooth se comuniquen con él. En el momento de la comunicación espera la letra "H" la cual es un comando enviado por el PDA para comenzar la sincronización

Esclavo:	<i>' El buetooth (BT) se pone en modo de esclavo</i>
Cls	<i>' borra la pantalla del LCD.</i>
Wait 2	<i>' espera 2 segundos.</i>
Atin = "AT+ZV RESET"	<i>' comando para resetear el BT.</i>
Print Atin	<i>' envía el comando al modulo BT.</i>
Wait 1	<i>' espera 1 segundo.</i>
Locate 1 , 1	<i>' Posiciona el cursor del LCD</i>
Lcd "Sistema Domotico"	<i>' Muestra este mensaje en el LCD</i>
Do	
\$timeout = 200000	<i>' Tiempo de espera de Datos</i>
Input Atin Noecho	<i>' Espera un comando de la PDA</i>
Loop Until Atin = "H"	<i>' haga esto hasta que se envíe "H" para</i>
	<i>'comenzar la sincronización con el pocket pc</i>

3.5.4.3. Proceso de Sincronización

El Proceso de Sincronización permite que el PDA envíe algunos parámetros que el microcontrolador necesita para realizar actividades como la de simulación.

a) En el PDA

Al enviar el comando "H" se inicia el proceso de sincronización, con la subrutina "horam" envía la hora.

'la subrutina sincronización envía la fecha y hora hacia la pasarela.

'la subrutina enviar selecciona todos los caracteres del text1 y los envía por el puerto serial

Private Sub sincronizacion()

Text1.Text = "H"

'alerta de comunicación

Call enviar

'envía el comando.

Call horam

'llama a la subrutina horam.

Call enviar

'envía la hora.

Call fecham

'llama a la subrutina fecham.

Call enviar *'envía la fecha.*
End Sub *'finaliza la subrutina*

La subrutina "horam" trabaja con el comando *Time* que es el encargado de extraer la hora del sistema operativo del PDA, que se encuentra en el formato am/pm, la subrutina es la encargada de convertir este formato al de 24 horas ya que el Reloj en Tiempo Real (Real Time) interno del microcontrolador trabaja con dicho formato.

//----- Código de la subrutina horam ----- //

'la subrutina horam extrae la hora del sistema operativo del PDA.
'la instrucción Right("Texto",# de caracteres) muestra una determinada cantidad de caracteres desde la derecha del texto ingresado.
'la instrucción Mid("Texto", inicio, # de caracteres) muestra una determinada cantidad de caracteres desde una ubicación de inicio del texto ingresado.
'con la instrucción Val se cambia un dato string a un valor entero.

```
Private Sub horam()  
If Right(Time, 2) = "PM" Then 'si la hora tiene el formato PM  
Text1.Text = Val(Mid(Time, 1, 1)) + 12 'a la hora actual se suman 12 hora.  
Else 'si la hora tiene el formato AM.  
    If Mid(Time, 1, 2) = "12" Then 'si la hora es 12AM.  
        Text1.Text = "00" & Mid(Time, 3, 6) 'se cambia por 0 horas.  
    Else  
        If Mid(Time, 2, 1) = ":" Then 'si se tiene dos punto.  
            Text1.Text = "0" & Mid(Time, 1, 7) 'se agrega un 0 a la hora.  
        Else 'sino.  
            Text1.Text = Mid(Time, 1, 8) 'se ingresa la hora sin cambios.  
        End If 'finaliza la condición.  
    End If 'finaliza la condición.  
End If 'finaliza la condición.
```

'con estas instrucciones se logra cambiar el formato de la hora de am/pm al formato de 24H.

Para la subrutina "fecham" se trabaja con el comando Date el mismo que extrae la fecha del sistema operativo de la PDA la que se encuentra en el formato americano Mes/ Día/ Año.

//----- Código de la subrutina fecham ----- //

'la subrutina fecham extrae la fecha del sistema operativo del PDA.

```
Private sub fecham()  
If Mid(Date, 2, 1) = "/" Then           'si se tiene un "/" significa que.  
    Text1.Text = "0" & Mid(Date, 1, 2)   'el mes es de valor menor a 10 con lo que  
If Mid(Date, 4, 1) = "/" Then         'se añade un cero, así también para el día.  
    Text1.Text = Text1.Text & "0" & Mid(Date, 3, 4)  
Else                                   'sino se toma los datos como están.  
    Text1.Text = Text1.Text & Mid(Date, 3, 5)  
End If                                 'finaliza la condición.  
Else  
    Text1.Text = Mid(Date, 1, 3)         'la fecha se almacena en el text1  
If Mid(Date, 5, 1) = "/" Then         'con el formato mm/dd/aa para  
    Text1.Text = Text1.Text & "0" & Mid(Date, 4, 4)  
Else                                   'luego ser transmitida con el comando enviar  
    Text1.Text = Text1.Text & Mid(Date, 4, 5)  
End If                                 'finaliza la condición.  
End If                                 'finaliza la condición.  
End sub                                 'finaliza la subrutina.
```

La subrutina "enviar" cuenta el número de caracteres que se encuentran en Text1.Text y los envía de forma serial uno por uno.

//----- Código de la subrutina enviar ----- //

'la subrutina enviar transmite el texto almacenado en el text1.

```
Private Sub enviar()  
    a = Len(Text1.Text)           'calcula la longitud del texto.  
    For j = 1 To a                 'crea un lazo mediante el cual  
        Comm1.Output = Mid(Text1.Text, j, 1) 'envía los caracteres uno por uno.  
    Next j                         'finaliza el lazo.  
    Comm1.Output = Chr(13)        'envía un Enter luego del texto  
    Text1.Text = ""               'limpia el cuadro de texto  
End Sub                            'termina la subrutina
```

b) En el microcontrolador

Se habilita la interrupción y la configuración del Real Time. Se utiliza el formato mes/día/año para la fecha, y hora/minutos/segundos para la hora.

```
Enable Interrupts                 'habilita las interrupciones  
Config Date = Mdy , Separator = / 'formato de separación /  
Config Clock = Soft              'usar el real time interno
```

Se utiliza el comando Input para esperar un dato que llega por el Puerto Serial y se almacena en la variable *Atin*. El primer dato que se recibe es la hora, que se almacena en la variable *Time\$* la cual va actualizando constantemente su valor. El segundo dato recibido corresponde a la fecha que es guardado en la variable denominada "*Date\$*". Luego de este proceso se calcula el día de la semana para la fecha ingresada, esta información es mostrada en el LCD.

Estas variables son modificadas continuamente, ya que son parte de las instrucciones del Real Time.

```
Dim Atin As String * 20          'variable para almacenar los datos.  
$timeout = 500000                'espera los datos 500 ms.  
Input Atin Noecho                'ingreso de la hora.  
Time$ = Atin                     'guarda la hora / minutos / segundos.
```

Input Atin Noecho	<i>'ingreso de la fecha.</i>
Date\$ = Atin	<i>'guarda el mes / día / año.</i>
Dim Bitdia As Byte	<i>'variable para almacenar el día</i>
Bitdia = Dayofweek()	<i>'calcula el día de la semana</i>
Atin = Lookupstr(bitdia , Weekdays)	<i>'busca el nombre en la tabla Weekdays</i>
Cls	<i>'borra la pantalla</i>
Locate 1 , 1	<i>'posiciona en el LCD</i>
Lcd Date\$	<i>'imprime Fecha en el LCD</i>
Locate 1 , 10	<i>'posiciona en el LCD</i>
Lcd Atin	<i>'imprime el día en el LCD</i>

'tabla de días de la semana

Weekdays:

Data "Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"

3.5.5. PESTAÑA ILUMINACIÓN

La opción Iluminación muestra una pantalla (Figura 3.34), en la cual se encuentran diez cargas eléctricas que indican su estado actual, encendido o apagado. Cada carga está representada por medio de una imagen de un foco el cual indica su estado, para esto, si el foco es de color amarillo la carga eléctrica está conectada, caso contrario está desconectada.

Las cargas pueden ser controladas de dos maneras, con el PDA y manualmente, como se conoce el estado de las cargas por medio del PDA se les puede modificar a conveniencia, sin la necesidad de estar físicamente presente en ese lugar.

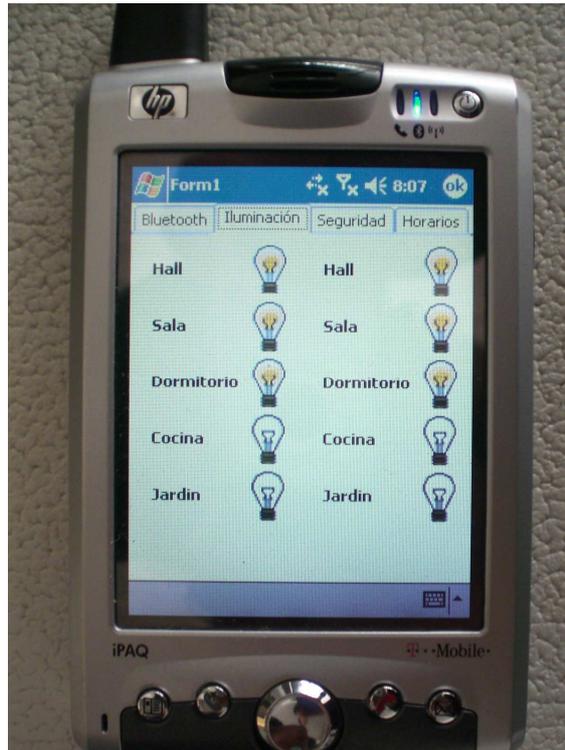


Figura 3.34. Pantalla Iluminación

Una vez activada la comunicación, existe un intercambio de datos que se efectúa cada vez que ocurra algún cambio en cualquiera de los dispositivos para lo cual hay un intercambio de información.

3.5.5.1. Forma de envío de los comandos

La forma de envío de la información referente a la carga emitida por microcontrolador ATMEGA64 es la siguiente (Figura 3.35):



Figura 3.35. Formato de envío de carga

Carga: se escribe la letra "R" si se trata de la información de una carga.

Número de la carga: este es un número que va del 0 al 9, es decir se puede controlar 10 cargas como máximo.

Estado de la carga: informa con una letra si se encuentra encendido "E" o apagado "A".

Ejemplo: R5E:

"R" se trata de una carga.

"5" la carga número 5.

"E" la carga está encendida.

a) **En el microcontrolador**

Luego del proceso de sincronización el microcontrolador chequea el estado actual de las cargas, si existe algún cambio de cualquiera de éstas, dicha información es enviada al PDA.

Se presenta las líneas de código del microcontrolador solo para el control de una carga.

'se verifica el estado inicial de cada carga

Dim Espera as Integer

'variable que almacena un valor de retardo

Control0 = Pine.7

'configura variables de control

Control1 = Pine.6

'para cada carga

Espera = 100

'se configura un retardo de 100 ms.

If Pine.7 = 0 Then

'si la carga 0 está encendida

Print "R0E"

'se envía el comando R0E al PDA.

Waitms Espera

'se genera el retardo.

End If

'finaliza la condición.

If Pine.6 = 0 Then	<i>'si la carga 1 está encendida</i>
Print "R1E"	<i>'se envía el comando R1E al PDA.</i>
Waitms Espera	<i>'se genera el retardo.</i>
End If	<i>'finaliza la condición.</i>

'se compara el estado anterior con el estado actual.

Do	<i>'inicia un lazo</i>
Call Lectura	<i>' lee un comando</i>
If Pine.7 = 1 And Control0 = 0 Then	<i>'1º carga detector de estado = pine.7</i>
Print "R0A"	<i>'interruptor = pina.0</i>
Control0 = Pine.7	<i>'guarda el estado actual.</i>
Call Lectura	<i>'lee un comando</i>
End If	<i>'finaliza la condición.</i>

If Pine.7 = 0 And Control0 = 1 Then	<i>'se comparan los 2 tipos de variación</i>
Print "R0E"	<i>'y se envía el comando de encendido</i>
Control0 = Pine.7	<i>'o apagado correspondiente a cada caso.</i>
Call Lectura	<i>'lee un comando.</i>
End If	<i>'finaliza la condición.</i>
Loop	<i>'mantiene el lazo</i>

'para las anteriores líneas de código se utiliza la subrutina Lectura.

Sub Lectura()	
Input Atin Noecho	<i>'espera un dato sin eco.</i>
End Sub	<i>'finaliza la subrutina.</i>

b) En el PDA

Para ingresar a esta nueva pantalla denominada Iluminación, previamente se realiza una comprobación de que existe comunicación entre el PDA y la Pasarela, caso contrario no se puede acceder a dicha pantalla.

If Comm1.PortOpen = False Then	<i>'si el puerto no esta abierto, envía el mensaje</i>
MsgBox "Escoja primero un dispositivo Bluetooth", vbInformation, "No hay conexion"	

```

Else
    Frm1.Visible = False           'oculta la pestaña bluetooth.
    Frm2.Visible = True           'muestra la pestaña iluminación.
    Frm3.Visible = False         'oculta la pestaña seguridad.
    Frm4.Visible = False         'oculta la pestaña horarios.
End If                             'finaliza la condición.

```

La pestaña *Iluminación* (Figura 3.36), consta de diez cuadros de imagen los cuales permiten mostrar el estado actual de los focos y haciendo un click sobre ellos se podrá modificar su estado.



Figura 3.36. Pestaña Iluminación

Cuando ocurre algún evento en el Puerto serial del PDA, esta información es almacenada en la variable *tin* y se llama a la subrutina *Comandos*.

```
//----- Código del objeto Comm1 ----- //
```

'Subrutina que se activa cuando ocurre algún evento en el puerto Comm1

End Function

'finaliza la función.

3.5.5.2. Cambio de estado de las cargas

Si se desea cambiar el estado de cualquiera de las cargas se realiza un click en la imagen del foco del cual se desea cambiar su estado.

a) En el PDA

Para lo cual el PDA envía a la pasarela, el número de la carga que se quiere cambiar de estado precedido de la letra "L".

'la subrutina Luz0 se activa cuando se hace click.

Private Sub Luz0_Click()

Text1.Text = "L0"

'si carga el comando "L0".

Call enviar

'se envía el comando

End Sub

'finaliza la subrutina.

b) En el microcontrolador

El microcontrolador lee la información que le llega por el Puerto Serial, la almacena en la variable *Atin*, luego verifica si el primer carácter del comando es la letra "L" y cambia el estado de la carga que se indica el segundo carácter de dicho comando.

Input Atin Noecho

'ingresa los datos a la variable Atin.

If Mid(atin , 1 , 1) = "L" Then

'si se tiene el comando "L"

If Mid(atin , 2 , 1) = "0" Then

'se pregunta el número de la carga.

If Pine.7 = 0 Then

'si esta encendido

Print "R0A"

'manda el comando de apagado.

Else

'sino

Print "R0E"

'manda el comando de encendido.

```

End If                                     'finaliza la condición
If Porta.0 = 1 Then                         'pregunta el estado actual
    Porta.0 = 0                             'cambia el estado
Else                                         'de la carga.
    Porta.0 = 1                             'si esta 0 cambia a 1
End If                                       'finaliza la condición.
Call Lectura                               'llama la subrutina Lectura
End If                                       'finaliza la condición.
End If                                       'finaliza la condición.

```

3.5.6. PESTAÑA SEGURIDAD

Al dar un click en la pestaña de Seguridad se ingresa a una pantalla (Figura 3.37), en la cual se observa el estado actual de los sensores. El color verde representa que el sensor está activado y el color rojo representa al sensor inactivo.

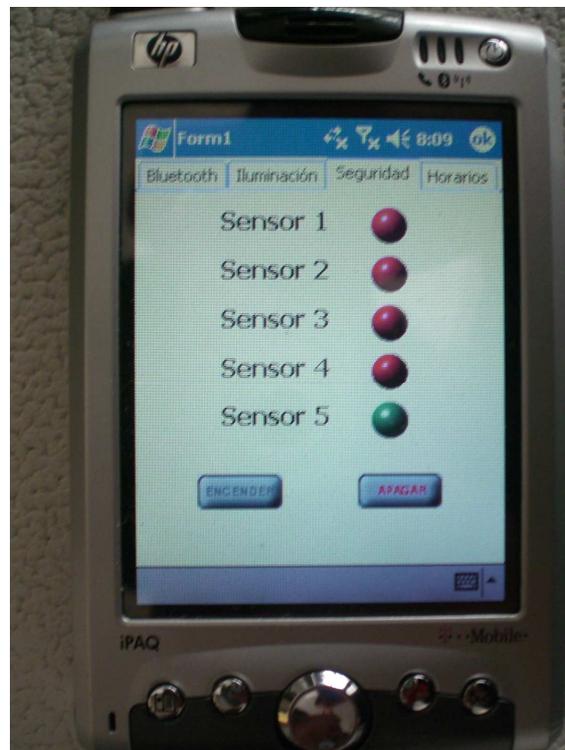


Figura 3.37. Pantalla de Seguridad

Consta además de dos botones: el primero se llama *ACTIVAR*, que al ser presionado habilita la alarma de seguridad, que encenderá una sirena (bocina) por un tiempo de dos minutos, cuando alguno de los sensores se active. En caso de que el sensor siga activado volverá a sonar la sirena y así sucesivamente hasta que los sensores estén inactivos o hasta que el usuario presione el segundo botón denominado *DESACTIVAR*, este botón deshabilita todo el sistema de seguridad.

3.5.6.1. Forma de envío de los comandos

La forma de envío de la información referente a los sensores emitida por microcontrolador ATMEGA64 es la siguiente (Figura 3.38):



Figura 3.38. Formato de envío de sensor

Sensor: se escribe la letra "S" si se refiere a la información de un sensor.

Número del sensor: el sistema puede tener como máximo 5 sensores.

Estado del sensor: informa con una letra si se encuentra activado "E" o inactivo "A".

Ejemplo: S3A:

"S" representa información de un sensor.

"3" la carga número 3.

"A" el sensor está inactivo.

a) En el microcontrolador

Se chequea el estado actual de los diferentes sensores, si existe algún cambio de cualquiera de estos, dicha información es enviada al PDA.

Se presenta las líneas de código del microcontrolador solo para dos sensores.

```
Sensor1 = Pind.4           'configura variables de control.
Sensor2 = Pind.5           'para cada sensor.
If Pind.4 = 0 Then         'si el sensor 0 está activo.
    Print "S1E"            'envía el comando S1E al PDA
    Waitms Espera         'genera el retardo
End If                     'finaliza la condición
If Pind.5 = 1 Then         'si el sensor 1 está activo.
    Print "S2E"            'envía el comando S2E al PDA
    Waitms Espera         'genera el retardo.
End If                     'finaliza la condición.
Do                          'inicia el lazo de chequeo de cargas y sensores
    Call Lectura           'llama la subrutina Lectura
    If Pind.4 = 1 And Sensor1 = 0 Then 'el 1° sensor está en el pind.4
        Print "S1E"        'compara su estado anterior con el actual
        Sensor1 = Pind.4   'guarda el estado actual
    End If
    If Pind.4 = 0 And Sensor1 = 1 Then 'si el estado del sensor cambia
        Print "S1A"        'se verifica si estaba encendido o apagado
        Sensor1 = Pind.4   'y se envía el comando correspondiente
    End If                 'a cada caso y así para todos los sensores
Loop
```

b) En el PDA

Para ingresar a la pantalla *Seguridad* previamente se realiza una comprobación de que existe comunicación entre la PDA y la Pasarela, caso contrario no se puede acceder a dicha pantalla.

End Function

'finaliza la función.

La pestaña *Seguridad* (Figura 3.39), consta de cinco cuadros de imagen los cuales permiten mostrar el estado actual de los sensores y haciendo un click sobre los botones de ACTIVAR o DESACTIVAR, se puede habilitar o no a la alarma de seguridad.



Figura 3.39. Pestaña Seguridad

3.5.7. PESTAÑA HORARIO

Al ingresar a la pantalla *Horario* (Figura 3.40), se encuentra un Combo Box, el mismo que muestra un lista de diez cargas eléctricas (focos), en el que se puede programar su horario de encendido y apagado.

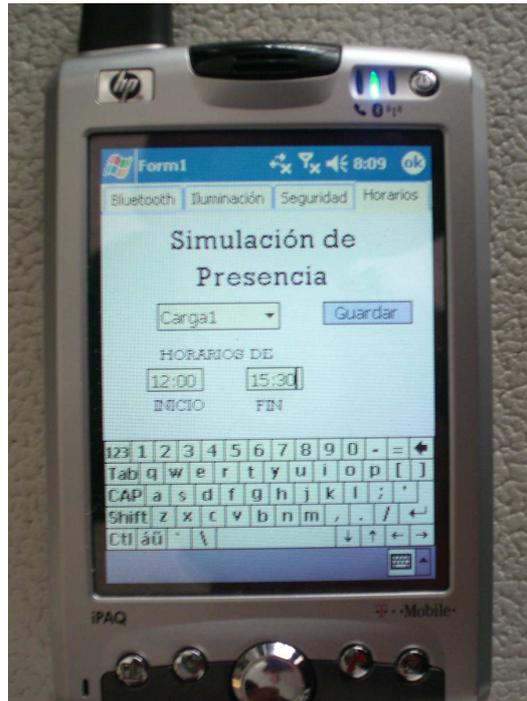


Figura 3.40. Pantalla Horario

Una vez seleccionada una carga, se muestra el horario que se encuentra programada actualmente. Este horario puede ser modificado por el usuario y con un click en el botón *GUARDAR* los valores quedan almacenados en la memoria interna del microcontrolador.

3.5.7.1. Forma de envío de los comandos

La forma de envío de la información referente al horario programado de cada carga enviada por microcontrolador Atmega64 tiene el siguiente formato (Figura 3.41):



Figura 3.41. Formato de envío de información de horario

Señal de inicio: el comando comienza con la letra "i" para informar al sistema que se va a enviar la hora de inicio.

Hora de inicio: tiene el siguiente formato "hh:mm" se envían dos dígitos correspondientes a la hora, que va desde 00 hasta 23; luego los dos puntos ":" y finalmente los minutos que van desde 00 hasta 59.

Señal de fin: continúa el comando con la letra "f" para informar al sistema que se va a enviar la hora de finalización.

Horario de fin: tiene el siguiente formato "hh:mm" se envían dos dígitos correspondientes a la hora que va desde 00 hasta 23; luego los dos puntos ":" y finalmente los minutos que van desde 00 hasta 59.

a) En el microcontrolador

El microcontrolador al recibir el comando "C" pregunta de qué número de carga desea saber su horario programado. Posteriormente entra a la subrutina *hora*, la misma que extrae los hora de inicio y fin almacenados en la memoria interna y los envía a la PDA.

//Se presenta las líneas de código para dos cargas.

If Mid(atin , 1 , 1) = "C" Then	<i>'si se recibe el comando con "C"</i>
Atin = Right(atin , 1)	<i>'se verifica a que carga corresponde</i>
If Atin = "1" Then	<i>'si es la primera carga</i>
Atin = "i"	<i>'se arma el formato de trama</i>
Call Hora(hi(1) , Hi(2))	<i>'se llama a la subrutina Hora</i>
Atin = Atin + "f"	<i>'se añade el separador de las horas</i>
Call Hora(hf(1) , Hf(2))	<i>'se llama a la subrutina Hora</i>
Print Atin	<i>'se envía la trama.</i>
Call Lectura	<i>'se llama a la subrutina Lectura</i>
End If	<i>'finaliza la condición</i>

If Atin = "2" Then	<i>'se procede a verificar</i>
Atin = "i"	<i>'a que carga corresponde</i>
Call Hora(hi(3) , Hi(4))	<i>'se arma la trama y se llama a la</i>
Atin = "f"	<i>'subrutina Hora que es la encargada</i>
Call Hora(hf(3) , Hf(4))	<i>'de mostrar los valores almacenados</i>
Print Atin	<i>'en memoria para luego enviar</i>
Call Lectura	<i>'la respuesta.</i>
End If	<i>'finaliza la verificación</i>
End If	<i>'finaliza la condición</i>

Subrutina hora es utilizada para dar el formato necesario para el envío de la hora.

Sub Hora(hin As Integer , Hfin As Integer)	<i>'Inicializa la subrutina Hora</i>
If Hin < 10 Then	<i>'si la hora almacenada es menor a 10</i>
Atin = Atin + "0"	<i>'entonces se antepone un 0</i>
End If	<i>'finaliza la condición</i>
Atin = Atin + Str(hin)	<i>'se añade las horas</i>
Atin = Atin + ":"	<i>'se separa con dos puntos</i>
If Hfin < 10 Then	<i>'si los minutos almacenados son menores a 10</i>
Atin = Atin + "0"	<i>'entonces se antepone un 0</i>
End If	<i>'finaliza la condición</i>
Atin = Atin + Str(hfin)	<i>'se añade los minutos</i>
End Sub	<i>'finaliza la subrutina</i>

b) En el PDA

Para ingresar a la pantalla *Horarios*, previamente se realiza la comprobación de que existe comunicación entre el PDA y la Pasarela, caso contrario no se puede acceder a dicha pantalla.

If Comm1.PortOpen = False Then	<i>'se verifica se está activo el Bluetooth</i>
MsgBox "Escoja primero un dispositivo Bluetooth", vbInformation, "No hay conexión"	

Else	<i>'si no lo está se muestra un mensaje de error</i>
Frm1.Visible = False	<i>'si está activa se ocultan las otras</i>
Frm2.Visible = False	<i>'pestañas y solo se muestra</i>
Frm3.Visible = False	<i>'la pestaña que se seleccionó.</i>
Frm4.Visible = True	
Text1.Text = "C1"	<i>'se envía el comando C1</i>
Call enviar	<i>'para recibir el horario de la carga 1</i>
End If	<i>'finaliza la verificación</i>

En la subrutina *Comandos* también se comprueba si el comando corresponde a la sección *Horarios*. Cuando se recibe el comando con el código "i" se extrae la información enviada por el microcontrolador, obteniendo así la hora de inicio y fin que son mostradas en los cuadros de texto respectivamente.

a = Len(tin)	<i>'se calcula la longitud del comando</i>
a = a - 2	<i>'se resta 2 caracteres (Enter y Space)</i>
tin = Mid(tin, 1, a)	<i>'se obtiene el comando completo</i>
If Mid(tin, 1, 1) = "i" Then	<i>'se chequea la hora de inicio</i>
Text3.Text = Mid(tin, 2, 5)	<i>'se extrae la hora de inicio</i>
If Text3.Text = "24:00" Then	<i>'si la hora es 24:00</i>
Text3.Text = "none "	<i>'no se tiene hora programada</i>
End If	<i>'finaliza la verificación</i>
Text4.Text = Mid(tin, 8, 5)	<i>'se extrae la hora de finalización</i>
If Text4.Text = "24:00" Then	<i>'si la hora es 24:00</i>
Text4.Text = "none "	<i>'no se tiene hora programada</i>
End If	<i>'finaliza la condición</i>
End If	<i>'finaliza la condición</i>

Si la hora de inicio y fin es la siguiente "24:00" significa que el horario programado está desactivado y consecuentemente se escribe en el cuadro de texto la palabra "none".

Cuando se realiza un click en un ítem del Combo Box llamado *Cmbhorario* se almacena en la variable *Cmbhora* la ubicación del ítem seleccionado, y se llama a la subrutina lista para realizar la petición del horario programado de la carga de dicho ítem.

```
Private Sub Cmbhorario_Click()           'Cuando se selecciona un dato del menú
    Cmbhora = cmbhorario.ListIndex      'se guarda el valor en la variable Cmbhora
    Call Lista                          'y se llama a la subrutina Lista
End Sub                                 'finaliza la subrutina
```

```
Private Sub Lista()                    'en la subrutina Lista se verifica
    If Cmbhora = 0 Then                'la carga que se seleccionó
        Text1.Text = "C1"              'se forma la trama con "C"
        Call enviar                    'y con el número de carga para
    End If                              'luego ser enviada a la pasarela
End Sub                                 'finaliza la subrutina.
```

3.5.7.2. Modificación del horario programado de una determinada carga

El cuadro de texto de hora de inicio se modifica en el Text3 y la hora de finalización en el Text4, cada vez que se seleccione el cuadro de texto se borran los datos que estaban en él para poder ser ingresados nuevamente. Si no se ingresó correctamente el valor, en la pantalla aparece el texto *"none"*.

Para el uso de esas condiciones se utiliza las funciones *"hinicio"* y *"hfin"*, además de la subrutina chequeo.

```
Private Sub Text3_Click()              'Cuando se hace click en la hora de inicio
    Call chequeo                       'se llama a la subrutina chequeo
    Text3.Text = ""                    'y se borra el texto anterior
End Sub                                 'finaliza la subrutina
```

```
Private Sub text3_KeyPress(KeyAscii As Integer) 'Cuando se ingresa la hora
```


3.5.7.3. Formato del proceso guardar

La forma de envío de la información del horario que ha sido modificado, enviado por el PDA (Pocket PC) tiene el siguiente formato (Figura 3.42):



Figura 3.42. Formato de envío de información de horario modificado

Número de la carga: éste es un número que va del 0 al 9, es decir, se puede controlar 10 cargas como máximo.

Hora de inicio: es el horario que se encuentra en el Text3 del Frame Horario.

Señal de fin: es el horario que se encuentra en el Text4 del Frame Horario

"d": éste carácter representa un comando en el que se indica que la información contenida se trata del horario de inicio y fin de una determinada carga.

a) En el PDA

Si se realiza una modificación en el horario de inicio o de fin del cuadro de texto en el PDA, esta información encapsula los horarios en el formato dado y se debe dar un click el botón GUARDAR, para enviar dichos datos.

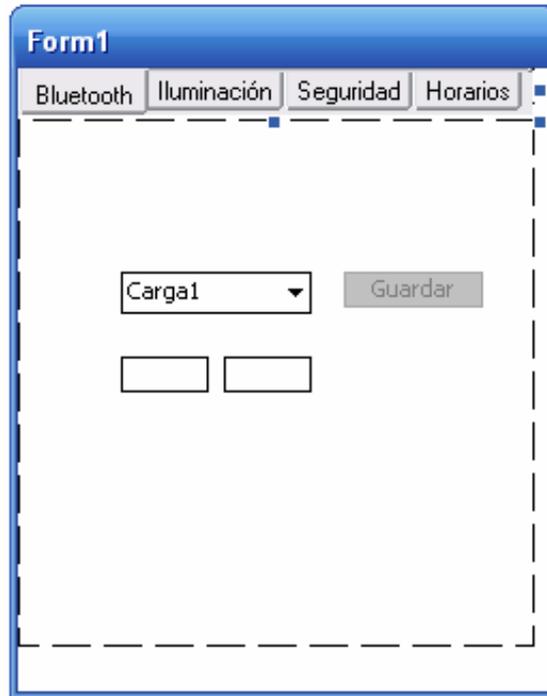


Figura 3.43. Pestaña Horarios

```
Private Sub Text3_Change()           'si existe algún cambio en la hora de inicio
    btnguardar.Enabled = True       'se activa el botón "guardar"
    Text1.Text = cmbhorario.ListIndex 'y se cambia los valores de la trama
    Text1.Text = Text1.Text + Text3.Text + Text4.Text + "d"
End Sub                             'finaliza la subrutina
```

```
Private Sub Text4_Change()          ' si existe algún cambio en la hora de fin
    btnguardar.Enabled = True       ' se activa el botón "guardar"
    Text1.Text = cmbhorario.ListIndex ' y se cambia los valores de la trama
    Text1.Text = Text1.Text + Text3.Text + Text4.Text + "d"
End Sub                             'finaliza la subrutina
```

Para el proceso GUARDAR se requiere a la subrutina chequeo, luego se pregunta si en el texto está la información "none", en caso de que esto suceda, se envía tanto a la hora de inicio como a la hora de fin la información "24:00".

```
Private Sub btnguardar_Click()      'si se presiona el botón guardar
```

Call chequeo	<i>'se llama a la subrutina chequeo</i>
If Text3.Text = "none " Or Text4.Text = "none " Then	<i>'se pregunta si no hay datos</i>
Text1.Text = cmbhorario.ListIndex	<i>'si no los hay, se genera una nueva</i>
Text1.Text = Text1.Text + "24:0024:00d"	<i>'trama con los valores de 24:00</i>
End If	<i>'finaliza la condición</i>
Call enviar	<i>'envía la trama</i>
btnguardar.Enabled = False	<i>'deshabilita el botón "guardar"</i>
End Sub	<i>'finaliza la subrutina</i>

b) En el microcontrolador

El microcontrolador verifica si en el comando recibido como último carácter se tiene la letra "d", con lo que se obtiene la información de hora de inicio y de fin para actualizar dicha información en su memoria interna, identificando previamente a que carga corresponde.

If Right(atin , 1) = "d" Then	<i>'si se recibe una "d" como último carácter.</i>
Dim Hm As String * 2	<i>'se inicializa las variables a utilizarse</i>
Dim Ti As Integer	
Hm = Mid(atin , 1 , 1)	<i>'se obtiene el número de carga</i>
Ti = Val(hm)	<i>'el valor se lo convierte a entero</i>
Ti = Ti * 2	<i>'se calcula la posición del dato</i>
Ti = Ti + 1	
Hm = Mid(atin , 2 , 2)	<i>'se extrae la hora de inicio.</i>
Hi(ti) = Val(hm)	<i>'se guarda las horas.</i>
Hm = Mid(atin , 5 , 2)	<i>'se extrae la hora de inicio.</i>
Hi(ti + 1) = Val(hm)	<i>'se guarda los minutos</i>
Hm = Mid(atin , 7 , 2)	<i>'se extrae la hora de fin.</i>
Hf(ti) = Val(hm)	<i>'se guarda las horas</i>
Hm = Mid(atin , 10 , 2)	<i>'se extrae la hora de fin.</i>
Hf(ti + 1) = Val(hm)	<i>'se guarda los minutos.</i>
End If	<i>'finaliza la condición</i>

Al momento en que el usuario dé por terminada la comunicación entre el PDA y la pasarela, cerrando la aplicación de control contenido en el PDA, el microcontrolador entra en el modo de simulación.

```

Do                                     'inicia el lazo
  Locate 2 , 4                          'posiciona el LCD
  Lcd Time$                             'muestra la hora actual
  Dim Ju As Integer                      'inicializa la variable Ju la que se
  Ju = 1                                 'encarga de almacenar el número de carga
  Call Simulacion(hi(1) , Hi(2) , Hf(1) , Hf(2) , Ju) 'se llama a la subrutina simulación
  Ju = 2                                 'para realizar el control de los horarios
  Call Simulacion(hi(3) , Hi(4) , Hf(3) , Hf(4) , Ju) 'de todas las cargas
  Ju = 3                                 'en estas líneas el código se muestran solo
  Call Simulacion(hi(4) , Hi(5) , Hf(4) , Hf(5) , Ju) 'para tres cargas.
  $timeout = 200000                      'genera un tiempo de espera de datos.
  Input Atin Noecho                       'ingreso de datos
Loop Until Atin = "H"                    'haga esto hasta que se inicialice la
                                         'sincronización

```

El modo de simulación no es más que una subrutina, en la cual se realiza una comprobación del horario contenido en cada una de las cargas con la hora actual proporcionado por el Real Time del microcontrolador, esto permite encender o apagar una determinada carga en el horario programado con el PDA.

```

Sub Simulacion(hi1 As Integer , Hi2 As Integer , Hf1 As Integer , Hf2 As Integer , Carg As Integer)
                                         'inicializa la subrutina simulación
Dim Bt As Integer                        'se inicializa la variable Bt
If Hi1 <> 24 Then                          'Pregunta si es la hora es diferente a 24
                                         '24 significa que no se ingresó horario
  Atin = Mid(time$ , 1 , 2)              'se ingresa a la variable Atin el valor
                                         'de la hora actual
  If Hf1 > Val(atin) And Hi1 < Val(atin) Then 'si la hora está dentro del intervalo

```

```

Bt = 1                                     'ingresado por el usuario se llama
Call Carga(carg , Bt)                     'a la subrutina carga.
Else                                       'sino
If Hi1 = Val(atin) Then                   'se pregunta si la hora es igual a la
If Hf1 = Val(atin) Then                   'almacenada en memoria
Atin = Mid(time$ , 4 , 2)                 'se chequea todos los casos
If Hi2 <= Val(atin) And Hf2 > Val(atin) Then 'para el encendido se llama a la
Bt = 1                                     'subrutina carga con Bt=1
Call Carga(carg , Bt)
End If
Else
Atin = Mid(time$ , 4 , 2)                 'se extrae los minutos
If Hi2 <= Val(atin) Then                 'se pregunta si los minutos son iguales
Bt = 1                                     'a los almacenados en memoria
Call Carga(carg , Bt)
Else
Bt = 0                                     'para el apagado se llama a la
Call Carga(carg , Bt)                   'subrutina carga con Bt=0
End If                                   'finaliza la condición
End If                                   'finaliza las condición
End Sub                                   'finaliza la subrutina.

```

La subrutina *Simulación* utiliza una segunda subrutina denominada *Carga* que es la encargada de leer el estado actual de la carga y efectuar la acción para la que fue programada, es decir, encender o apagar dicha carga.

```

Sub Carga(c As Integer , Num As Integer)  'se inicializa la subrutina carga
                                           'si Num es 1 significa que la carga se
If Num = 1 Then                            'va a encender
If C = 1 Then                              'la variable C nos indica el # de carga
If Pine.7 = 1 Then                         'se verifica el estado actual de la carga

```

If Porta.0 = 1 Then	<i>'si es 1 significa que esta apagado</i>
Porta.0 = 0	<i>'para luego invertir el valor del bit.</i>
Else	<i>'de salida si es 1 se cambia a 0 y</i>
Porta.0 = 1	<i>'viceversa</i>
End If	<i>'finaliza la condición</i>
End If	<i>'finaliza la condición</i>
End If	<i>'finaliza la condición</i>
Else	<i>'caso contrario</i>
If Num = 2 Then	<i>'si Num es 2 significa que la carga</i>
If C = 1 Then	<i>'se va a apagar</i>
If Pine.7 = 0 Then	<i>'se verifica el estado de la carga</i>
If Porta.0 = 1 Then	<i>'si es 0 significa que esta</i>
Porta.0 = 0	<i>'encendido para apagarlo</i>
Else	<i>'se invierte el estado del bit de salida</i>
Porta.0 = 1	<i>'si es 1 se cambia a 0 y viceversa</i>
End If	<i>' finaliza la condición</i>
End If	<i>' finaliza la condición</i>
End If	<i>' finaliza la condición</i>
End If	<i>' finaliza la condición</i>
End Sub	<i>' finaliza la subrutina</i>

CAPÍTULO 4

PRUEBAS Y RESULTADOS

4.1. PRUEBAS CON REFERENCIA AL ALCANCE

4.1.1. CONDICIÓN DE PRUEBAS

Las pruebas fueron desarrolladas en una vivienda de las siguientes características:

- Ubicada en la ciudad de Ambato, Barrio Miñarica I, pasaje Tirso de Molina N° 150.
- El área de construcción es de 73,64 metros cuadrados.
- La vivienda consta de dos plantas, con terraza.
- Las paredes son de ladrillo.
- Puertas interiores de madera.
- Portón principal metálico.
- Consta de un jardín frontal y un patio trasero.

La pasarela domótica se ubicó en la parte central de la segunda planta a la misma que se conectó un foco, el cual permitió conocer si existe comunicación con el PDA.

4.1.2. ESTABLECIMIENTO DE LA CONEXIÓN

La prueba se basa en comprobar que desde diferentes puntos se pueda realizar correctamente la comunicación y el proceso de sincronización entre el PDA y la pasarela.

Para la realización de las pruebas se tomó como referencia diferentes radios de cobertura de la señal por cada planta. Para esto, se colocó el PDA en distintos puntos de la segunda planta (ver Figura 4.1), y se realizó la prueba por cinco ocasiones en cada punto, con lo que se obtuvo los siguientes resultados (ver Tabla 4.1):

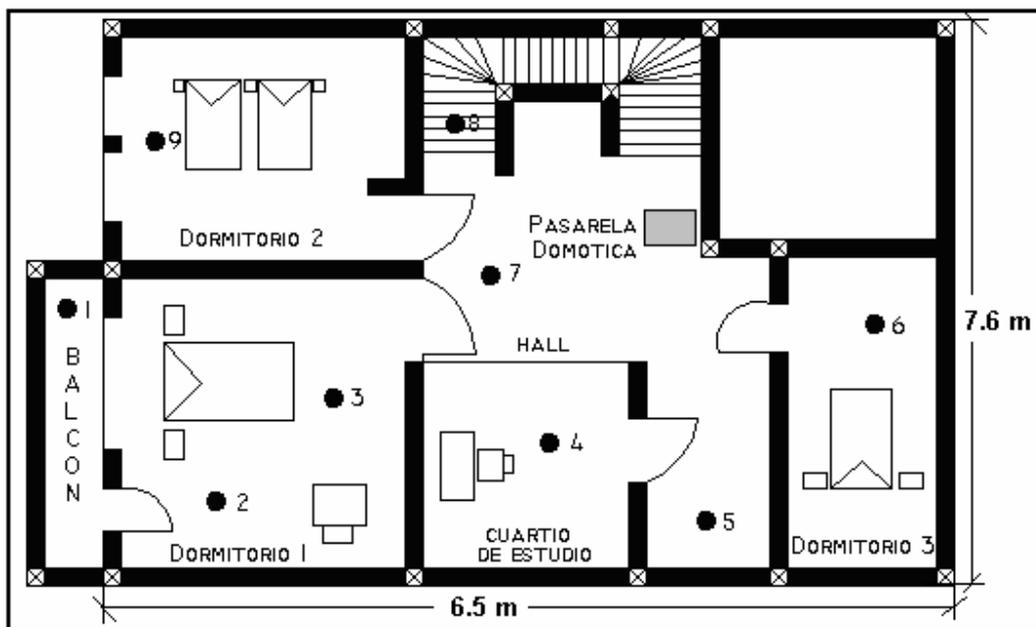


Figura 4.1. Segunda planta (Escala 1:100)

Ubicación	Distancia (metros)	Obstáculo	Porcentaje exitoso (%)
-----------	--------------------	-----------	------------------------

Punto 1	8.1	2 Paredes	100
Punto 2	7.0	1 Pared	100
Punto 3	5.0	1 Pared	100
Punto 4	3.4	1 Pared	100
Punto 5	3.8	Ninguno	100
Punto 6	2.7	1 Puerta	100
Punto 7	2.3	Ninguno	100
Punto 8	3.0	Gradas	100
Punto 9	6.7	Pared+Puerta	100

Tabla 4.1. Establecimiento de la conexión, segunda planta

De manera similar las pruebas fueron realizadas en la primera planta (Figura 4.2), teniendo en cuenta que en ningún caso fue reubicada la pasarela domótica, en relación a las pruebas que se realizaron en la segunda planta.

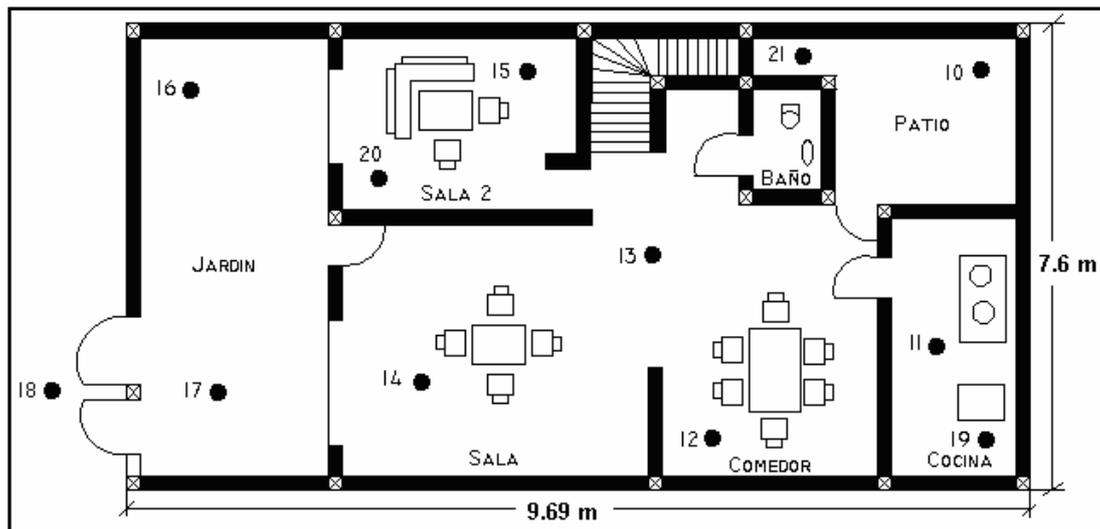


Figura 4.2. Primera planta (Escala 1:100)

El resultado de estas pruebas se muestra en la Tabla 4.2.

Ubicación	Distancia (metros)	Obstáculo	Resultado
Punto 10	4.8	Loza	100
Punto 11	4.7	Loza	100
Punto 12	4.4	Loza	100
Punto 13	2.1	Loza	100
Punto 14	5.7	Loza	100
Punto 15	4.1	Loza	100
Punto 16	8.9	Loza	80
Punto 17	8.3 P. metal	Abierta	100
		Cerrada	60
Punto 18	10.4 P. metal	Abierta	100
		Cerrada	20
Punto 19	6.3	Loza	60
Punto 20	5.5	Loza	100
Punto 21	4.6	Loza	80

Tabla 4.2. Establecimiento de la conexión, primera planta

Para tener una buena conexión Bluetooth hay que tener en cuenta los siguientes aspectos: que los dispositivos Bluetooth se encuentren dentro del radio de cobertura de comunicación, asimismo uno de los aspectos importantes es el número y el material de los obstáculos que debe atravesar la señal. La señal puede, o no atravesar un obstáculo, esto depende del tipo de material que está hecho el obstáculo. El metal es uno de los materiales que impide que una señal pueda traspasar fácilmente, además provoca que la señal rebote.

Debido a que la loza de la vivienda está construida por varillas, y la distancia de ciertos puntos a la que se realizó las pruebas no se pudo obtener una conexión 100% segura en todos los puntos.

Además, se realizó pruebas con línea de vista, obteniendo distancias de hasta 45 metros, estas distancias van cambiando conforme a los obstáculos que se presentan y el tipo de material de estos.

4.1.3. MANTENCIÓN DE LA CONEXIÓN

Esta prueba se basa en que desde el instante que se realice la comunicación entre el PDA y la pasarela no se cierre la aplicación del Sistema Domótico (no se dé por terminada la comunicación).

Ubicación	Distancia (metros)	Obstáculo	Resultado
Punto 1	8.1	2 Paredes	OK*
Punto 2	7.0	1 Pared	OK*
Punto 3	5.0	1 Pared	OK*
Punto 4	3.4	1 Pared	OK*
Punto 5	3.8	Ninguno	OK*
Punto 6	2.7	1 Puerta	OK*
Punto 7	2.3	Ninguno	OK*
Punto 8	3.0	Gradas	OK*
Punto 9	6.7	Pared+Puerta	OK*

Tabla 4.3. Mantenimiento de la conexión, segunda planta

* El resultado de conectividad fue exitoso, encender, apagar y conocer el estado actual del foco.

Para este ensayo se tomó como referencia los mismos puntos de la prueba de establecimiento de conexión; encendimos y apagamos el foco, ubicado en la segunda planta (Figura 4.1), con lo que obtuvimos los siguientes resultados (Tabla 4.3).

De manera similar las pruebas fueron realizadas en la primera planta (Figura 4.2), teniendo en cuenta que en ningún caso fue reubicada la pasarela domótica, en relación a las pruebas que se realizaron en la segunda planta.

El resultado de dichas pruebas se lo muestra a continuación:

Ubicación	Distancia (metros)	Obstáculo	Resultado
Punto 10	4.8	Loza	OK
Punto 11	4.7	Loza	OK
Punto 12	4.4	Loza	OK
Punto 13	2.1	Loza	OK
Punto 14	5.7	Loza	OK
Punto 15	4.1	Loza	OK
Punto 16	8.9	Loza	OK
Punto 17	8.3 P. metal	Abierta	OK
		Cerrada	NO
Punto 18	10.4 P. metal	Abierta	OK
		Cerrada	NO
Punto 19	6.3	Loza	OK
Punto 20	5.5	Loza	OK
Punto 21	4.6	Loza	OK

Tabla 4.4. Mantenión de la conexión primera planta

OK El resultado del mantenimiento de la conexión fue exitoso al, encender, apagar y conocer el estado actual de la carga (foco).

NO En estos puntos no se logró controlar el estado del foco, por lo que no hubo una transmisión correcta de los datos. Debido a que el metal es uno de los materiales que dificulta tener una buena comunicación Bluetooth.

Para evitar este problema se podría colocar en la primera planta del domicilio, otro dispositivo Bluetooth, el cual realice la función de repetidor, con lo que se tendría mayor cobertura.

4.2. PRUEBAS DEL SISTEMA DOMÓTICO

Para las pruebas de la aplicación del Sistema Domótico, se lo ha dividido en tres partes: iluminación, seguridad y horario.

4.2.1. ILUMINACIÓN

La Pantalla Iluminación tiene dos funciones específicas: mostrar el estado actual de todas las cargas eléctricas, así como también el control de cada una de ellas.

4.2.1.1. Estado de la carga

Al ingresar a la pantalla "*Iluminación*", se observa el estado actual de las diferentes cargas eléctricas (focos).

Para esta prueba se conectaron diez focos a la pasarela, de los cuales se encendieron los siguientes: Hall_2, cocina_1, dormitorio_2, jardín_1, mediante los interruptores respectivos. Entonces, al momento de ingresar a la pantalla "*Iluminación*" se obtuvo el siguiente resultado (Figura 4.3).

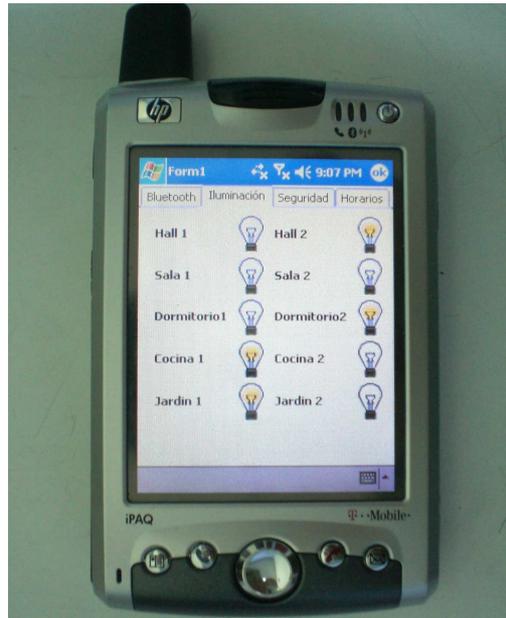


Figura 4.3. Lectura del estado actual de las cargas

El resultado que se obtuvo fue el esperado, los focos que estuvieron encendidos son representados con la imagen de un foco de color amarillo.

4.2.1.2. Control de las cargas eléctricas

Esta prueba se basa en ir controlando las diferentes cargas eléctricas desde la Pocket PC así como también mediante el interruptor.



Figura 4.4.Control de las cargas eléctricas

Al encender un foco desde la Pocket PC se muestra la imagen del foco de color amarillo, y al apagarlo se muestra una imagen de un foco de color blanco.

Y por último desde el mando manual se vuelve a intercambiar el estado de las cargas; los focos apagados se los enciende y viceversa (Figura 4.5).



Figura 4.5. Control de las cargas desde el mando manual

De los resultados obtenidos se concluye que las opciones que brinda la pantalla *Iluminación* trabajan correctamente.

4.2.2. SEGURIDAD

El funcionamiento de la opción *Seguridad* se basa en dos modos de trabajo: el primero muestra el estado actual de los diferentes sensores (activo / inactivo) a través de la Pocket PC, en la segunda opción se habilita la alarma.

4.3.1.1. Muestreo de sensores

Al ingresar a la pantalla *Seguridad* se indica los sensores que se encuentran activos en ese instante, para esta prueba se activaron los sensores 1, 3, 5,

magnético, humo y movimiento, respectivamente. Se ingresó a la opción “Seguridad” y se observó la siguiente pantalla (Figura 4.6.)



Figura 4.6. Visualización del estado actual de los sensores

Las esferas de color verde nos indican que el sensor se encuentra activado, mientras que las esferas rojas, que está desactivado. Con lo que se comprobó que la pantalla “Seguridad” trabaja correctamente.

4.3.1.2. Habilitación de la alarma

Para la utilización de esta opción se presiona la opción *Habilitar* en caso de no estar previamente activa.

Para efectuar esta prueba se cerró la aplicación Sistema Domótico, y se activó uno de los sensores que posteriormente fue desactivado, con lo que comenzó a sonar la sirena durante 30 segundos.

A continuación se mantuvo activo el sensor magnético, lo que produjo que se accione la sirena de manera indefinida hasta que la alarma sea desactivada, para lo cual se debe restablecer la comunicación entre la Pocket PC y la pasarela.

4.2.3. HORARIO

La opción *Horario*, basa su funcionamiento en el almacenamiento de los horarios de inicio y fin, los que se ingresan a través de la Pocket PC, luego de lo cual se activan automáticamente cuando el usuario cierra la comunicación con la pasarela y cuando cumple el intervalo ingresado.

4.2.3.1 Almacenamiento del Horario

Para esta prueba se ingresó caracteres de todos los tipos, los mismos que no fueron necesariamente números, obteniendo como resultado, que si no se cumple el estándar fijado (hh:mm), el sistema restringe la hora y no muestra el carácter presionado, además si no se ingresa completo el horario, aparece en el cuadro de texto el mensaje “none” (Figura 4.7).



Figura 4.7. Horarios ingresados incorrectos

Luego se procedió a guardar esta información presionando el botón *Guardar*, con lo cual en todos los casos se obtuvo que los datos fueron almacenados correctamente.

4.2.3.1 Proceso de Simulación

Luego de cerrar la aplicación, observamos que las cargas se encienden como lo planeado en el intervalo de tiempo programado. Los horarios programados se presentan en la Tabla 4.5.

Carga	Hora inicio	Hora fin	Resultado
1	18:57	18:59	OK*
2	20:10	21:05	OK*
3	23:55	00:05	NO**
4	1:10	2:00	OK*
5	8:00	10:00	OK*
6	12:00	12:00	NO**
7	13:50	14:15	OK*
8	16:25	17:00	OK*
9	19:45	20:35	OK*
10	7:00	13:00	OK*

Tabla 4.5. Horarios programados

*Del horarios programado se obtuvo una simulación exitosa.

**Como los horarios ingresados no cumplieron con el formato establecido, los datos no fueron almacenados por lo que no se realizó la simulación (La hora de encendido no puede ser mayor o igual a la hora de apagado).

Para esta prueba además se apagó el foco en el horario que fue configurado para que se encuentre encendido, consiguiendo que el foco, luego de un tiempo que va de 200 ms. a 1 seg. se vuelva a encender.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se comprueba que la tecnología Bluetooth proporciona distancias de comunicación superiores a los diez metros ya que básicamente dependen de la clase de dispositivo que se esté utilizando, como ejemplo se puede alcanzar una comunicación con línea de vista hasta cien metros con dispositivos de clase 1 (Para esta prueba se utilizó dos dispositivos Bluetooth KC-11, de clase 1 y versión 1.2).
- La tecnología Bluetooth se diferencia de la comunicación Infrarroja puesto que para el envío de datos ésta necesita que sus dispositivos estén en línea de vista directa y a una distancia máxima de 15 metros, en cambio con tecnología Bluetooth no es necesario, pues para poder comunicarse entre dispositivos, la señal irradiada puede atravesar paredes, incluso la velocidad de transmisión supera enormemente a la infrarroja ya que puede llegar hasta 768 Kbp/s.
- El tipo de PDA que proporciona mejores servicios es el que ofrece una Pocket PC, ya que utiliza una plataforma Windows dándose a conocer considerablemente y la manipulación por parte del usuario resulta sencilla.

- La aplicación implementada en el PDA puede correr en distintos sistemas operativos de Windows Mobile, por ejemplo: Pocket PC 2002, Pocket PC 2003, Windows CE 5.0, entre otros.
- El sistema Domótico construido proporciona libertad de movimiento al usuario ya que el control se lo realiza de manera inalámbrica, y se lo puede realizar cuando el usuario lo requiera.
- El Sistema diseñado tiene la ventaja de ser completamente versátil y sencillo de utilizar, puesto que el usuario tiene la posibilidad de controlar una determinada carga, cuando se encuentra conectado al sistema o de dejar programado su funcionamiento en un determinado horario establecido por él.
- El sistema domótico es independiente del tipo de carga que se conecte a la salida de la pasarela, pues el hardware implementado brinda la posibilidad de control de diferentes tipos de cargas eléctricas ON/OFF, con un máximo de corriente de diez amperios.
- Para el proceso de conexión entre el PDA y la pasarela, el usuario debe encontrarse dentro del radio de cobertura de señal que proporcionan los dispositivos Bluetooth, evitando así problemas de comunicación.
- Se podría implementar un sistema de respaldo en el que, el usuario, desde un nodo central pueda manipular todas las opciones que brinda el Sistema Domótico.
- El Sistema podría interactuar con las tecnologías Domóticas más utilizadas como X-10, para realizar el control desde la pasarela hacia las diferentes cargas.

- El sistema da la posibilidad de actualizar el software tanto del PDA como del microcontrolador, con lo que se podría modificar la opción de simulación de presencia para que puedan ser programados el día y la hora en que se desea que una determinada carga funcione.
- La cobertura del sistema de control podría ser ampliada, con la implementación de varios módulos de Bluetooth y así formar una red de mayor alcance.
- Además, refiriéndonos a la parte de seguridad se podría implementar la opción del envío de mensajes SMS cuando se active algún sensor.
- También, con la utilización del chip TCP/IP W3100A y una IP pública, se puede ampliar la cobertura de control del Sistema Domótico por medio de Internet, con lo cual se puede controlar el dispositivo que el usuario requiera.

- El sistema es muy versátil, pues brinda la posibilidad de efectuar un sinnúmero de cambios para ampliar sus funciones y mejorar su sistema.
- El Sistema puede ser aplicado en departamentos y en domicilios de diferentes características, dependiendo de la ubicación en que se instale la pasarela, ya que su instalación no es compleja.
- El sistema puede ser implementado en lugares donde existan personas discapacitadas a quienes se les dificulte la posibilidad de moverse libremente o para simplemente facilitar el encendido o apagado de un foco.
- En la actualidad ciertos Hoteles brindan a sus clientes, que rentan una suite presidencial, un sistema que consta de un teclado por medio del cual se puede controlar de manera inalámbrica diferentes tipos de cargas eléctricas. Además, puede funcionar como un control remoto universal. El Sistema Domótico construido, brinda las mismas funciones anteriormente mencionadas, con la gran ventaja de que pueden ser controladas mediante un PDA que también permite el acceso a Internet, escuchar música, juegos, cámara, etc., aparte de ser versátil y sencillo de utilizar.
- Se puede implementar en un domicilio donde se requiera dar un toque de elegancia y confort, además de seguridad, ya que el hogar de una persona es siempre su refugio preferido. En él puede descansar tranquilamente y compartir momentos especiales con su familia.

5.2. RECOMENDACIONES

- Para que funcione la opción Simulación de presencia (pestaña Horarios), se debe dar por terminada la comunicación entre el PDA y la pasarela.

- Antes de apagar la Pocket PC, se recomienda cerrar la aplicación Sistema Domótico, pues al apagar el sistema operativo se desconectan los dispositivos inalámbricos, lo que puede ocasionar errores al reanudar la aplicación.

REFERENCIA BIBLIOGRÁFICAS

[1] BLUETOOTH SPECIAL INTEREST GROUP. "Bluetooth Core", Specification of the Bluetooth System, Versión 1.1, 22 de Febrero de 2003. Disponible en Internet: <URL:<http://www.bluetooth.com/dev/specifications.asp> >.

[2] BLUETOOTH SPECIAL INTEREST GROUP. "Bluetooth Profiles", Specification of the Bluetooth System, Versión 1.1, 22 de Febrero de 2003. Disponible en Internet: <URL:<http://www.bluetooth.com/dev/specifications.asp> >.

[3] M. Ali and G. J. Hayes, "Analysis of Integrated-F Antennas for Bluetooth Applications," 2000 IEEE-APS Conference on Antennas and Propagation for Wireless Communications, November 2000.

[4] R. Wansch, H. Humpfer, and J. Hupp, "An Integrated F-Antenna for Diversity Reception in a DECT Data Transmission Module," IEEE Antennas Propagation Society International Symposium, July 2000.

[5] MEZOE, Cambridge Consultants Ltd. BlueStack User Manual, 2001, p. 20

[6] BLUETOOTH SPECIAL INTEREST GROUP. Personal Area Networking Profile. Specification of the Bluetooth System, Versión 1.0, 14 de Febrero del 2003. Disponible en Internet: <URL:<http://www.bluetooth.com/dev/specifications.asp>>.

[7] <http://www.casadomo.com/noticiasDetalle.aspx>

[8] <http://es.wikipedia.org/wiki/Domotica>

[9] http://www.atmel.com/dyn/resources/prod_documents/doc2490.pdf

[10] <http://www.aulaclie.es/articulos/wifi.html>

[11] <http://www.tech-faq.com/lang/es/bluetooth.shtml>

ANEXO A

DIAGRAMAS DE FLUJOS

DIAGRAMA GENRAL DEL PDA

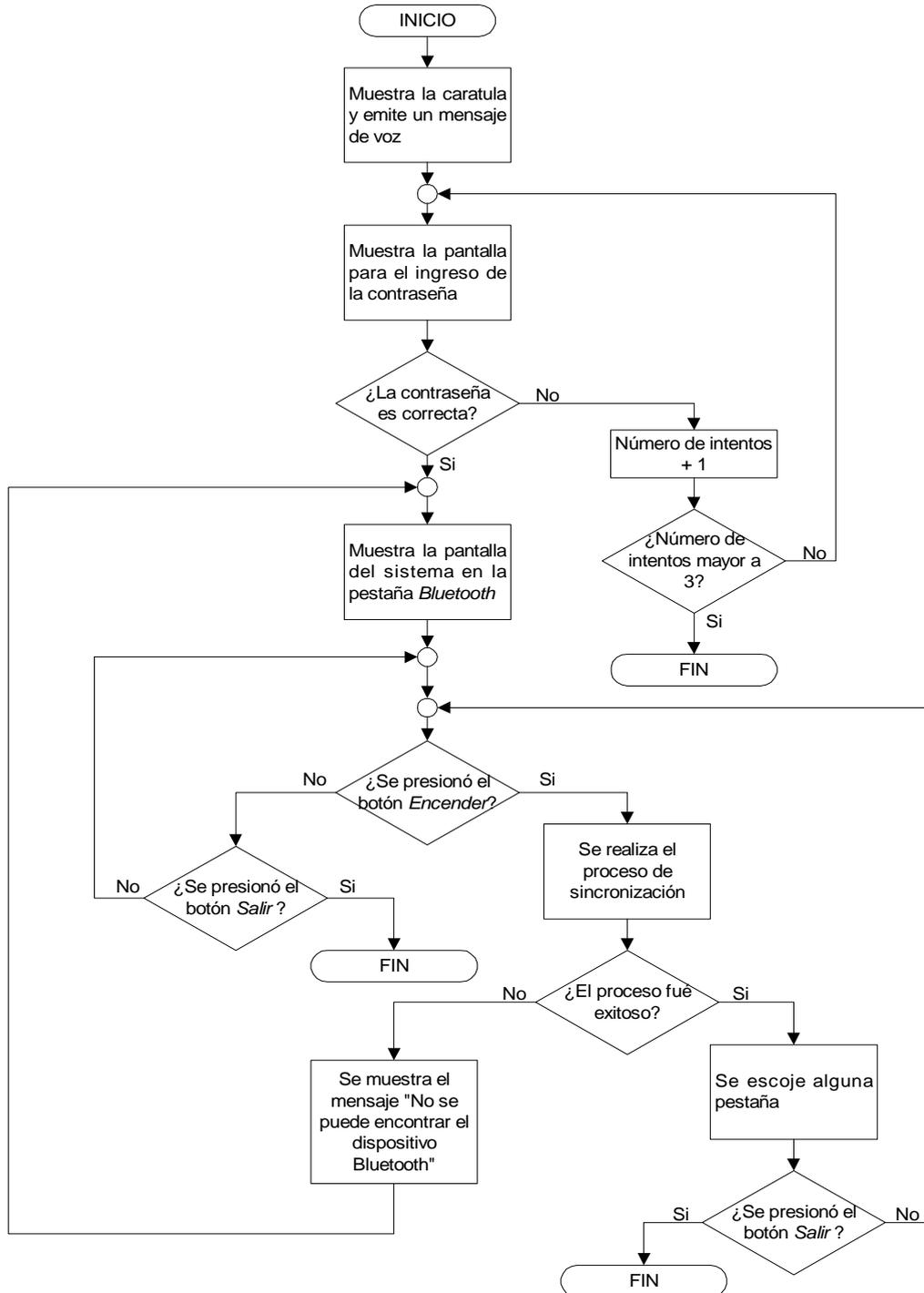


DIAGRAMA DEL PROCESO DE SINCRONIZACIÓN DEL PDA

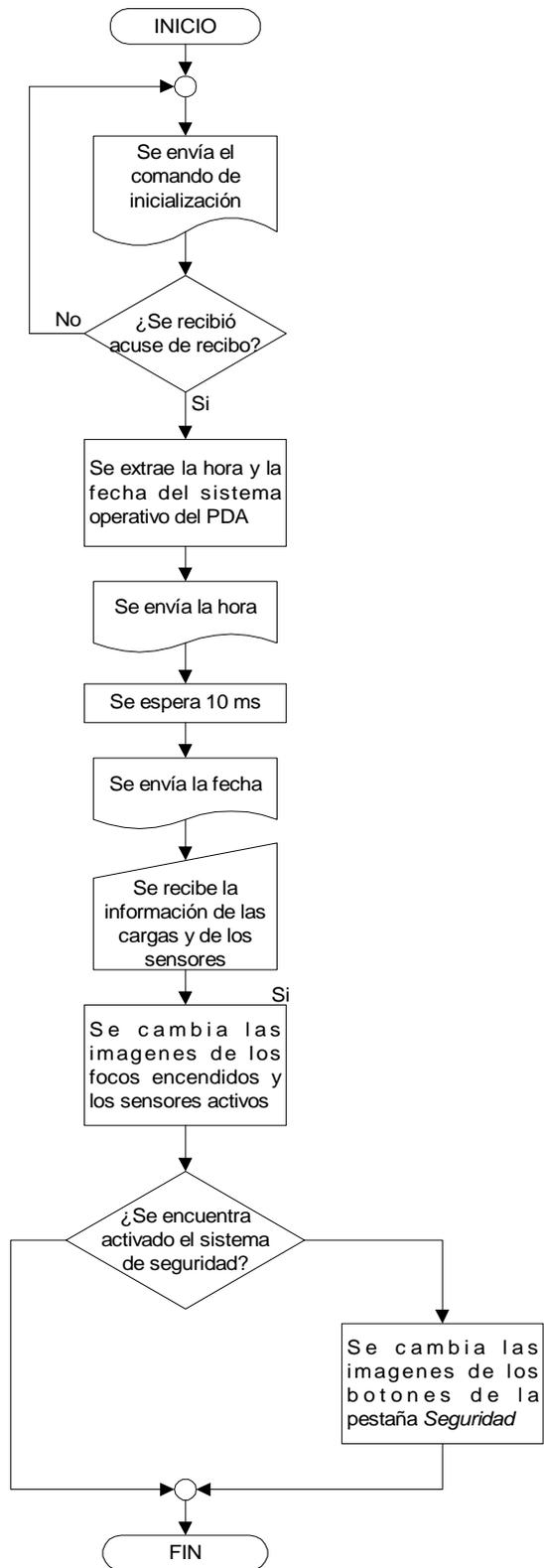


DIAGRAMA PARA LA PESTAÑA DE ILUMINACIÓN DEL PDA

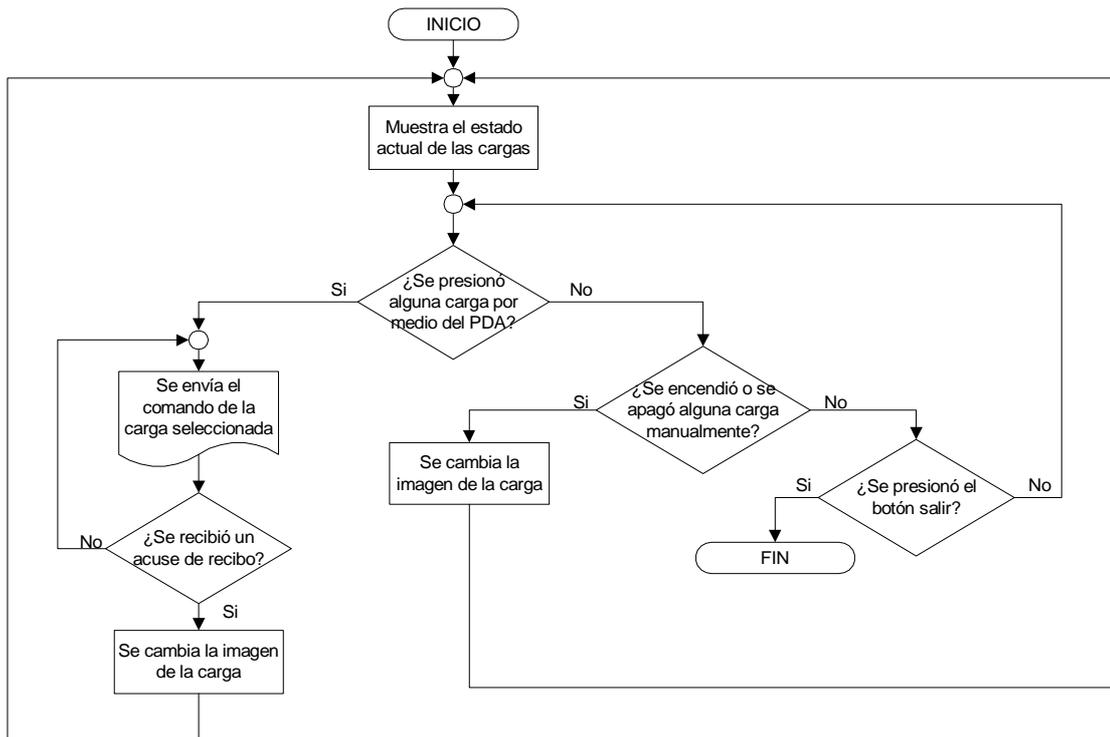


DIAGRAMA PARA LA PESTAÑA DE SEGURIDAD DEL PDA

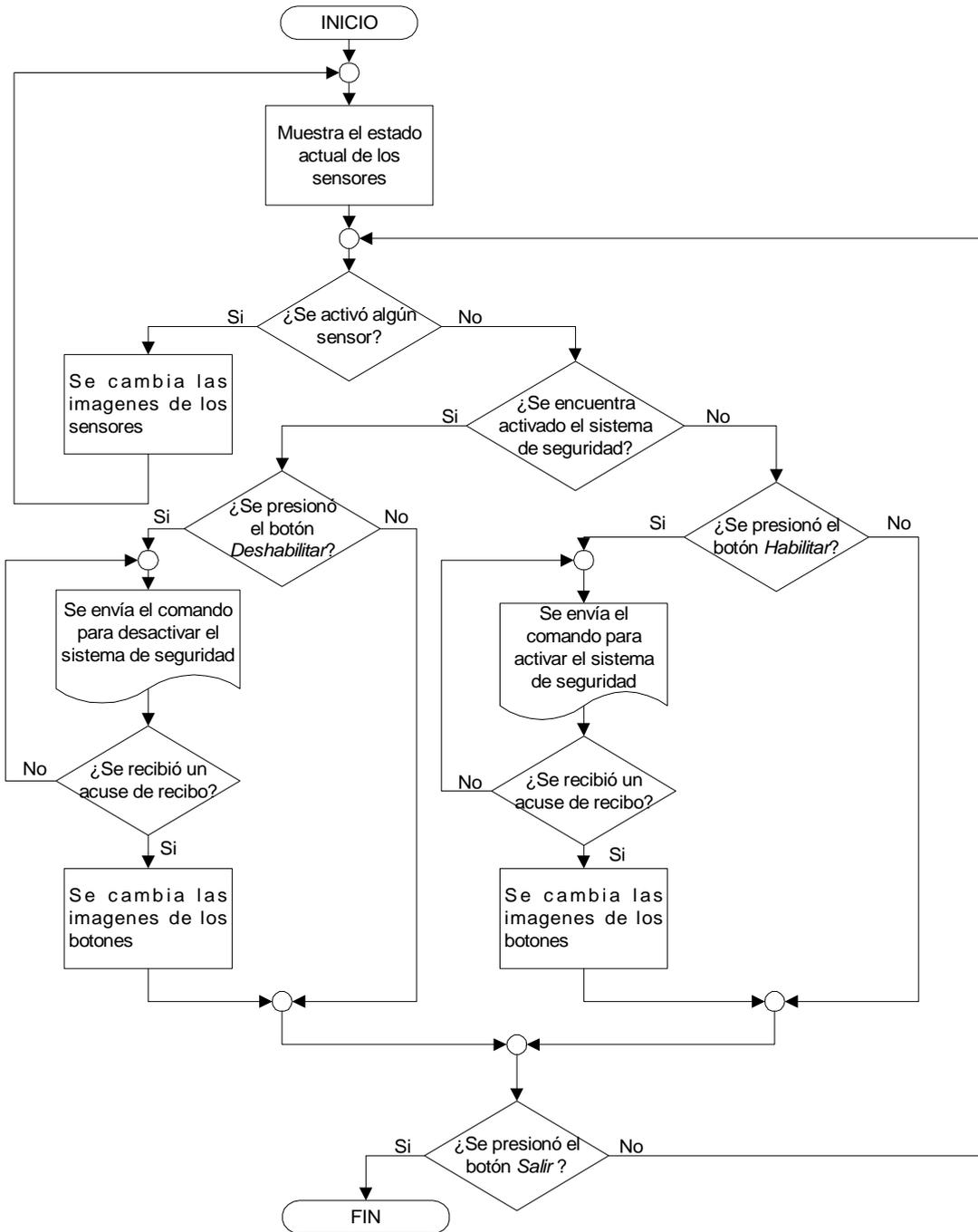


DIAGRAMA PARA LA PESTAÑA DE SEGURIDAD DEL PDA

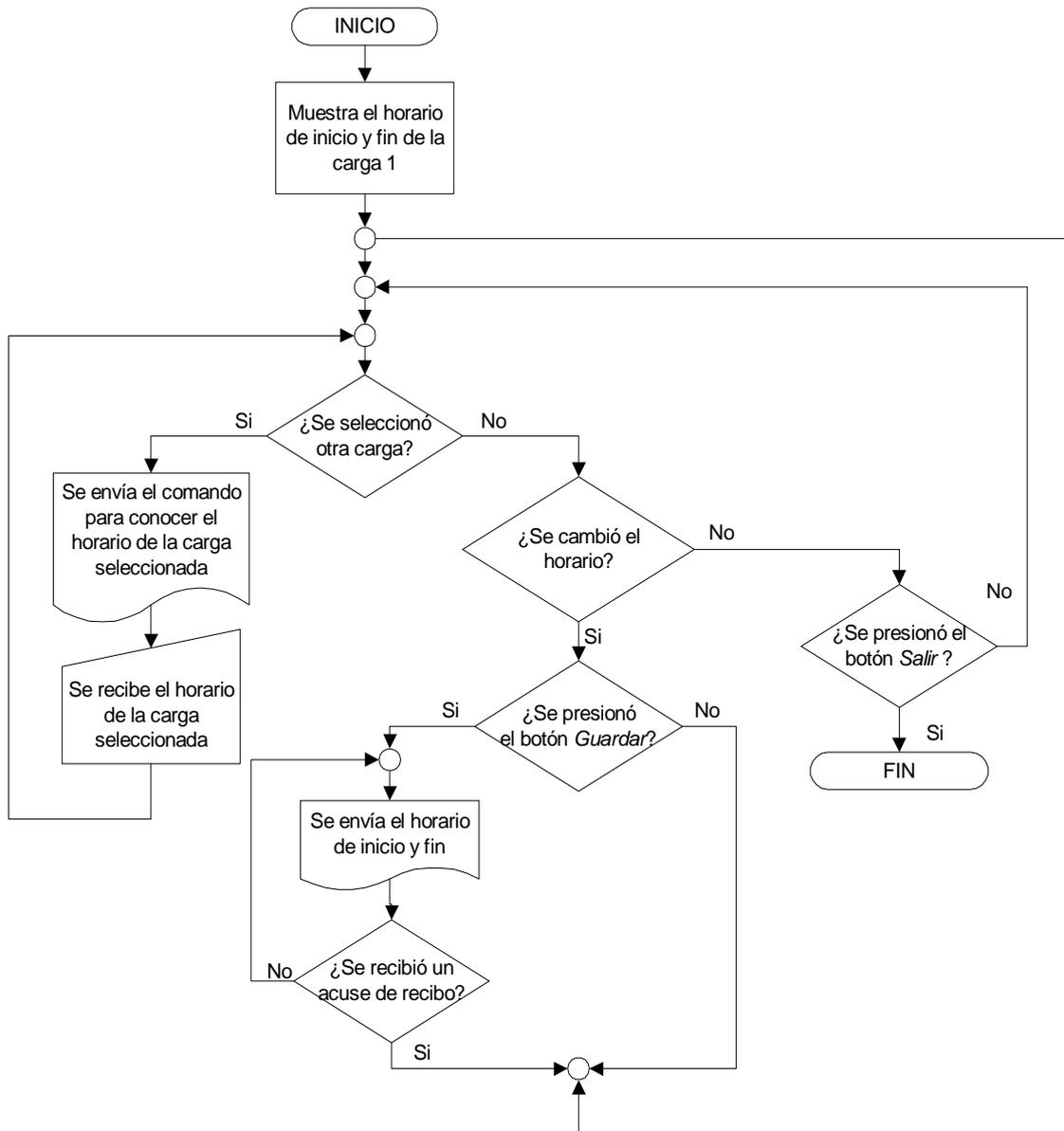


DIAGRAMA DEL MICROCONTROLADOR

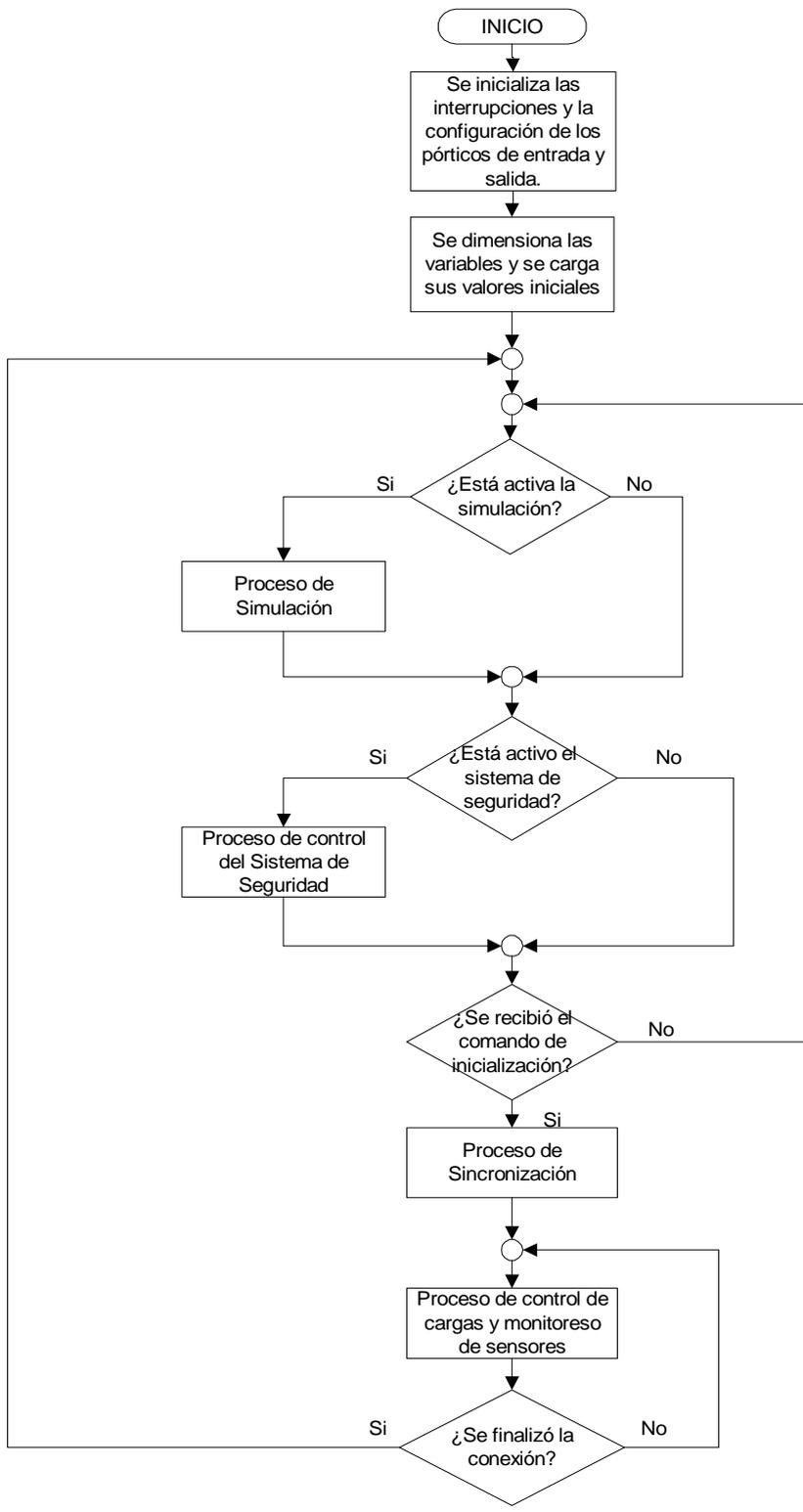


DIAGRAMA DEL PROCESO DE SINCRONIZACIÓN DEL MICROCONTROLADOR

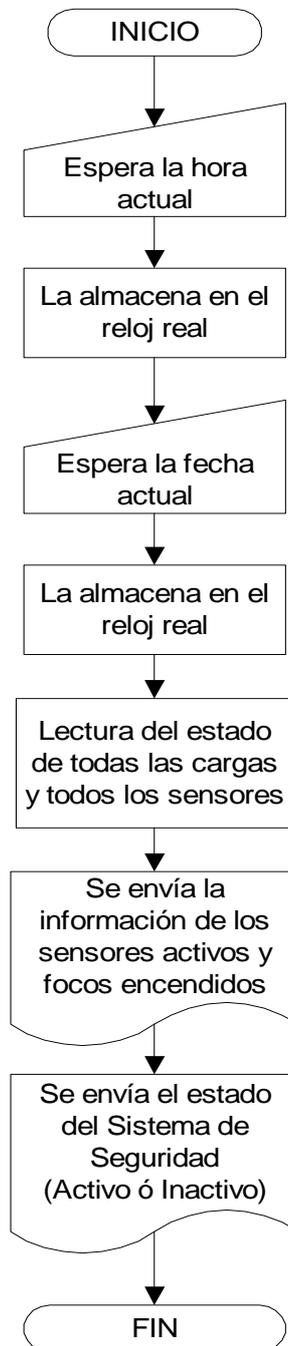


DIAGRAMA DEL SISTEMA DE CONTROL DEL MICROCONTROLADOR

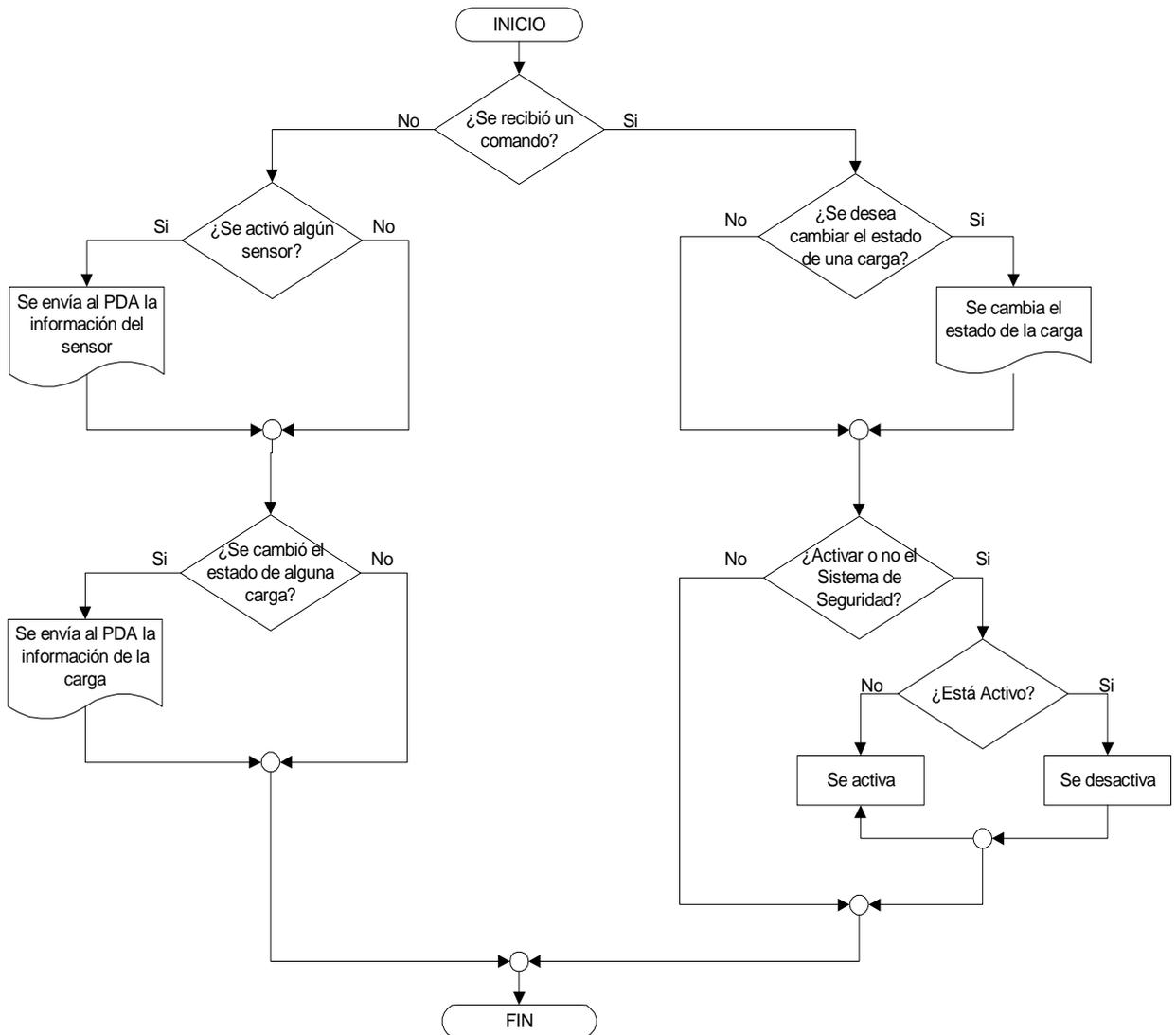


DIAGRAMA DEL SISTEMA DE SEGURIDAD DEL MICROCONTROLADOR

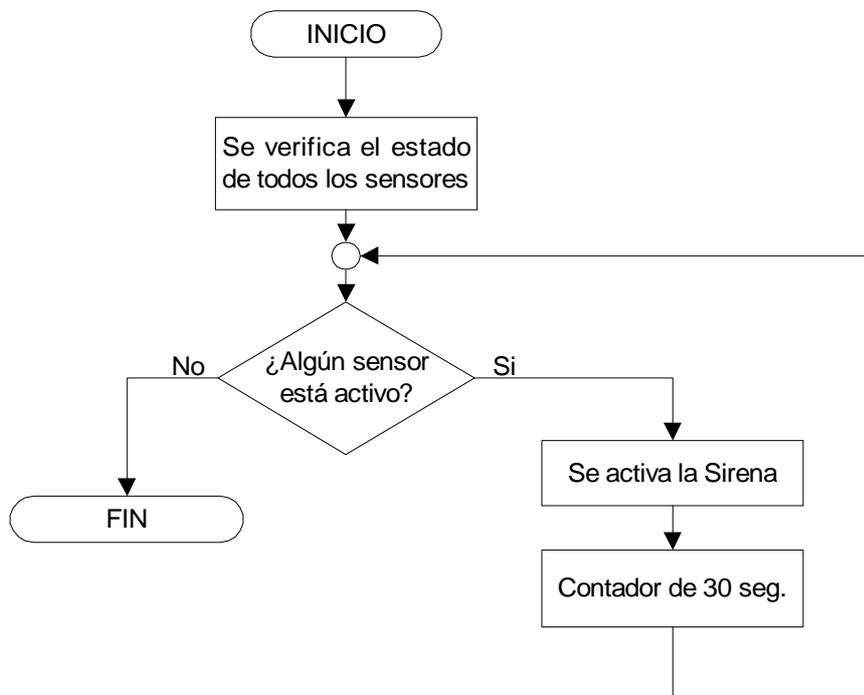
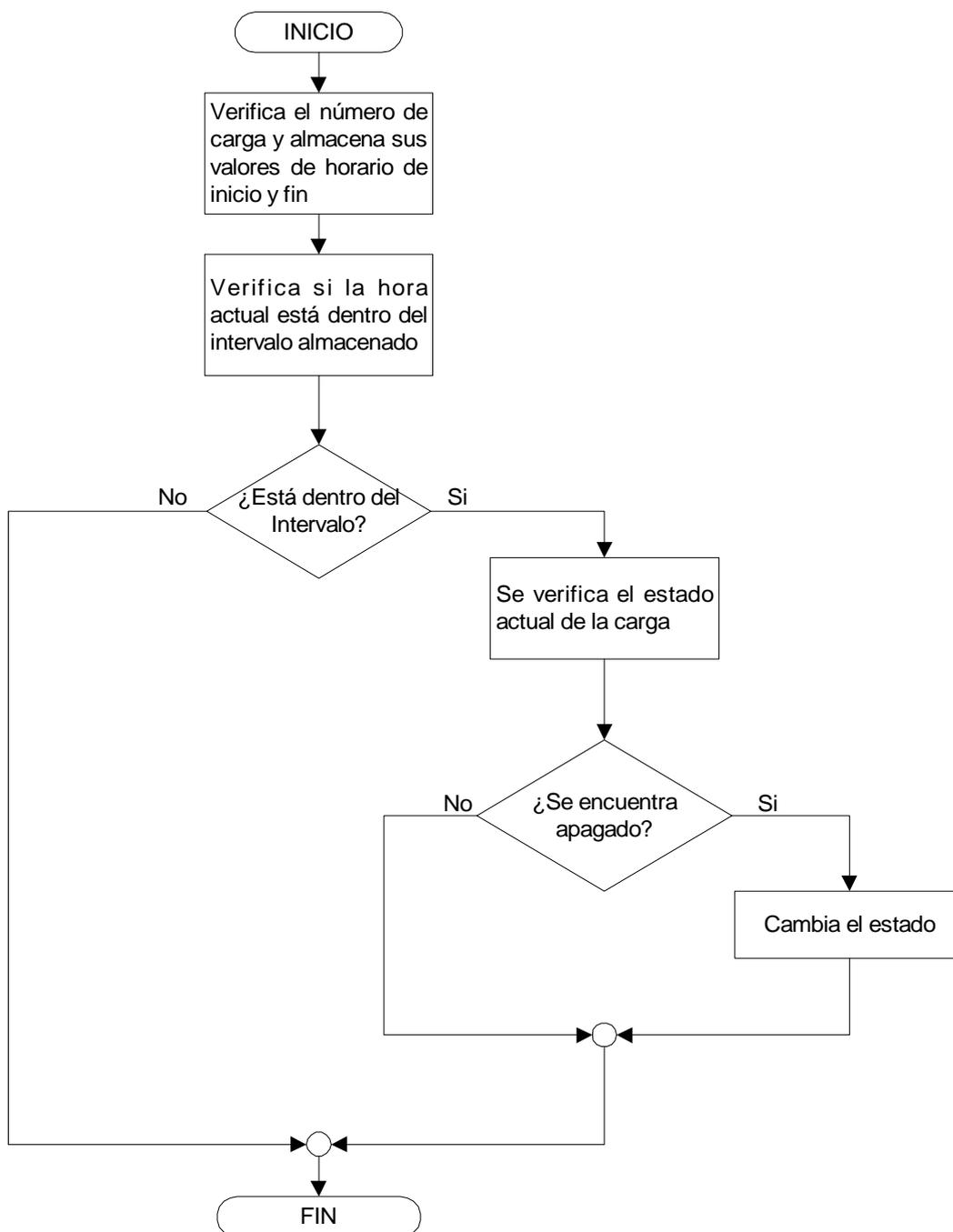


DIAGRAMA DEL PROCESO DE SIMULACIÓN DEL MICROCONTROLADOR

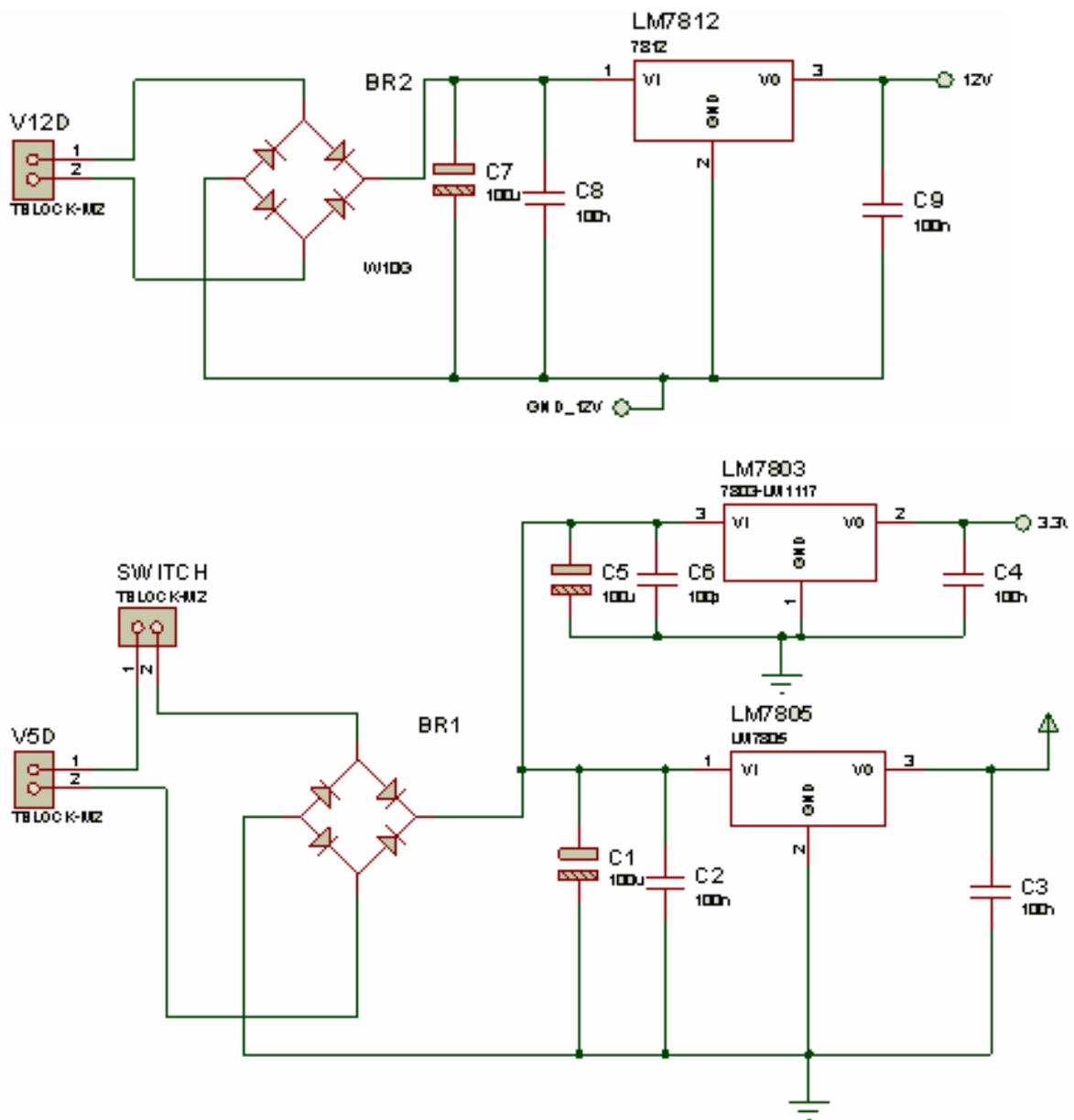


ANEXO B

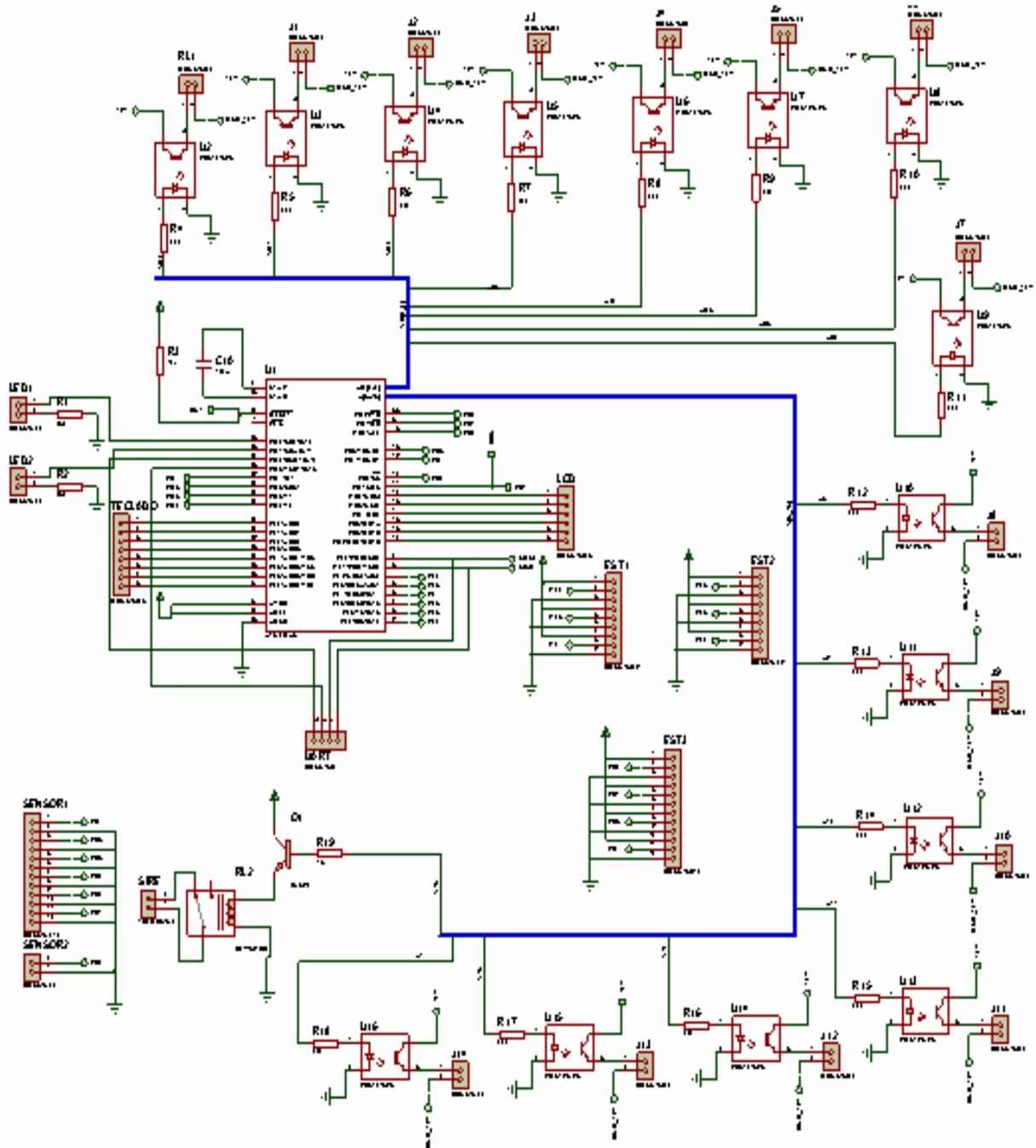
CIRCUITOS DEL SIETMA DOMÓTICO

CIRCUITOS ESQUEMÁTICOS

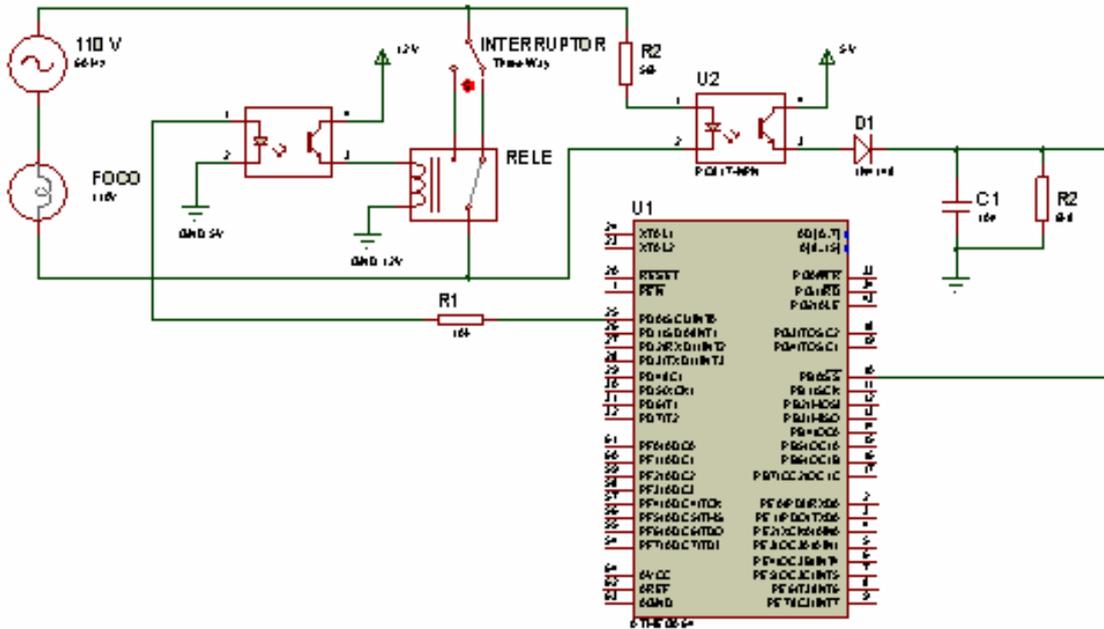
a) FUENTES DE PODER



b) CONTROL DE LA PASARELA



c) DETECTOR DE ESTADO DE LA CARGA



CIRCUITOS IMPRESOS

a) FUENTES DE PODER Y EL CONTROL DE LA PASARELA

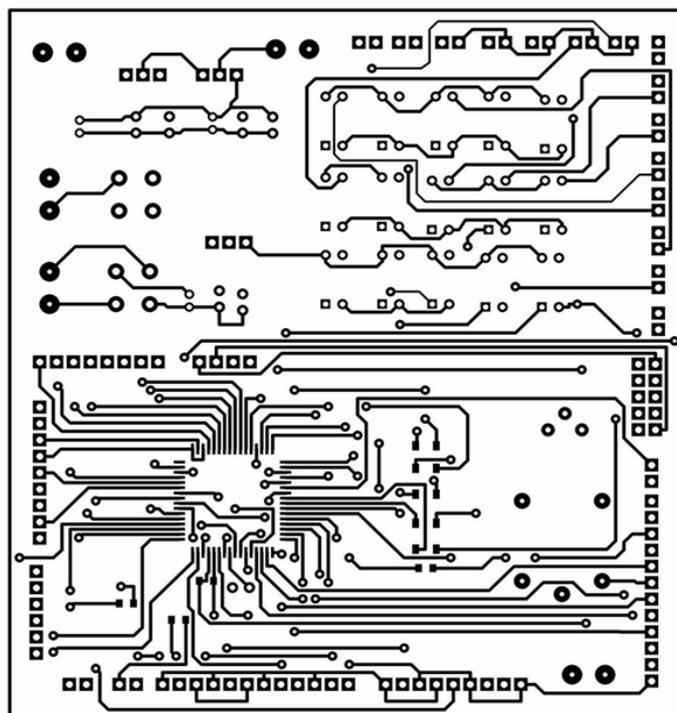


Figura B.2. Circuito impreso parte superior

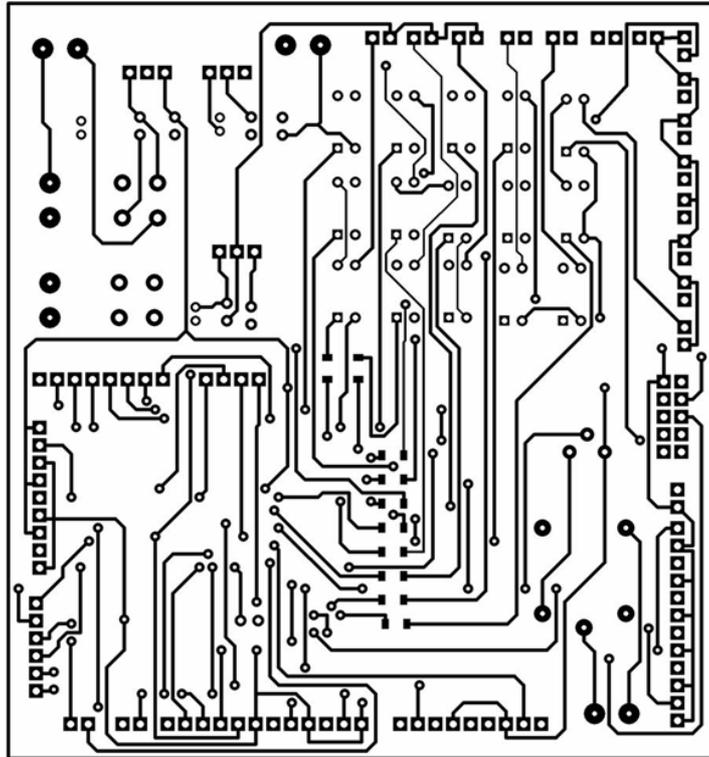


Figura B.3. Circuito impreso parte inferior

b) **DETECTOR DE ESTADO DE LA CARGA**

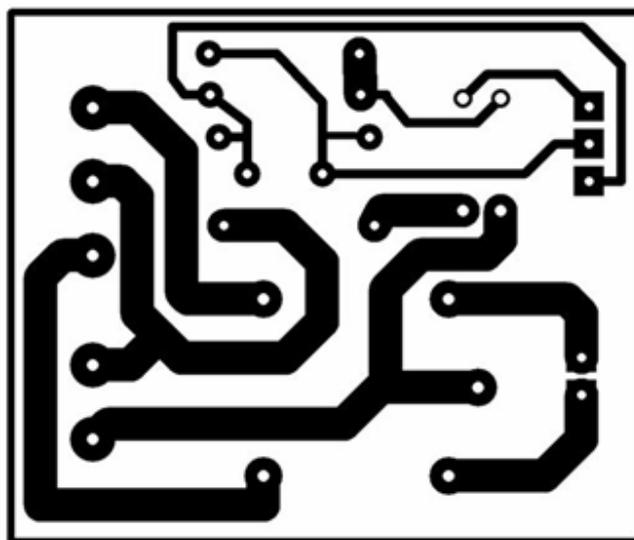


Figura B.5. Circuito impreso

DISTRIBUCIÓN DE ELEMENTOS

a) FUENTES DE PODER Y EL CONTROL DE LA PASARELA

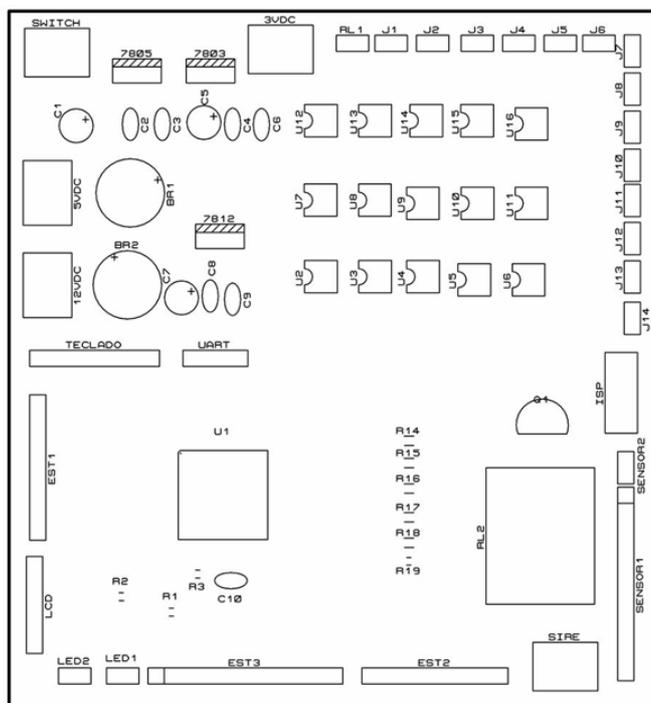


Figura B.1. Vista frontal de la placa de la pasarela.

b) DETECTOR DE ESTADO DE LA CARGA

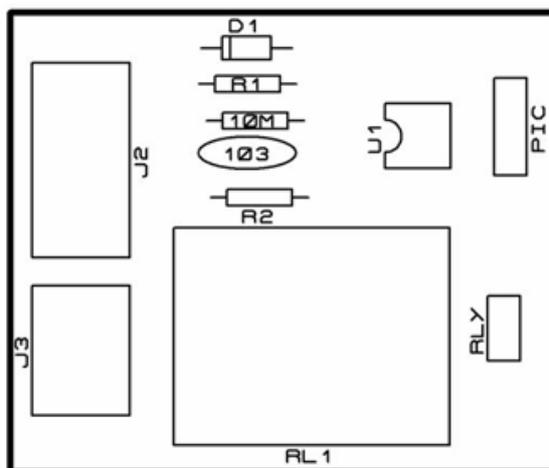


Figura B.4. Vista frontal de la placa de la carga

ANEXO C

HERRAMIENTA PARA PROGRAMACIÓN

BASCOM-AVR

El BASCOM-AVR© es un compilador de BASIC para la familia AVR de ATMEL, desarrollado por la empresa Holandesa MCS Electronics. Ha sido desarrollado sobre W95/98/NT y dispone de todas las características de la familia BASCOM.

CARACTERÍSTICAS DE BASCOM-AVR

- BASIC estructurado con etiquetas.
- Programación estructurada con sentencias IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- Generación de código máquina nativo en lugar de código interpretado.
- Bit, Byte, Entero, Word, Largo, y variables tipo String. (Solo con la Prof. Edición).
- Los programas compilados trabajan con todos los microprocesadores (no-MEGA) de AVR que tienen memoria interior. La Prof.-edición apoyará la serie de MEGA también. Puesto que los 1200 no tienen SRAM, no funcionará con los 1200.
- Las instrucciones y comandos de este BASIC son bastante similares a las del Visual Basic y QuickBASIC de Microsoft.
- Comandos específicos para el manejo de displays LCD, integrados I2C e integrados 1WIRE Chips, teclado de PC, teclado de matriz, recepción RC5, software UART. SPI, LCD Gráficos, envío de IR RC5 o código Sony.
- Soporta variables locales, uso de funciones, y librerías.
- Emulador terminal integrado con opción de download.
- Simulador integrado por probar.

- Programador de ISP integrado (aplicación nota AVR910.ASM). Se agregarán otros programadores por pedido.
- Integrado el soporte del programador STK200 y STK300. También soporta el Electronics Programme de bajo costo.
- Editor con subrayador de sentencias.
- Ayuda ON LINE en el editor.

COMANDOS E INSTRUCCIONES

- **De estructura y condicionales**

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNT0, STEP, EXIT FOR, ON, GOTO/GOSUB, SELECT, CASE.

- **De entrada/salida**

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIFTLIN, SHIFTLIN, SHIFTLIN, GETATKBD, SPC

- **Funciones numéricas**

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

- **I2C**

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

- **ONE WIRE**
1WRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST, 1WSEARCHNEXT
- **SPI**
SPIINIT, SPIIN, SPIOUT, SPIMOVE
- **Gestión de interrupciones**
ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.
- **Manipulación de bits**
SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.
- **Variables**
DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD.
- **Varios**
REM, SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP,OUT, ALIAS, DIM , ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, BIN2GREY, GREY2BIN, CRC8, CRC16, CHECKSUM.
- **Directivas**
\$INCLUDE, \$BAUD and \$CRYSTAL, \$SERIALINPUT, \$SERIALOUTPUT, \$RAMSIZE, \$RAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD, \$EXTERNAL, \$LIB.

- **Cadenas**

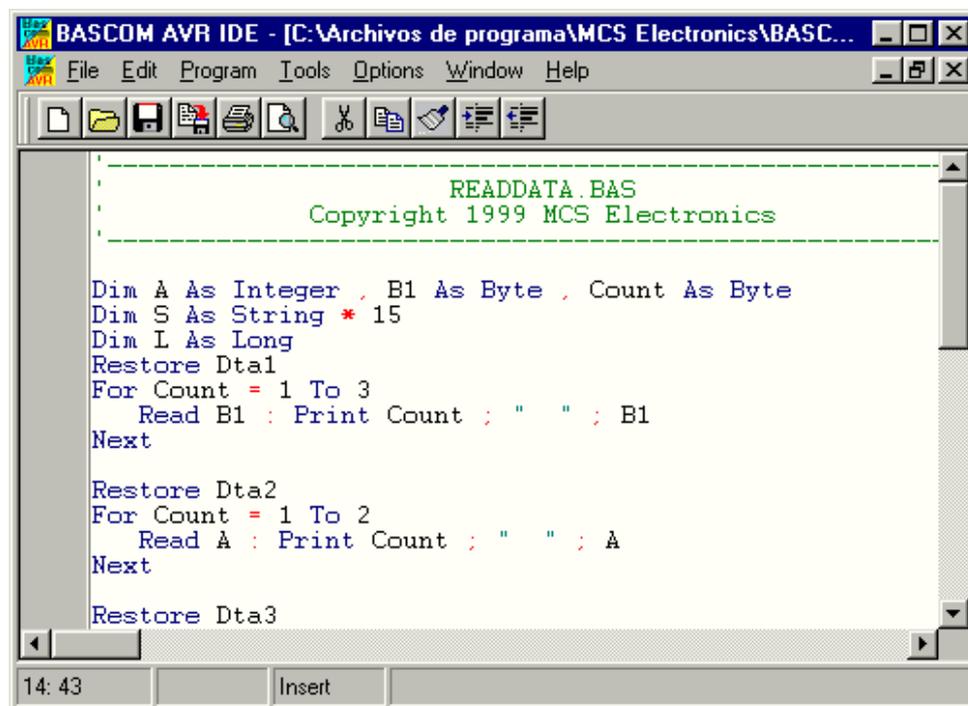
STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

El entorno de desarrollo de BASCOM, es muy sencillo de utilizar. Simplemente deberá realizar las siguientes operaciones:

- Escribir un programa en BASIC, sobre el editor.
- Compilarlo.
- Depurar el resultado con ayuda del simulador integrado.
- Programar el microcontrolador con el programador.

Editor

El programa puede ser escrito sobre un editor MDI intuitivo en color. Este editor, además de las características habituales, soporta Undo, Redo, marcado e indexación de bloques.



```
READDATA.BAS
Copyright 1999 MCS Electronics

Dim A As Integer , B1 As Byte , Count As Byte
Dim S As String * 15
Dim L As Long
Restore Dta1
For Count = 1 To 3
  Read B1 : Print Count ; " " ; B1
Next
Restore Dta2
For Count = 1 To 2
  Read A : Print Count ; " " ; A
Next
Restore Dta3
```

Figura C.1. Pantalla de programación

Simulador

El simulador le permite probar su programa antes de escribirlo al microprocesador. Se puede mirar variables, caminar a través del programa una línea en el momento o correr a una línea específica, o usted puede alterar variables.

Para mirar un valor de las variables se puede apuntar el cursor del ratón encima de la misma.

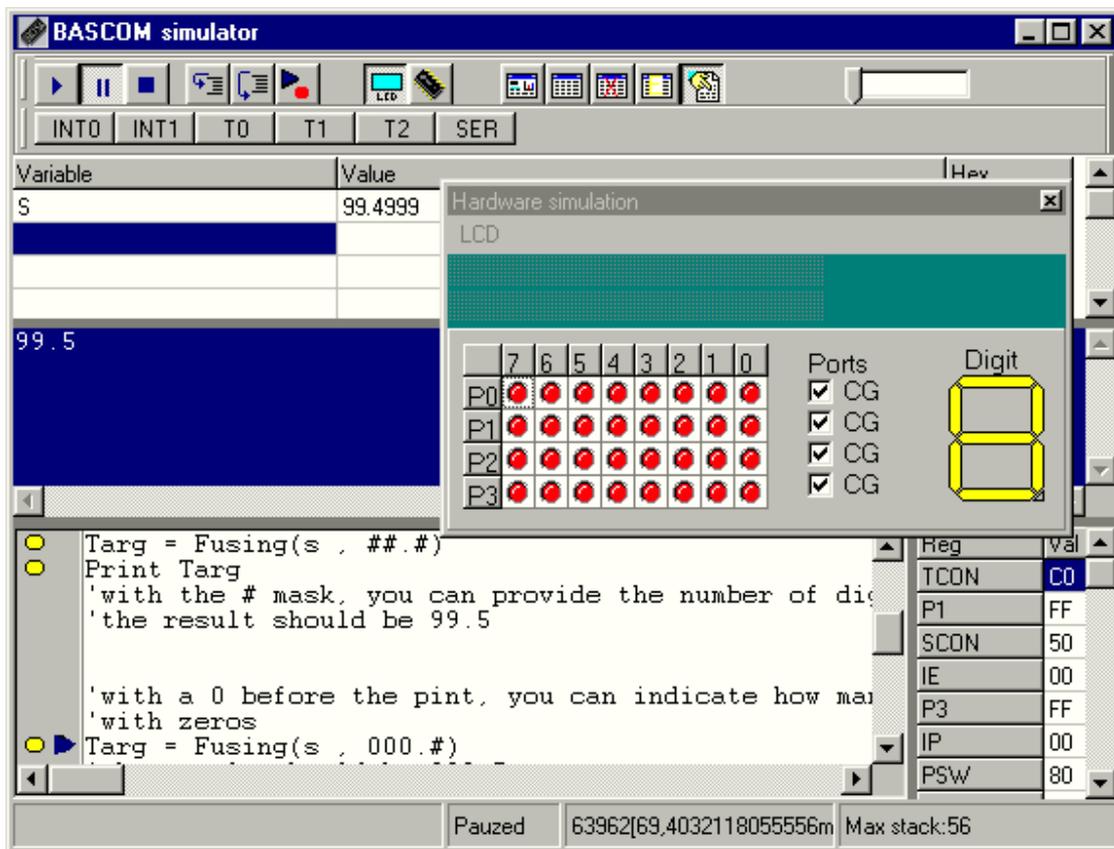


Figura C.2. uP TAB del simulador

Un rasgo poderoso es el emulador del hardware, emula el LCD, y los puertos. El emulador de LCD también emula la figura de costumbre de los caracteres de LCD.

Cuando ha terminado de probar el programa en el simulador, llega el momento de llevar el programa al microcontrolador a través de cualquier programador AVR o universal.

MICROSOFT EMBEDDED VISUAL BASIC

El paquete de herramientas de desarrollo ofrecido por Microsoft no tiene como objetivo sólo a la plataforma Pocket PC, ya que pueden ser usadas también para la creación de aplicaciones que tengan como objetivo las plataformas Palm-size PC y Handheld PC, predecesoras de Pocket PC. Dicho paquete de herramientas no se ejecuta en el dispositivo móvil, sino en un sistema corriente de sobremesa. Con él se efectuaría la edición, compilación y depuración de proyectos que, finalmente, se transferirán al dispositivo móvil para efectuar la instalación definitiva.

Microsoft eMbedded Visual Basic no genera código directamente ejecutable, sino un código intermedio que, posteriormente, es necesario interpretar durante la ejecución. De esta forma se obtienen archivos de código realmente pequeños. Cada dispositivo donde deseemos usar las aplicaciones contará con ese intérprete o runtime que, dependiendo del tipo de procesador que tenga el dispositivo, se encargará de generar el código ejecutable correspondiente.

Embedded Visual Basic no genera programas ejecutables directamente (.exe), eVb genera archivos con la extensión .vb, estos archivos son una especie de script muy optimizado con los datos exactos para ser mostrados en la interfase grafica de Pocket PC, Handheld PC y PalmSize PC.

Estos archivos .vb son enlazados a un programa ejecutable (pvbload.exe) así como se enlazan los archivos .doc a word, por ejemplo. Este programa "pvbload.exe" se encarga de interpretar los comandos que se encuentran en el

archivo .vb y traduce y ejecuta con ayuda de archivos dll para su correcta ejecución en Pocket PC.

El runtime de eVB esta presente en la ROM del sistema Windows CE que tengamos.

Como los archivos .vb son enlazados a pvbload.exe, ellos heredan el icono de este programa. Es por esto que todos los programas eVb tienen el mismo icono.



Barra de Herramientas (Tolva)

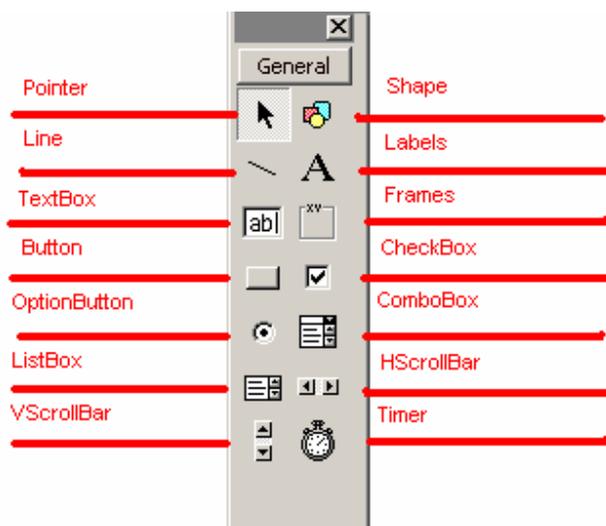


Figura C.3. Barra Tolva

A continuación se va a explicar cada uno de los controles intrínsecos de eVB,

- **Pointer**

Con este se puede cancelar cualquier otro comando que se tenga seleccionado, si nos arrepentimos de haber escogido un control TextBox, utilizamos a Pointer para anular o cancelar la selección.

- **Shape**
Permite crear figuras geométricas en la forma.
- **Line**
Permite dibujar una línea de diferentes anchos y colores en la forma.
- **Labels**
Es uno de los controles más usados, pues en ellos vamos a colocar el texto que deseamos aparezca en las formas para ayudar al usuario en la captura de los datos de una aplicación, por ejemplo. El texto de los labels no puede ser cambiado por el usuario.
- **TextBox**
Es otro de los controles más usados en nuestras aplicaciones, ya que en ellos podemos capturar o visualizar información de diferente índole para actualizar nuestras bases de datos en el Pocket PC.
- **Frames**
Este control es un contenedor de controles, permite insertar en él, cualquier control de los que aparecen en la barra de herramientas. Este control particularmente se lo utiliza para simular diferentes formas, debido al inconveniente de eVB que no permite descargar las formas de memoria.
- **Button**
Permite crear botones para seleccionar diferentes opciones por ejemplo para guardar los datos o para cancelar la grabación de los mismos, para seleccionar una opción diferente, etc.
- **CheckBox**
Permite seleccionar una o varias opciones referentes a un tema.

- **OptionButton**

Permite seleccionar una opción entre varias. Este es excluyente.

- **ComboBox**

Este control permite mostrar una lista de datos desplegable que se muestra y oculta cuando se lo activa dando clic. Este control sirve para colocar en la lista de ciudades, por ejemplo, para escoger una.

- **ListBox**

Este control es similar al ComboBox, la diferencia es que la lista no es desplegable como la del ComboBox. Este control se lo puede utilizar por ejemplo para mostrar los nombres de los empleados de una empresa.

- **HScrollBar y VScrollBar**

Estos dos controles se usan para simular un desplazamiento vertical u horizontal en la pantalla, por ejemplo.

- **Timer**

Este control permite controlar la ejecución de un procedimiento cada determinado tiempo.

ACTIVESYNC

Activesync es el programa que permite sincronizar los datos de nuestro PDA o Pocket PC con el PC de escritorio, permite también instalar los programas en el Pocket. En este momento se encuentra en su versión 4.1 y se puede bajar de forma gratuita del sitio de Microsoft y viene en el CD que acompaña a los Pocket PC.

GLOSARIO

Inmótica

Por "inmótica" se entiende como la incorporación al equipamiento de edificios de uso terciario o industrial (oficinas, edificios corporativos, hoteleros, empresariales y similares), de sistemas de gestión técnica automatizada de las instalaciones, con el objetivo de reducir el consumo de energía, aumentar el confort y la seguridad de los mismos.

Pasarela Domótica

Se denomina pasarela domótica al hardware central, el cual es el encargado de permitir la interacción del usuario con las distintas cargas eléctricas que desee controlar.

Bluetooth

Es el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura, globalmente y sin licencia de corto rango.

Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

Piconet

Una red de dispositivos que se conectan utilizando Bluetooth. Una Piconet puede constar de dos a ocho dispositivos, en la que existirá un máster y los demás serán esclavos.

Unidad Maestra

Éste es el dispositivo que, dentro de una Piconet, definirá cual será la frecuencia a utilizarse para sincronizar todas las unidades esclavas.

Unidad Esclava

Todas las unidades de una Piconet que no sean máster.

Parked Units

Dispositivos que están sincronizados en una Piconet, pero que no poseen MAC address.

Modo Sniff y Hold

Este modo permite a los dispositivos sincronizados dentro de una Piconet, permanecer en modo "ahorro de energía".

Backbone

Mecanismo de conectividad primario en un sistema distribuido. Todos los sistemas que tengan conexión al backbone (columna vertebral) pueden interconectarse entre sí, aún cuando puedan hacerlo directamente o mediante redes alternativas. Un backbone conecta dos puntos o redes distanciadas geográficamente, a altas velocidades.

Interfaz de usuario

En software, una interfaz de usuario es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. Metafóricamente se entiende la Interfaz como conversación entre el usuario y el sistema (o entre el usuario y el diseñador).

AVR

Los AVR son una familia de microcontroladores RISC de Atmel. El AVR es una CPU de arquitectura Harvard. Tiene 32 registros de 8 bits, algunas instrucciones sólo operan en un subconjunto de estos registros. La concatenación de los 32

registros, los registros de entrada/salida y la memoria de datos conforman un espacio de direcciones unificado, al cual se accede a través de operaciones de carga/almacenamiento. A diferencia de los microcontroladores PIC's, el stack se ubica en este espacio de memoria unificado, y no está limitado a un tamaño fijo.

Arquitectura RISC

Para cualquier nivel de desempeño dado, un chip RISC típicamente tendrá menos transistores dedicados a la lógica principal. Esto permite a los diseñadores una flexibilidad considerable al instante de realizar un programa. Las características generalmente encontradas en los diseños RISC son:

- Codificación uniforme de instrucciones (ejemplo: el código de operación se encuentra siempre en la misma posición de bit en cada instrucción, la cual es siempre una palabra), lo que permite una decodificación más rápida.
- Un conjunto de registros homogéneo, permitiendo que cualquier registro sea utilizado en cualquier contexto y así simplificar el diseño del compilador (aunque existen muchas formas de separar los ficheros de registro de entero y coma flotante).
- Modos de direccionamiento simple con modos más complejos reemplazados por secuencias de instrucciones aritméticas simples.
- Algunos tipos de datos soportados en el hardware (por ejemplo, algunas máquinas CISC tiene instrucciones para tratar con tipos byte, cadena; tales instrucciones no se encuentran en una máquina RISC).

Software

Se denomina software, al equipamiento lógico o soporte lógico de todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Hardware

El hardware comprende todos los componentes físicos (que se pueden tocar).

Actuador

Se denominan actuadores a aquellos elementos que pueden provocar un efecto sobre un proceso automatizado.

ANEXO D

CÓDIGOS DE PROGRAMACIÓN

PROGRAMA PARA EL MICROCONTROLADOR ATMEGA64

```
$regfile = "m64def.dat"           ' se especifica el micro a usar
$crystal = 8000000                ' definir la frec. interna del micro
$baud = 57600
'$baud1 = 57600

$crystal = 8000000                ' 8 MHz crystal

'Open "COM2:" For Binary As #1

'Print #1 , "Hello"

Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 =
Portb.7 , E = Portb.3 , Rs = Portb.2
Config Lcd = 16 * 2

Declare Sub Hora(hin As Integer , Hfin As Integer)      'Declaración de la
subrutina Hora

Declare Sub Simulacion(hi1 As Integer , Hi2 As Integer , Hf1 As Integer , Hf2 As
Integer , Carg As Integer)      'Declaración de la subrutina Hora

Declare Sub Carga(c As Integer , Num As Integer)

Declare Sub Lectura()

Config Porta = Output
Config Portc = Output

Config Porte.2 = Input
Config Porte.3 = Input
Config Porte.4 = Input
Config Porte.5 = Input
Config Porte.6 = Input
Config Porte.7 = Input

Config Portb.0 = Input
Config Portb.1 = Input
```

Config Portg = Input
Config Portf = Input

Config Portd.4 = Input
Config Portd.5 = Input
Config Portd.6 = Input
Config Portd.7 = Input

Enable Interrupts
'[iniciar el reloj]

' habilita las interrupciones

Config Date = Mdy , Separator = /

' formato de separacion /

Config Clock = Soft

' usar el real time interno

'Enable Serial
'On Urxc Leer

' habilitar el modulo como esclavo

Dim Atin As String * 20

'variable para almacenar caracteres

Dim Btin As String * 10

Dim Hi(20) As Integer

'variables de hora de inicio

Dim Hf(20) As Integer

'variables de hora de fin

Dim Espera As Integer

Dim Control0 As Integer

Dim Control1 As Integer

Dim Control2 As Integer

Dim Control3 As Integer

Dim Control4 As Integer

Dim Control5 As Integer

Dim Control6 As Integer

Dim Control7 As Integer

Dim Control8 As Integer

Dim Control9 As Integer

Dim Sensor1 As Integer

Dim Sensor2 As Integer

Dim Sensor3 As Integer

Dim Sensor4 As Integer

Dim Sensor5 As Integer

Hi(1) = 24

Hi(2) = 0

'hora de inicio carga 1

Hi(3) = 24

Hi(4) = 0

'hora de inicio carga 2

Hi(5) = 24

Hi(6) = 0

'hora de inicio carga 3

Hi(7) = 24

Hi(8) = 0 'hora de inicio carga 4
Hi(9) = 24
Hi(10) = 0 'hora de inicio carga 5
Hi(11) = 24
Hi(12) = 0 'hora de inicio carga 6
Hi(13) = 24
Hi(14) = 0 'hora de inicio carga 7
Hi(15) = 24
Hi(16) = 0 'hora de inicio carga 8
Hi(17) = 24
Hi(18) = 0 'hora de inicio carga 9
Hi(19) = 24
Hi(20) = 0 'hora de inicio carga 10

Hf(1) = 24
Hf(2) = 0 'hora de fin carga 1
Hf(3) = 24
Hf(4) = 0 'hora de fin carga 2
Hf(5) = 24
Hf(6) = 0 'hora de fin carga 3
Hf(7) = 24
Hf(8) = 0 'hora de fin carga 4
Hf(9) = 24
Hf(10) = 0 'hora de fin carga 5
Hf(11) = 24
Hf(12) = 0 'hora de fin carga 6
Hf(13) = 24
Hf(14) = 0 'hora de fin carga 7
Hf(15) = 24
Hf(16) = 0 'hora de fin carga 8
Hf(17) = 24
Hf(18) = 0 'hora de fin carga 9
Hf(19) = 24
Hf(20) = 0 'hora de fin carga 10

Esclavo: 'Inicio de programa, el bt se pone en
modo de esclavo
Cls 'borra la pantalla
Wait 2 'espera 2 segundos
Atin = "AT+ZV RESET" 'comando para resetear el bt
Print Atin 'envia el comando
'Print #2 , Atin
'Input Atin 'espera una respuesta
Wait 1 'espera 1 segundo

Espera = 100
Locate 1 , 1

```

Lcd "Sistema Domotico"
Dim Juan As Integer
Juan = 0
Do                                     'lazo que verifica
  Locate 2 , 4
  Lcd Time$
  Dim Ju As Integer
  If Juan = 0 Then
    Ju = 1
    Call Simulacion(hi(1) , Hi(2) , Hf(1) , Hf(2) , Ju)
    Ju = 2
    Call Simulacion(hi(3) , Hi(4) , Hf(3) , Hf(4) , Ju)
    Ju = 3
    Call Simulacion(hi(5) , Hi(6) , Hf(5) , Hf(6) , Ju)
    Ju = 4
    Call Simulacion(hi(7) , Hi(8) , Hf(7) , Hf(8) , Ju)
    Ju = 5
    Call Simulacion(hi(9) , Hi(10) , Hf(9) , Hf(10) , Ju)
    Ju = 6
    Call Simulacion(hi(11) , Hi(12) , Hf(11) , Hf(12) , Ju)
    Ju = 7
    Call Simulacion(hi(13) , Hi(14) , Hf(13) , Hf(14) , Ju)
    Ju = 8
    Call Simulacion(hi(15) , Hi(16) , Hf(15) , Hf(16) , Ju)
    Ju = 9
    Call Simulacion(hi(17) , Hi(18) , Hf(17) , Hf(18) , Ju)
    Ju = 10
    Call Simulacion(hi(19) , Hi(20) , Hf(19) , Hf(20) , Ju)
    Juan = Juan + 1
  Else
    Juan = Juan + 1
    Juan = Juan Mod 10
  End If
  $timeout = 200000
  'Input #1 , Btin
  Input Atin Noecho
Loop Until Atin = "H" Or Btin = "H"      ' haga esto hasta que
'comienza la sincronizacion con la pocket pc
$timeout = 500000
Input Atin Noecho
Time$ = Atin                            ' hora / minutos / segundos
Time$ = Time$
Input Atin Noecho
Date$ = Atin                             ' mes / dia / año

Dim Bitdia As Byte
Home                                     'cursor home

```

```
Bitdia = Dayofweek()  
Atin = Lookupstr(bitdia , Weekdays)  
Cls  
Locate 1 , 1  
Lcd Date$  
Locate 1 , 10  
Lcd Atin  
Control0 = Pine.7  
Control1 = Pine.6  
Control2 = Pine.5  
Control3 = Pinb.1  
Control4 = Pinb.0  
Control5 = Pine.4  
Control6 = Pine.3  
Control7 = Pine.2  
Control8 = Pinf.0  
Control9 = Pinf.1
```

'ingresa los valores iniciales de los sensores

```
Sensor1 = Pind.4  
Sensor2 = Pind.5  
Sensor3 = Pind.6  
Sensor4 = Pind.7  
Sensor5 = Ping.0  
Espera = 100  
If Pine.7 = 0 Then  
luces  
    Print "R0E"  
    Waitms Espera  
End If  
If Pine.6 = 0 Then  
    Print "R1E"  
    Waitms Espera  
End If  
If Pine.5 = 0 Then  
    Print "R2E"  
    Waitms Espera  
End If  
If Pinb.1 = 0 Then  
    Print "R3E"  
    Waitms Espera  
End If  
If Pinb.0 = 0 Then  
    Print "R4E"  
    Waitms Espera  
End If
```

'ingresa los valores iniciales de las

```
If Pine.4 = 0 Then
  Print "R5E"
  Waitms Espera
End If
If Pine.3 = 0 Then
  Print "R6E"
  Waitms Espera
End If
If Pine.2 = 0 Then
  Print "R7E"
  Waitms Espera
End If
If Pinf.0 = 0 Then
  Print "R8E"
  Waitms Espera
End If
If Pinf.1 = 0 Then
  Print "R9E"
  Waitms Espera
End If
```

'ingresa los valores iniciales de los sensores

```
If Pind.4 = 1 Then
  Print "S1E"
  Waitms Espera
End If
If Pind.5 = 1 Then
  Print "S2E"
  Waitms Espera
End If
If Pind.6 = 1 Then
  Print "S3E"
  Waitms Espera
End If
If Pind.7 = 1 Then
  Print "S4E"
  Waitms Espera
End If
If Ping.0 = 1 Then
  Print "S5E"
  Waitms Espera
End If
Do
  Call Lectura
```

```
If Pine.7 = 1 And Control0 = 0 Then                                '1º carga detector de estado =
pine.7                                                            'interruptor = pina.0
  Print "R0A"
  Control0 = Pine.7
  Call Lectura
End If
```

```
If Pine.7 = 0 And Control0 = 1 Then
  Print "R0E"
  Control0 = Pine.7
  Call Lectura
End If
```

```
If Pine.6 = 1 And Control1 = 0 Then                                '2º carga detector de estado =
pine.6                                                            'interruptor = pina.1
  Print "R1A"
  Control1 = Pine.6
  Call Lectura
End If
```

```
If Pine.6 = 0 And Control1 = 1 Then
  Print "R1E"
  Control1 = Pine.6
  Call Lectura
End If
```

```
If Pine.5 = 1 And Control2 = 0 Then                                '3º carga detector de estado =
pine.5                                                            'interruptor = pina.2
  Print "R2A"
  Control2 = Pine.5
  Call Lectura
End If
```

```
If Pine.5 = 0 And Control2 = 1 Then
  Print "R2E"
  Control2 = Pine.5
  Call Lectura
End If
```

```
If Pinb.1 = 1 And Control3 = 0 Then                                '4º carga detector de estado =
pinb.1                                                            'interruptor = pina.3
  Print "R3A"
  Control3 = Pinb.1
  Call Lectura
End If
```

```
If Pinb.1 = 0 And Control3 = 1 Then
```

```
Print "R3E"  
Control3 = Pinb.1  
Call Lectura  
End If
```

```
If Pinb.0 = 1 And Control4 = 0 Then  
pinb.0  
Print "R4A"  
Control4 = Pinb.0  
Call Lectura  
End If
```

'5º carga detector de estado =
'interruptor = pina.4

```
If Pinb.0 = 0 And Control4 = 1 Then  
Print "R4E"  
Control4 = Pinb.0  
Call Lectura  
End If
```

```
If Pine.4 = 1 And Control5 = 0 Then  
pine.4  
Print "R5A"  
Control5 = Pine.4  
Call Lectura  
End If
```

'6º carga detector de estado =
'interruptor = pina.5

```
If Pine.4 = 0 And Control5 = 1 Then  
Print "R5E"  
Control5 = Pine.4  
Call Lectura  
End If
```

```
If Pine.3 = 1 And Control6 = 0 Then  
pine.3  
Print "R6A"  
Control6 = Pine.3  
Call Lectura  
End If
```

'7º carga detector de estado =
'interruptor = pina.6

```
If Pine.3 = 0 And Control6 = 1 Then  
Print "R6E"  
Control6 = Pine.3  
Call Lectura  
End If
```

```
If Pine.2 = 1 And Control7 = 0 Then  
pine.2  
Print "R7A"
```

'8º carga detector de estado =
'interruptor = pina.7

```
Control7 = Pine.2
Call Lectura
End If
```

```
If Pine.2 = 0 And Control7 = 1 Then
Print "R7E"
Control7 = Pine.2
Call Lectura
End If
```

```
If Pinf.0 = 1 And Control8 = 0 Then
pinf.0
Print "R8A"
Control8 = Pinf.0
Call Lectura
End If
```

'9º carga detector de estado =
'interruptor = pinc.0

```
If Pinf.0 = 0 And Control8 = 1 Then
Print "R8E"
Control8 = Pinf.0
Call Lectura
End If
```

```
If Pinf.1 = 1 And Control9 = 0 Then
pinf.1
Print "R9A"
Control9 = Pinf.1
Call Lectura
End If
```

'10º carga detector de estado =
'interruptor = pinc.1

```
If Pinf.1 = 0 And Control9 = 1 Then
Print "R9E"
Control9 = Pinf.1
Call Lectura
End If
```

```
If Pind.4 = 1 And Sensor1 = 0 Then
Print "S1E"
Sensor1 = Pind.4
Call Lectura
End If
```

'1º sensor = pind.4

```
If Pind.4 = 0 And Sensor1 = 1 Then
Print "S1A"
Sensor1 = Pind.4
Call Lectura
End If
```


End If

```
If Mid(atin , 1 , 1 ) = "L" Then
  If Mid(atin , 2 , 1 ) = "0" Then
    If Pine.7 = 0 Then
      Print "R0A"
    Else
      Print "R0E"
    End If
    If Porta.0 = 1 Then
      Porta.0 = 0
    Else
      Porta.0 = 1
    End If
    Call Lectura
  End If
```

```
If Mid(atin , 2 , 1 ) = "1" Then
  If Pine.6 = 0 Then
    Print "R1A"
  Else
    Print "R1E"
  End If
  If Porta.1 = 1 Then
    Porta.1 = 0
  Else
    Porta.1 = 1
  End If
  Call Lectura
End If
```

```
If Mid(atin , 2 , 1 ) = "2" Then
  If Pine.5 = 0 Then
    Print "R2A"
  Else
    Print "R2E"
  End If
  If Porta.2 = 1 Then
    Porta.2 = 0
  Else
    Porta.2 = 1
  End If
  Call Lectura
End If
```

```
If Mid(atin , 2 , 1 ) = "3" Then
  If Pinb.1 = 0 Then
```

```

    Print "R3A"
Else
    Print "R3E"
End If
If Porta.3 = 1 Then
    Porta.3 = 0
Else
    Porta.3 = 1
End If
Call Lectura
End If

If Mid(atin , 2 , 1 ) = "4" Then
    If Pinb.0 = 0 Then
        Print "R4A"
    Else
        Print "R4E"
    End If
    If Porta.4 = 1 Then
        Porta.4 = 0
    Else
        Porta.4 = 1
    End If
    Call Lectura
End If

If Mid(atin , 2 , 1 ) = "5" Then
    If Pine.4 = 0 Then
        Print "R5A"
    Else
        Print "R5E"
    End If
    If Porta.5 = 1 Then
        Porta.5 = 0
    Else
        Porta.5 = 1
    End If
    Call Lectura
End If

If Mid(atin , 2 , 1 ) = "6" Then
    If Pine.3 = 0 Then
        Print "R6A"
    Else
        Print "R6E"
    End If
    If Porta.6 = 1 Then

```

```
    Porta.6 = 0
Else
    Porta.6 = 1
End If
Call Lectura
End If
```

```
If Mid(atin , 2 , 1 ) = "7" Then
    If Pine.2 = 0 Then
        Print "R7A"
    Else
        Print "R7E"
    End If
    If Porta.7 = 1 Then
        Porta.7 = 0
    Else
        Porta.7 = 1
    End If
    Call Lectura
End If
```

```
If Mid(atin , 2 , 1 ) = "8" Then
    If Pinf.0 = 0 Then
        Print "R8A"
    Else
        Print "R8E"
    End If
    If Portc.0 = 1 Then
        Portc.0 = 0
    Else
        Portc.0 = 1
    End If
    Call Lectura
End If
```

```
If Mid(atin , 2 , 1 ) = "9" Then
    If Pinf.1 = 0 Then
        Print "R9A"
    Else
        Print "R9E"
    End If
    If Portc.1 = 1 Then
        Portc.1 = 0
    Else
        Portc.1 = 1
    End If
    Call Lectura
```

End If

End If

!*****

If Mid(atin , 1 , 1) = "C" Then

'comando para enviar la hora

Atin = Right(atin , 1)

If Atin = "1" Then

Atin = "i"

Call Hora(hi(1) , Hi(2))

Atin = Atin + "f"

Call Hora(hf(1) , Hf(2))

Print Atin

Call Lectura

End If

If Atin = "2" Then

Atin = "i"

Call Hora(hi(3) , Hi(4))

Atin = "f"

Call Hora(hf(3) , Hf(4))

Print Atin

Call Lectura

End If

If Atin = "3" Then

Atin = "i"

Call Hora(hi(5) , Hi(6))

Atin = "f"

Call Hora(hf(5) , Hf(6))

Print Atin

Call Lectura

End If

If Atin = "4" Then

Atin = "i"

Call Hora(hi(7) , Hi(8))

Atin = "f"

Call Hora(hf(7) , Hf(8))

Print Atin

Call Lectura

End If

If Atin = "5" Then

Atin = "i"

Call Hora(hi(9) , Hi(10))

Atin = "f"

Call Hora(hf(9) , Hf(10))

Print Atin

```

    Call Lectura
End If
If Atin = "6" Then
    Atin = "i"
    Call Hora(hi(11) , Hi(12))
    Atin = "f"
    Call Hora(hf(11) , Hf(12))
    Print Atin
    Call Lectura
End If
If Atin = "7" Then
    Atin = "i"
    Call Hora(hi(13) , Hi(14))
    Atin = "f"
    Call Hora(hf(13) , Hf(14))
    Print Atin
    Call Lectura
End If
If Atin = "8" Then
    Atin = "i"
    Call Hora(hi(15) , Hi(16))
    Atin = "f"
    Call Hora(hf(15) , Hf(16))
    Print Atin
    Call Lectura
End If
If Atin = "9" Then
    Atin = "i"
    Call Hora(hi(17) , Hi(18))
    Atin = "f"
    Call Hora(hf(17) , Hf(18))
    Print Atin
    Call Lectura
End If
If Atin = "A" Then
    Atin = "i"
    Call Hora(hi(19) , Hi(20))
    Atin = "f"
    Call Hora(hf(19) , Hf(20))
    Print Atin
    Call Lectura
End If
End If
If Right(atin , 1) = "d" Then
    Dim Hm As String * 2
    Dim Ti As Integer
    Hm = Mid(atin , 1 , 1)

```

'comando para recibir la hora

```
Ti = Val(hm)
Ti = Ti * 2
Ti = Ti + 1
Hm = Mid(atin , 2 , 2)
If Mid(hm , 1 , 1) = "0" Then
    Hm = Mid(atin , 3 , 1)
    Hi(ti) = Val(hm)
Else
    Hi(ti) = Val(hm)
End If
Hm = Mid(atin , 5 , 2)
Hi(ti + 1) = Val(hm)
Hm = Mid(atin , 7 , 2)
Hf(ti) = Val(hm)
Hm = Mid(atin , 10 , 2)
Hf(ti + 1) = Val(hm)
End If
```

```
If Right(atin , 2) = "ER" Then
    Goto Esclavo
End If
```

```
If Left(atin , 2) = "##" Then
    Goto Esclavo
End If
```

```
'Locate 2 , 4
'LCD Time$
```

Loop

End

```
Sub Lectura()
    Input Atin
End Sub
```

```
Sub Hora(hin As Integer , Hfin As Integer)
    If Hin < 10 Then
        Atin = Atin + "0"
    End If
    Atin = Atin + Str(hin)
    Atin = Atin + ":"
    If Hfin < 10 Then
        Atin = Atin + "0"
    End If
    Atin = Atin + Str(Hfin)
    Atin = Atin + ":"
    Atin = Atin + Str(Atin)
End Sub
```

```
End If
Atin = Atin + Str(hfin)
End Sub
```

```
Sub Simulacion(hi1 As Integer , Hi2 As Integer , Hf1 As Integer , Hf2 As Integer ,
Carg As Integer)
Dim Bt As Integer
If Hi1 <> 24 Then                                     'Pregunta si es la hora es diferente a
24, 24 significa que no se ingreso horario
    Atin = Mid(time$ , 1 , 2)                         'Ingresa a la variable Atin el valor de
la hora actual
    If Hf1 > Val(atin) And Hi1 < Val(atin) Then
        Bt = 1
        Call Carga(carg , Bt)
    Else
        If Hi1 = Val(atin) Then
            If Hf1 = Val(atin) Then
                Atin = Mid(time$ , 4 , 2)
                If Hi2 <= Val(atin) And Hf2 > Val(atin) Then
                    Bt = 1
                    Call Carga(carg , Bt)
                Else
                    Bt = 0
                    'Call Carga(carg , Bt)
                End If
            Else
                Atin = Mid(time$ , 4 , 2)
                If Hi2 <= Val(atin) Then
                    Bt = 1
                    Call Carga(carg , Bt)
                Else
                    Bt = 0
                    'Call Carga(carg , Bt)
                End If
            End If
        End If
    Else
        If Hf1 <= Val(atin) Then
            Atin = Mid(time$ , 4 , 2)
            If Hf2 > Val(atin) Then
                Bt = 1
                Call Carga(carg , Bt)
            Else
                Bt = 0
                'Call Carga(carg , Bt)
            End If
        End If
    Else
        Bt = 0
    End If
End Sub
```

```

        Call Carga(carg , Bt)
    End If
End If
End If
Atin = Mid(time$ , 1 , 2)
If Hf1 = Val(atin) Then
    Atin = Mid(time$ , 4 , 2)
    If Hf2 = Val(atin) Then
        Atin = Mid(time$ , 7 , 2)
        If Val(atin) = 0 Then
            Bt = 2
            Call Carga(carg , Bt)
        End If
    End If
End If
End If
End Sub

```

```

Sub Carga(c As Integer , Num As Integer)

```

```

    If Num = 1 Then
        If C = 1 Then
            If Pine.7 = 1 Then
                If Porta.0 = 1 Then
                    Porta.0 = 0
                Else
                    Porta.0 = 1
                End If
            End If
        End If
    End If
    If C = 2 Then
        If Pine.6 = 1 Then
            If Porta.1 = 1 Then
                Porta.1 = 0
            Else
                Porta.1 = 1
            End If
        End If
    End If
    If C = 3 Then
        If Pine.5 = 1 Then
            If Porta.2 = 1 Then
                Porta.2 = 0
            Else
                Porta.2 = 1
            End If
        End If
    End If
End Sub

```

'Verifica el estado de la carga
'si es 1 significa que esta
'apagado y lo cambia para encenderlo.

```
If C = 4 Then
  If Pinb.1 = 1 Then
    If Porta.3 = 1 Then
      Porta.3 = 0
    Else
      Porta.3 = 1
    End If
  End If
End If
If C = 5 Then
  If Pinb.0 = 1 Then
    If Porta.4 = 1 Then
      Porta.4 = 0
    Else
      Porta.4 = 1
    End If
  End If
End If
If C = 6 Then
  If Pine.4 = 1 Then
    If Porta.5 = 1 Then
      Porta.5 = 0
    Else
      Porta.5 = 1
    End If
  End If
End If
If C = 7 Then
  If Pine.3 = 1 Then
    If Porta.6 = 1 Then
      Porta.6 = 0
    Else
      Porta.6 = 1
    End If
  End If
End If
If C = 8 Then
  If Pine.2 = 1 Then
    If Porta.7 = 1 Then
      Porta.7 = 0
    Else
      Porta.7 = 1
    End If
  End If
End If
If C = 9 Then
  If Pinf.0 = 1 Then
```

```

    If Portc.0 = 1 Then
        Portc.0 = 0
    Else
        Portc.0 = 1
    End If
End If
End If
If C = 10 Then
    If Pinf.1 = 1 Then
        If Portc.1 = 1 Then
            Portc.1 = 0
        Else
            Portc.1 = 1
        End If
    End If
End If
Wait 1
Else
    If Num = 2 Then

        If C = 1 Then
            If Pine.7 = 0 Then
                If Porta.0 = 1 Then
                    Porta.0 = 0
                Else
                    Porta.0 = 1
                End If
            End If
        End If
    End If
    If C = 2 Then
        If Pine.6 = 0 Then
            If Porta.1 = 1 Then
                Porta.1 = 0
            Else
                Porta.1 = 1
            End If
        End If
    End If
    If C = 3 Then
        If Pine.5 = 0 Then
            If Porta.2 = 1 Then
                Porta.2 = 0
            Else
                Porta.2 = 1
            End If
        End If
    End If
End If

```

'Verifica el estado de la carga
'si es 0 significa que esta
'encendido y lo cambia para apagarlo.

```
If C = 4 Then
  If Pinb.1 = 0 Then
    If Porta.3 = 1 Then
      Porta.3 = 0
    Else
      Porta.3 = 1
    End If
  End If
End If
If C = 5 Then
  If Pinb.0 = 0 Then
    If Porta.4 = 1 Then
      Porta.4 = 0
    Else
      Porta.4 = 1
    End If
  End If
End If
If C = 6 Then
  If Pine.4 = 0 Then
    If Porta.5 = 1 Then
      Porta.5 = 0
    Else
      Porta.5 = 1
    End If
  End If
End If
If C = 7 Then
  If Pine.3 = 0 Then
    If Porta.6 = 1 Then
      Porta.6 = 0
    Else
      Porta.6 = 1
    End If
  End If
End If
If C = 8 Then
  If Pine.2 = 0 Then
    If Porta.7 = 1 Then
      Porta.7 = 0
    Else
      Porta.7 = 1
    End If
  End If
End If
If C = 9 Then
  If Pinf.0 = 0 Then
```

```

        If Portc.0 = 1 Then
            Portc.0 = 0
        Else
            Portc.0 = 1
        End If
    End If
End If
If C = 10 Then
    If Pinf.1 = 0 Then
        If Portc.1 = 1 Then
            Portc.1 = 0
        Else
            Portc.1 = 1
        End If
    End If
End If
Wait 1                                'tiempo de retardo usado para permitir que
se establezca el detector de estado
End If
End If
End Sub

```

```

'Leer:
' Input Atin Noecho
'Return

```

Weekdays:

```
Data "Lunes" , "Martes" , "Miercoles" , "Jueves" , "Viernes" , "Sabado" , "Domingo"
```

PROGRAMA DE LA PDA

FORMULARIO CARATULA

Option Explicit

```

Private Sub Form_Load()
    Img.Picture = App.Path + "\btn\Caratula.bmp" 'carga la imagen de presentacion
    Caratula.SIPVisible = False                'oculta el teclado
End Sub

```

```
Private Sub Tmrbtn_Timer()
```

```

PlaySoundFile ("sonidos\inicio.wav")      'reproduce el sonido de inicio
cont.Show                                 'carga el formulario de la contraseña
Tmrbtn.Enabled = False                    'desabilita el timer
End Sub

```

FORMULARIO CONTRASEÑA

Option Explicit

```

Private Sub Command1_Click()
    If Label4.Caption = "1234" Then        'comprueba si la contraseña es correcta
        cont.SIPVisible = False           'oculta el teclado
        Menu.SIPVisible = False           'oculta el teclado del siguiente formulario
        Menu.Show                          'muestra el siguiente formulario

    Else
        MsgBox "¡La Clave Ingresada es Incorrecta!", vbCritical, "Error" 'muestra un
msj de error
        Text1.Text = ""                   'limpia el texto de la contraseña
        Label4.Caption = ""                'limpia el valor de la contraseña
    End If
End Sub

```

```

Private Sub Form_Load()
imgclave.Picture = App.Path + "\btn\Clave.bmp" 'muestra la imagen de la clave
Caratula.Hide                                'oculta la pantalla anterior
End Sub

```

```

Private Sub text1_KeyPress(KeyAscii As Integer)
Dim a As Integer                             'define la variable de longitud
Dim i As Integer
Dim Clave As String
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8, 13
a = Len(Text1.Text)
If KeyAscii = 13 Then
    If Label4.Caption = "1234" Then
        Menu.Show
    Else
        Call Command1_Click
    End If
End If
If KeyAscii = 8 And a >= 0 Then

```

```

If a = 0 Then
    Label4.Caption
Else
    Text1.Text = ""
    For i = 1 To a - 1
        Text1.Text = Text1.Text + "*"
    Next i
    Clave = Label4.Caption
    Label4.Caption = Mid(Clave, 1, a - 1)
End If
Else
    If KeyAscii = 13 Then
        Text1.Text = ""
        Label4.Caption = ""
        KeyAscii = 0
    Else
        If a < 6 Then
            Label4.Caption = Label4.Caption + Chr(KeyAscii)
            KeyAscii = 0
            Text1.Text = Text1.Text + "*"
        Else
            KeyAscii = 0
        End If
    End If
End If

Case Else
    KeyAscii = 0
End Select
End Sub

```

```

Private Sub Tmrfoco_Timer()
    Text1.SetFocus
    Tmrfoco.Enabled = False
End Sub

```

Formulario Menu

```

Option Explicit
Dim a As Integer
Dim i As Integer
Dim j As Integer
Dim tin As String 'variable para almacenamiento del texto de entrada

```

```
Dim contador As Integer
Dim hor As Integer 'variable que permite ingresar los valores de las horas de inicio
y fin
Dim Cmbhora As String
```

```
Private Sub btnguardar_Click()
Call chequeo
If Text3.Text = "none " Or Text4.Text = "none " Then
    Text1.Text = cmbhorario.ListIndex
    Text1.Text = Text1.Text + "24:0024:00d"
End If
Call enviar
btnguardar.Enabled = False
End Sub
```

```
Private Sub Comm1_OnComm()
tin = tin + Comm1.Input
Call Comandos
Text2.Text = tin
End Sub
```

```
Private Sub Comandos()
If Right(tin, 1) = Chr(10) Then
    a = Len(tin)
    a = a - 2
    tin = Mid(tin, 1, a)
    If Mid(tin, 1, 1) = "i" Then
        Text3.Text = Mid(tin, 2, 5)
        If Text3.Text = "24:00" Then
            Text3.Text = "none "
        End If
        Text4.Text = Mid(tin, 8, 5)
        If Text4.Text = "24:00" Then
            Text4.Text = "none "
        End If
    End If
    If Mid(tin, 1, 1) = "R" Then
        Luz (Right(tin, 2))
    End If
    If Mid(tin, 1, 1) = "S" Then
        sensor (Right(tin, 2))
    End If
    tin = ""
End If
```

```
End Sub
Function sensor(sen As String) As Integer
If Right(sen, 1) = "A" Then
```

```

If Mid(sen, 1, 1) = "1" Then
    imgsens1.Picture = App.Path + "\btn\srojo.bmp"
End If
If Mid(sen, 1, 1) = "2" Then
    imgsens2.Picture = App.Path + "\btn\srojo.bmp"
End If
If Mid(sen, 1, 1) = "3" Then
    imgsens3.Picture = App.Path + "\btn\srojo.bmp"
End If
If Mid(sen, 1, 1) = "4" Then
    imgsens4.Picture = App.Path + "\btn\srojo.bmp"
End If
If Mid(sen, 1, 1) = "5" Then
    imgsens5.Picture = App.Path + "\btn\srojo.bmp"
End If
End If
If Right(sen, 1) = "E" Then
    If Mid(sen, 1, 1) = "1" Then
        imgsens1.Picture = App.Path + "\btn\sverde.bmp"
    End If
    If Mid(sen, 1, 1) = "2" Then
        imgsens2.Picture = App.Path + "\btn\sverde.bmp"
    End If
    If Mid(sen, 1, 1) = "3" Then
        imgsens3.Picture = App.Path + "\btn\sverde.bmp"
    End If
    If Mid(sen, 1, 1) = "4" Then
        imgsens4.Picture = App.Path + "\btn\sverde.bmp"
    End If
    If Mid(sen, 1, 1) = "5" Then
        imgsens5.Picture = App.Path + "\btn\sverde.bmp"
    End If
End If
End If

End Function
Function Luz(std As String) As Integer
If Right(std, 1) = "A" Then
    If Mid(std, 1, 1) = "0" Then
        Luz0.Picture = App.Path + "\btn\luzoff.bmp"
    End If
    If Mid(std, 1, 1) = "1" Then
        Luz1.Picture = App.Path + "\btn\luzoff.bmp"
    End If
    If Mid(std, 1, 1) = "2" Then
        Luz2.Picture = App.Path + "\btn\luzoff.bmp"
    End If
    If Mid(std, 1, 1) = "3" Then

```

```
Luz3.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "4" Then
    Luz4.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "5" Then
    Luz5.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "6" Then
    Luz6.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "7" Then
    Luz7.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "8" Then
    Luz8.Picture = App.Path + "\btn\luzoff.bmp"
End If
If Mid(std, 1, 1) = "9" Then
    Luz9.Picture = App.Path + "\btn\luzoff.bmp"
End If
End If
If Right(std, 1) = "E" Then
    If Mid(std, 1, 1) = "0" Then
        Luz0.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "1" Then
        Luz1.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "2" Then
        Luz2.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "3" Then
        Luz3.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "4" Then
        Luz4.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "5" Then
        Luz5.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "6" Then
        Luz6.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "7" Then
        Luz7.Picture = App.Path + "\btn\luzon.bmp"
    End If
    If Mid(std, 1, 1) = "8" Then
```

```
Luz8.Picture = App.Path + "\btn\luzon.bmp"  
End If  
If Mid(std, 1, 1) = "9" Then  
    Luz9.Picture = App.Path + "\btn\luzon.bmp"  
End If  
End If  
End Function
```

```
Private Sub Form_Load()  
cont.Hide  
Dim IIResult As Long
```

```
'calling shell with parameter value
```

```
IIResult = Shell("iPAQWireless.exe", App.Path & "\myapp.htm#main_contents")
```

```
'calling shell without parameter value
```

```
IIResult = Shell("iexplore.exe", "") 'shell
```

```
If IIResult <> 0 Then MsgBox "Shell() error: " & IIResult  
Menu.SIPVisible = False  
PlaySoundFile ("\sonidos\encendido.wav")  
If Comm1.PortOpen = False Then  
    Picbtn1.Picture = App.Path + "\btn\btnencender.bmp"  
    picbtn2.Picture = App.Path + "\btn\btnapagar1o.bmp"  
    Imgblue.Picture = App.Path + "\btn\blueoff.bmp"  
Else  
    Picbtn1.Picture = App.Path + "\btn\btnencender1o.bmp"  
    picbtn2.Picture = App.Path + "\btn\btnapagar1.bmp"  
    Imgblue.Picture = App.Path + "\btn\bluelogo.bmp"  
End If  
Frm1.Visible = True  
Frm2.Visible = False  
Frm3.Visible = False  
Frm4.Visible = False  
cmbhorario.AddItem "Carga1"  
cmbhorario.AddItem "Carga2"  
cmbhorario.AddItem "Carga3"  
cmbhorario.AddItem "Carga4"  
cmbhorario.AddItem "Carga5"  
cmbhorario.AddItem "Carga6"  
cmbhorario.AddItem "Carga7"  
cmbhorario.AddItem "Carga8"  
cmbhorario.AddItem "Carga9"  
cmbhorario.AddItem "Carga10"
```

```
Luz0.Picture = App.Path + "\btn\luzoff.bmp"  
Luz1.Picture = App.Path + "\btn\luzoff.bmp"  
Luz2.Picture = App.Path + "\btn\luzoff.bmp"  
Luz3.Picture = App.Path + "\btn\luzoff.bmp"  
Luz4.Picture = App.Path + "\btn\luzoff.bmp"  
Luz5.Picture = App.Path + "\btn\luzoff.bmp"  
Luz6.Picture = App.Path + "\btn\luzoff.bmp"  
Luz7.Picture = App.Path + "\btn\luzoff.bmp"  
Luz8.Picture = App.Path + "\btn\luzoff.bmp"  
Luz9.Picture = App.Path + "\btn\luzoff.bmp"
```

```
imgsensor1.Picture = App.Path + "\btn\strojo.bmp"  
imgsensor2.Picture = App.Path + "\btn\strojo.bmp"  
imgsensor3.Picture = App.Path + "\btn\strojo.bmp"  
imgsensor4.Picture = App.Path + "\btn\strojo.bmp"  
imgsensor5.Picture = App.Path + "\btn\strojo.bmp"
```

End Sub

```
Private Sub Form_OKClick()  
    If Comm1.PortOpen = True Then  
        Comm1.PortOpen = False  
    End If  
    App.End  
End Sub
```

```
Private Sub Luz0_Click()  
    Text1.Text = "L0"  
    Call enviar  
End Sub
```

```
Private Sub Luz1_Click()  
    Text1.Text = "L1"  
    Call enviar  
End Sub
```

```
Private Sub Luz2_Click()  
    Text1.Text = "L2"  
    Call enviar  
End Sub
```

```
Private Sub Luz3_Click()
```

```
Text1.Text = "L3"  
Call enviar  
End Sub
```

```
Private Sub Luz4_Click()  
Text1.Text = "L4"  
Call enviar  
End Sub
```

```
Private Sub Luz5_Click()  
Text1.Text = "L5"  
Call enviar  
End Sub
```

```
Private Sub Luz6_Click()  
Text1.Text = "L6"  
Call enviar  
End Sub
```

```
Private Sub Luz7_Click()  
Text1.Text = "L7"  
Call enviar  
End Sub
```

```
Private Sub Luz8_Click()  
Text1.Text = "L8"  
Call enviar  
End Sub
```

```
Private Sub Luz9_Click()  
Text1.Text = "L9"  
Call enviar  
End Sub
```

```
Private Sub Picbtn1_Click()  
If Comm1.PortOpen = False Then  
Picbtn1.Picture = App.Path + "\btn\btncender2.bmp"  
Tmrbtn1.Enabled = True  
End If  
End Sub
```

```
Private Sub Text3_Change()  
btnguardar.Enabled = True  
Text1.Text = cmbhorario.ListIndex  
Text1.Text = Text1.Text + Text3.Text + Text4.Text + "d"  
End Sub
```

```
Private Sub Text4_Change()  
btnguardar.Enabled = True  
Text1.Text = cmbhorario.ListIndex  
Text1.Text = Text1.Text + Text3.Text + Text4.Text + "d"  
End Sub
```

```
Private Sub Tmrbtn1_Timer()  
Picbtn1.Picture = App.Path + "\btn\btnencender1o.bmp"  
Tmrbtn1.Enabled = False  
Menu.SIPVisible = False  
On Error Resume Next  
Comm1.CommPort = 7  
Comm1.Settings = "57600,N,8,1"  
Comm1.PortOpen = True  
Comm1.InputLen = 0  
Comm1.RThreshold = 1  
Comm1.SThreshold = 1  
If Comm1.PortOpen = False Then  
MsgBox "No se pudo conectar con el dispositivo, por favor vuelva a  
intentarlo", vbCritical, "Dispositivo no disponible"  
Picbtn1.Picture = App.Path + "\btn\btnencender.bmp"  
picbtn2.Picture = App.Path + "\btn\btnapagar1o.bmp"  
Else  
Imgblue.Picture = App.Path + "\btn\bluelogo.bmp"  
picbtn2.Picture = App.Path + "\btn\btnapagar.bmp"  
Call sincronizacion  
Tmrpuerto.Enabled = True  
conta.Caption = "0"  
End If  
End Sub
```

```
Private Sub Picbtn2_Click()  
If Comm1.PortOpen = True Then  
picbtn2.Picture = App.Path + "\btn\btnapagar2.bmp"  
Tmrbtn2.Enabled = True  
End If  
End Sub
```

```
Private Sub Tmrbtn2_Timer()  
picbtn2.Picture = App.Path + "\btn\btnapagar1o.bmp"  
Tmrbtn2.Enabled = False  
On Error Resume Next  
Comm1.PortOpen = False  
If Comm1.PortOpen = False Then  
Imgblue.Picture = App.Path + "\btn\blueoff.bmp"  
Picbtn1.Picture = App.Path + "\btn\btnencender.bmp"
```

```

    Tmrpuerto.Enabled = False
    conta.Caption = "0"
Else
    Picbtn1.Picture = App.Path + "\btn\btencender1o.bmp"
    picbtn2.Picture = App.Path + "\btn\btnapagar.bmp"
End If
End Sub

Private Sub tabStrip_Click()
    Select Case tabStrip.SelectedItem.Index
    Case 1
        Frm1.Visible = True
        Frm2.Visible = False
        Frm3.Visible = False
        Frm4.Visible = False
    Case 2
        If Comm1.PortOpen = False Then
            MsgBox "Escoja primero un dispositivo Bluetooth", vbInformation, "No hay
conexion"
        Else
            Frm1.Visible = False
            Frm2.Visible = True
            Frm3.Visible = False
            Frm4.Visible = False

        End If
    Case 3
        If Comm1.PortOpen = False Then
            MsgBox "Escoja primero un dispositivo Bluetooth", vbInformation, "No hay
conexion"
        Else
            Frm1.Visible = False
            Frm2.Visible = False
            Frm3.Visible = True
            Frm4.Visible = False

        End If
    Case 4
        If Comm1.PortOpen = False Then
            MsgBox "Escoja primero un dispositivo Bluetooth", vbInformation, "No hay
conexion"
        Else
            Frm1.Visible = False
            Frm2.Visible = False
            Frm3.Visible = False
            Frm4.Visible = True
            Text1.Text = "C1"
            Call enviar
        End If
    End Select
End Sub

```

```
End If
End Select
End Sub
```

```
Private Sub Tmrload_Timer()
Menu.SIPVisible = False
Tmrload.Enabled = False
End Sub
```

```
Function hinicio(KeyAscii As Integer, tec1 As String, tec2 As String) As Integer
If tec2 = "none " Then
Dim a As Integer
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 8
a = Len(tec1)
If a < 5 Then
If a = 0 Then
Select Case KeyAscii
Case 48, 49, 50
Case Else
KeyAscii = 0
End Select
Else
If a = 1 Then
If tec1 = "0" Or tec1 = "1" Then
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8
Case Else
KeyAscii = 0
End Select
Else
Select Case KeyAscii
Case 48, 49, 50, 51, 8
Case Else
KeyAscii = 0
End Select
End If
Else
If a = 2 Then
Select Case KeyAscii
Case 58, 8
Case Else
KeyAscii = 0
End Select
Else
If a = 3 Then
```

```
    Select Case KeyAscii
    Case 48, 49, 50, 51, 52, 53, 8
    Case Else
    KeyAscii = 0
    End Select
```

```
        End If
    End If
End If
Else
End If
Case Else
KeyAscii = 0
End Select
Else
```

```
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 8
a = Len(tec1)
```

```
If a < 5 Then
```

```
If a = 0 Then
```

```
    Select Case KeyAscii
```

```
    Case 48, 49, 50
```

```
    If val(Mid(tec2, 1, 1)) < KeyAscii - 48 Then
```

```
        KeyAscii = 0
```

```
    End If
```

```
    Case Else
```

```
    KeyAscii = 0
```

```
    End Select
```

```
Else
```

```
    If a = 1 Then
```

```
        If tec1 = "0" Or tec1 = "1" Then
```

```
            Select Case KeyAscii
```

```
            Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8
```

```
            If val(Mid(tec2, 1, 1)) = val(Mid(tec1, 1, 1)) Then
```

```
                If val(Mid(tec2, 2, 1)) < KeyAscii - 48 Then
```

```
                    KeyAscii = 0
```

```
                End If
```

```
            End If
```

```
            Case Else
```

```
            KeyAscii = 0
```

```
            End Select
```

```
        Else
```

```
            Select Case KeyAscii
```

```
            Case 48, 49, 50, 51, 8
```

```
            If val(Mid(tec2, 1, 1)) = val(Mid(tec1, 1, 1)) Then
```

```

        If val(Mid(tec2, 2, 1)) < KeyAscii - 48 Then
            KeyAscii = 0
        End If
    End If
    Case Else
        KeyAscii = 0
    End Select
End If
Else
    If a = 2 Then
        Select Case KeyAscii
            Case 58, 8
            Case Else
                KeyAscii = 0
            End Select
        Else
            If a = 3 Then
                Select Case KeyAscii
                    Case 48, 49, 50, 51, 52, 53, 8
                    If Mid(tec2, 1, 2) = Mid(tec1, 1, 2) Then
                        If val(Mid(tec2, 4, 1)) < KeyAscii - 48 Then
                            KeyAscii = 0
                        End If
                    End If
                Case Else
                    KeyAscii = 0
                End Select
            Else
                If a = 4 Then
                    If Mid(tec2, 1, 4) = Mid(tec1, 1, 4) Then
                        If val(Mid(tec2, 5, 1)) <= KeyAscii - 48 Then
                            KeyAscii = 0
                        End If
                    End If
                End If
            End If
        End If
    End If
Else
    Select Case KeyAscii
        Case 8
        Case Else
            KeyAscii = 0
        End Select
    End If

```

```
Case Else
If KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Select
End If
hinicio = KeyAscii
End Function
```

```
Function hfin(KeyAscii As Integer, tec1 As String, tec2 As String) As Integer
If tec1 = "none " Then
Dim a As Integer
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 8
a = Len(tec2)
If a < 5 Then
If a = 0 Then
Select Case KeyAscii
Case 48, 49, 50
Case Else
KeyAscii = 0
End Select
Else
If a = 1 Then
If tec1 = "0" Or tec1 = "1" Then
Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8
Case Else
KeyAscii = 0
End Select
Else
Select Case KeyAscii
Case 48, 49, 50, 51, 8
Case Else
KeyAscii = 0
End Select
End If
Else
If a = 2 Then
Select Case KeyAscii
Case 58, 8
Case Else
KeyAscii = 0
End Select
Else
```

```

        If a = 3 Then
            Select Case KeyAscii
                Case 48, 49, 50, 51, 52, 53, 8
                Case Else
                    KeyAscii = 0
                End Select
            End If
        End If
    End If
End If
Else
    Select Case KeyAscii
        Case 8
        Case Else
            KeyAscii = 0
        End Select
    End If
    Case Else
        KeyAscii = 0
    End Select
Else

```

```

Select Case KeyAscii
Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58

```

```

a = Len(tec2)
If a < 5 Then
If a = 0 Then
    Select Case KeyAscii
        Case 48, 49, 50
        If val(Mid(tec1, 1, 1)) > KeyAscii - 48 Then
            KeyAscii = 0
        End If
        Case Else
            KeyAscii = 0
        End Select
    Else
        If a = 1 Then
            If tec2 = "0" Or tec2 = "1" Then
                Select Case KeyAscii
                    Case 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 8
                    If val(Mid(tec1, 1, 1)) = val(Mid(tec2, 1, 1)) Then
                        If val(Mid(tec1, 2, 1)) > KeyAscii - 48 Then
                            KeyAscii = 0
                        End If
                    End If
                End If
            Case Else

```

```

    KeyAscii = 0
End Select
Else
    Select Case KeyAscii
    Case 48, 49, 50, 51, 8
    If val(Mid(tec1, 1, 1)) = val(Mid(tec2, 1, 1)) Then
        If val(Mid(tec1, 2, 1)) > KeyAscii - 48 Then
            KeyAscii = 0
        End If
    End If
    Case Else
    KeyAscii = 0
    End Select
End If
Else
    If a = 2 Then
        Select Case KeyAscii
        Case 58, 8
        Case Else
        KeyAscii = 0
        End Select
    Else
        If a = 3 Then
            Select Case KeyAscii
            Case 48, 49, 50, 51, 52, 53, 8
            If Mid(tec1, 1, 2) = Mid(tec2, 1, 2) Then
                If val(Mid(tec1, 4, 1)) > KeyAscii - 48 Then
                    KeyAscii = 0
                End If
            End If
            Case Else
            KeyAscii = 0
            End Select
        Else
            If a = 4 Then
                If Mid(tec1, 1, 4) = Mid(tec2, 1, 4) Then
                    If val(Mid(tec1, 5, 1)) >= KeyAscii - 48 Then
                        KeyAscii = 0
                    End If
                End If
            End If
        End If
    End If
End If
End If
Else

```

```
Select Case KeyAscii
Case 8
Case Else
KeyAscii = 0
End Select
End If
Case Else
If KeyAscii = 8 Then
Else
KeyAscii = 0
End If
End Select
End If
hfin = KeyAscii
End Function
```

```
Private Sub chequeo()
If Len(Text3.Text) = 5 Then
Else
Text3.Text = "none "
End If
If Len(Text4.Text) = 5 Then
Else
Text4.Text = "none "
End If
End Sub
Private Sub txtfin_KeyPress(ByVal KeyAscii As Integer)
KeyAscii = hfin(KeyAscii, txtinicio.Text, txtfin.Text)
End Sub
Private Sub txtinicio_Click()
Call chequeo
txtinicio.Text = ""
End Sub
Private Sub txtfin_Click()
Call chequeo
txtfin.Text = ""
End Sub
Private Sub txtinicio_KeyPress(ByVal KeyAscii As Integer)
KeyAscii = inicio(KeyAscii, txtinicio.Text, txtfin.Text)
End Sub
Private Sub text3_KeyPress(KeyAscii As Integer)
KeyAscii = inicio(KeyAscii, Text3.Text, Text4.Text)
End Sub
Private Sub Text3_Click()
Call chequeo
Text3.Text = ""
```

```

End Sub
Private Sub text4_KeyPress(KeyAscii As Integer)
KeyAscii = hfin(KeyAscii, Text3.Text, Text4.Text)
End Sub
Private Sub Text4_Click()
Call chequeo
Text4.Text = ""
End Sub
Private Sub sincronizacion()
Text1.Text = "I"           'comando de inicio
Call enviar
Text1.Text = "H"           'alerta de comunicacion
Call enviar
Call horam
Call enviar
If Mid(Date, 2, 1) = "/" Then
Text1.Text = "0" & Mid(Date, 1, 2)
If Mid(Date, 4, 1) = "/" Then
Text1.Text = Text1.Text & "0" & Mid(Date, 3, 4)
Else
Text1.Text = Text1.Text & Mid(Date, 3, 5)
End If
Else
Text1.Text = Mid(Date, 1, 3)
If Mid(Date, 5, 1) = "/" Then
Text1.Text = Text1.Text & "0" & Mid(Date, 4, 4)
Else
Text1.Text = Text1.Text & Mid(Date, 4, 5)
End If
End If
Call enviar
End Sub
Private Sub enviar()
a = Len(Text1.Text)
For j = 1 To a
Comm1.Output = Mid(Text1.Text, j, 1)
For i = 1 To 1000
Next i
Next j
Comm1.Output = Chr(13)
For i = 1 To 1000
Next i
Text1.Text = ""
Comm1.OutBufferCount = 0
End Sub
Private Sub horam()
If Right(Time, 2) = "PM" Then

```

```
If Mid(Time, 1, 1) = "1" Then
    Text1.Text = "13" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "2" Then
    Text1.Text = "14" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "3" Then
    Text1.Text = "15" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "4" Then
    Text1.Text = "16" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "5" Then
    Text1.Text = "17" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "6" Then
    Text1.Text = "18" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "7" Then
    Text1.Text = "19" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "8" Then
    Text1.Text = "20" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 1) = "9" Then
    Text1.Text = "21" & Mid(Time, 2, 6)
End If
If Mid(Time, 1, 2) = "10" Then
    Text1.Text = "22" & Mid(Time, 3, 6)
End If
If Mid(Time, 1, 2) = "11" Then
    Text1.Text = "23" & Mid(Time, 3, 6)
End If
If Mid(Time, 1, 2) = "12" Then
    Text1.Text = Mid(Time, 1, 8)
End If
Else
    If Mid(Time, 1, 2) = "12" Then
        Text1.Text = "00" & Mid(Time, 3, 6)
    Else
        If Mid(Time, 2, 1) = ":" Then
            Text1.Text = "0" & Mid(Time, 1, 7)
        Else
            Text1.Text = Mid(Time, 1, 8)
        End If
    End If
End If
End If
```

```

Dim hn As Integer
Dim mn As Integer
Dim sn As Integer
hn = val(Mid(Text1.Text, 7, 2))
hn = hn + 2
If hn < 60 Then
    If hn < 10 Then
        Text1.Text = Mid(Text1.Text, 1, 6) & "0" & hn
    Else
        Text1.Text = Mid(Text1.Text, 1, 6) & hn
    End If
Else
    hn = hn - 60
    mn = val(Mid(Text1.Text, 4, 2))
    mn = mn + 1
    If mn < 60 Then
        If mn < 10 Then
            Text1.Text = Mid(Text1.Text, 1, 3) & "0" & mn & ":" & "0" & hn
        Else
            Text1.Text = Mid(Text1.Text, 1, 3) & mn & ":" & "0" & hn
        End If
    Else
        sn = val(Mid(Text1.Text, 1, 2))
        mn = mn - 60
        sn = sn + 1
        If sn < 24 Then
            If sn < 10 Then
                Text1.Text = "0" & sn & ":" & "0" & mn & ":" & "0" & hn
            Else
                Text1.Text = sn & ":" & "0" & mn & ":" & "0" & hn
            End If
        Else
            sn = sn - 24
            Text1.Text = "0" & sn & ":" & "0" & mn & ":" & "0" & hn
        End If
    End If
End If
End Sub
Private Sub Tmrpuerto_Timer()
Dim valc As Integer
If Comm1.CTSHolding = False Then
    valc = val(conta.Caption) + 1
    conta.Caption = valc
    If conta.Caption = "7" Then
        MsgBox "El dispositivo se desconecto o esta fuera del area de servicio",
vbCritical, "Dispositivo no disponible"
        Call Picbtn2_Click
    End If
End If
End Sub

```

```

        conta.Caption = "0"
        Tmprpuerto.Enabled = False
    End If
End If
End Sub
Private Sub Cmbhorario_Click()
    Cmbhora = cmbhorario.ListIndex
    Call Lista
End Sub
Private Sub Lista()          'si se escoje un horario de una carga
    If Cmbhora = 0 Then      'se verifica que dato se pide
        Text1.Text = "C1"
        Call enviar          'se envia el comando S sincronizacion con el
portico PB1
    End If
    If Cmbhora = 1 Then
        Text1.Text = "C2"
        Call enviar
    End If
    If Cmbhora = 2 Then
        Text1.Text = "C3"
        Call enviar
    End If
    If Cmbhora = 3 Then
        Text1.Text = "C4"
        Call enviar
    End If
    If Cmbhora = 4 Then
        Text1.Text = "C5"
        Call enviar
    End If
    If Cmbhora = 5 Then
        Text1.Text = "C6"
        Call enviar
    End If
    If Cmbhora = 6 Then
        Text1.Text = "C7"
        Call enviar
    End If
    If Cmbhora = 7 Then
        Text1.Text = "C8"
        Call enviar
    End If
    If Cmbhora = 8 Then
        Text1.Text = "C9"
        Call enviar
    End If

```

```

    If Cmbhora = 9 Then
        Text1.Text = "CA"
        Call enviar
    End If
End Sub
Modulo Shell

```

Option Explicit

```

Declare Function MyCreateProcess Lib "Coredll" Alias "CreateProcessW" (ByVal
lpApplicationName As String, ByVal lpCommandLine As String, ByVal
lpProcessAttributes As Long, ByVal lpThreadAttributes As Long, ByVal
blInheritHandles As Long, ByVal dwCreationFlags As Long, ByVal lpEnvironment
As Long, ByVal lpCurrentDirectory As Long, ByVal lpStartupInfo As Long, ByVal
lpProcessInformation As String) As Long
Declare Function MyGetLastError Lib "Coredll" Alias "GetLastError" () As Long
Declare Function MyCloseHandle Lib "Coredll" Alias "CloseHandle" (ByVal hObject
As Long) As Long
Function Shell(ByVal Application As String, ByVal Parameters As String) As Long

```

```

    Dim lsPI As String, llResult As Long, lhProcess As Long, lhThread As Long

```

```

    lsPI = LongToBytes(0) & LongToBytes(0) & LongToBytes(0) & LongToBytes(0)
    llResult = MyCreateProcess(Application, Parameters, 0, 0, 0, 0, 0, 0, 0, lsPI)
    If llResult <> 0 Then
        lhThread = BytesToLong(MidB(lsPI, 5, 4))
        Call MyCloseHandle(lhThread)
        lhProcess = BytesToLong(MidB(lsPI, 1, 4))
        Call MyCloseHandle(lhProcess)
        llResult = 0
    Else
        llResult = MyGetLastError()
        If llResult = 0 Then llResult = -1
    End If
    Shell = llResult

```

```

End Function
Function LongToBytes(ByVal Value As Long) As String

```

```

    Dim lsHex As String, i As Integer

```

```

    lsHex = Right("00000000" & Hex(Value), 8)
    For i = 1 To 7 Step 2
        LongToBytes = ChrB(CInt("&H" & Mid(lsHex, i, 2))) & LongToBytes
    Next

```

```

End Function

```

Function BytesToLong(ByVal Value As String) As Long

```
    Dim IsHex As String, i As Integer

    For i = 1 To 4
        IsHex = Hex(AscB(MidB(Value, i, 1))) & IsHex
    Next
    BytesToLong = CLng("&H" & IsHex)
```

End Function

Modulo Full Screen

```
*****
' Implements Full Screen declarations and functions
*****
' FileName: basFullScreen.bas
' Creator: Christian Forsberg
' Created: 2001-02-05
*****
' Version  Date  Who Comment
' 00.00.000 010205 CFO Created
*****
```

Option Explicit

```
' Full screen constants
Private Const SW_HIDE = 0
Private Const SW_SHOW = 5
Private Const SHFS_SHOWTASKBAR = &H1
Private Const SHFS_HIDETASKBAR = &H2
Private Const SHFS_SHOWSIPBUTTON = &H4
Private Const SHFS_HIDESIPBUTTON = &H8
Private Const SHFS_SHOWSTARTICON = &H10
Private Const SHFS_HIDESTARTICON = &H20

' API declarations
Declare Function ShowWindow Lib "Coredll" (ByVal hWnd As Long, ByVal nCmdShow As Long) As Long
Declare Function FindWindow Lib "Coredll" Alias "FindWindowW" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
Declare Function MoveWindow Lib "Coredll" (ByVal hWnd As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long
Declare Function SetForegroundWindow Lib "Coredll" (ByVal hWnd As Long) As Boolean
```

```
Declare Function SHFullScreen Lib "aygshell" (ByVal hwndRequester As Long,
ByVal dwState As Long) As Boolean
Public Sub FormFullScreen(ByVal hWnd As Long, ByVal FullScreen As Boolean)
```

```
' Make Form show in full screen (or turn full screen off).
```

```
' IN: hWnd, handle of Form window
```

```
' FullScreen, to turn on full screen or not (True/False)
```

```
' Known bugs:
```

```
' Version Date Who Comment
```

```
' 00.00.000 010205 CFO Created
```

```
*****
```

```
Dim lhWnd
```

```
If FullScreen Then
```

```
lhWnd = FindWindow("menu_worker", "")
ShowWindow lhWnd, SW_HIDE
```

```
lhWnd = FindWindow("HHTaskBar", "")
ShowWindow lhWnd, SW_HIDE
```

```
SetForegroundWindow hWnd
SHFullScreen hWnd, SHFS_HIDESIPBUTTON + SHFS_HIDETASKBAR +
SHFS_HIDESTARTICON
MoveWindow hWnd, 0, 0, 240, 320, 0
```

```
Else
```

```
SHFullScreen hWnd, SHFS_SHOWSTARTICON
```

```
lhWnd = FindWindow("menu_worker", "")
ShowWindow lhWnd, SW_SHOW
lhWnd = FindWindow("HHTaskBar", "")
ShowWindow lhWnd, SW_SHOW
```

```
End If
```

```
End Sub
```

```
Modulo BasSound
```

```
*****
```

```
' Implements Sound declarations and functions
```

```
*****
```

```
' FileName: basSound.bas
```

```
' Creator: Christian Forsberg
```

```

' Created: 2001-02-05
!*****

' Version  Date  Who Comment
' 00.00.000 010205 CFO Created
!*****

Option Explicit

' Declarations for Sound
Public Declare Function PlaySound Lib "coredll.dll" Alias "PlaySoundW" (ByVal
lpzName As String, ByVal hModule As Long, ByVal dwFlags As Long) As Long

' Sound constants
Public Const SND_ALIAS = &H10000
Public Const SND_ALIAS_ID = &H110000
Public Const SND_ALIAS_START = 0
Public Const SND_APPLICATION = &H80
Public Const SND_ASYNC = &H1
Public Const SND_FILENAME = &H20000
Public Const SND_LOOP = &H8
Public Const SND_MEMORY = &H4
Public Const SND_NODEFAULT = &H2
Public Const SND_NOSTOP = &H10
Public Const SND_NOWAIT = &H2000
Public Const SND_PURGE = &H40
Public Const SND_RESERVED = &HFF000000
Public Const SND_RESOURCE = &H40004
Public Const SND_SYNC = &H0
Public Const SND_TYPE_MASK = &H170007
Public Const SND_VALID = &H1F
Public Const SND_VALIDFLAGS = &H17201F
Public Sub PlaySoundFile(ByVal FileName As String)

' Play a sound (WAV) file.
' IN: FileName, name of WAV-file
' Known bugs:
' Version  Date  Who Comment
' 00.00.000 010205 CFO Created
!*****

    Dim I As Long

    I = PlaySound(App.Path + "\" + FileName, 0, SND_FILENAME + SND_SYNC +
SND_NODEFAULT)

End Sub

Modulo BasListView
!*****

```

```

' Implements ListView declarations and functions
*****

' FileName: basListView.bas
' Creator: Christian Forsberg
' Created: 2001-02-05
*****

' Version  Date  Who Comment
' 00.00.000 010205 CFO Created
*****

Option Explicit

' API declarations
Public Declare Function GetFocus Lib "Coredll" () As Long
Public Declare Function SendMessage Lib "Coredll" Alias "SendMessageW"
(ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal
IParam As Long) As Long
Public Declare Function SendMessageString Lib "Coredll" Alias "SendMessageW"
(ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal
IParam As String) As Long

' ListView constants
Public Const LVM_GETEXTENDEDLISTVIEWSTYLE = &H1037
Public Const LVM_SETEXTENDEDLISTVIEWSTYLE = &H1036
Public Const LVM_GETITEMSTATE = &H102C
Public Const LVM_SETITEMSTATE = &H102B
Public Const LVM_SETCOLUMNWIDTH = &H101E
Public Const LVSCW_AUTOSIZE = -1
Public Const LVSCW_AUTOSIZE_USEHEADER = -2
Public Const LVS_EX_GRIDLINES = &H1
Public Const LVS_EX_SUBITEMIMAGES = &H2
Public Const LVS_EX_CHECKBOXES = &H4
Public Const LVS_EX_TRACKSELECT = &H8
Public Const LVS_EX_HEADERDRAGDROP = &H10
Public Const LVS_EX_FULLROWSELECT = &H20 ' applies to report mode only
Public Const LVS_EX_ONECLICKACTIVATE = &H40
Public Const LVIS_STATEIMAGEMASK = &HF000
Public Sub SetFullRowSelect(hWnd As Long)

' Set a ListView to full row select and enable headers to be moved.
' IN: hWnd, hWnd to ListView control (retrieved with GetFocus function)
' Known bugs:
' Version  Date  Who Comment
' 00.00.000 010205 CFO Created
*****

Dim IStyle As Long

IStyle = SendMessage(hWnd, LVM_GETEXTENDEDLISTVIEWSTYLE, 0, 0)

```

```
    IStyle = IStyle Or LVS_EX_FULLROWSELECT Or  
    LVS_EX_HEADERDRAGDROP  
    Call SendMessage(hWnd, LVM_SETEXTENDEDLISTVIEWSTYLE, 0, IStyle)
```

```
End Sub
```

```
Public Sub AutoSizeColumns(lvw As ListViewCtrl)
```

```
' Autosize all columns in ListView.
```

```
' IN: lvw, ListView control
```

```
' Known bugs:
```

```
' Version   Date   Who Comment
```

```
' 00.00.000 010205 CFO Created
```

```
*****
```

```
    Dim I As Long
```

```
    Dim hWnd As Long
```

```
' Get ListView control windows handle
```

```
lvw.SetFocus
```

```
hWnd = GetFocus()
```

```
' Autosize columns
```

```
For I = 0 To lv.ColumnHeaders.Count - 1
```

```
    Call SendMessage(hWnd, LVM_SETCOLUMNWIDTH, I, LVSCW_AUTOSIZE)
```

```
Next
```

```
End Sub
```

```
Public Sub SetCheckBoxes(hWnd As Long)
```

```
' Set a ListView to have CheckBoxes.
```

```
' IN: hWnd, hWnd to ListView control (retrieved with GetFocus function)
```

```
' Known bugs:
```

```
' Version   Date   Who Comment
```

```
' 00.00.000 010301 CFO Created
```

```
*****
```

```
    Dim IStyle As Long
```

```
IStyle = SendMessage(hWnd, LVM_GETEXTENDEDLISTVIEWSTYLE, 0, 0)
```

```
IStyle = IStyle Or LVS_EX_CHECKBOXES
```

```
Call SendMessage(hWnd, LVM_SETEXTENDEDLISTVIEWSTYLE, 0, IStyle)
```

```
End Sub
```

```
Public Function GetCheckState(hWnd As Long, Index As Long) As Boolean
```

```
' Get the state of a CheckBox in a ListView.
```

```
' IN: hWnd, hWnd to ListView control (retrieved with GetFocus function)
```

```
'     Index, ListIndex in ListView
```

```
' OUT: GetCheckState, checkbox state (True/False)
```

```

' Known bugs:
' Version   Date   Who Comment
' 00.00.000 010301 CFO Created
'*****

    GetCheckState = CInt((SendMessage(hWnd, LVM_GETITEMSTATE, Index - 1,
LVIS_STATEIMAGEMASK) / 2 ^ 12) - 1) > 0

End Function
Public Sub SetCheckState(hWnd As Long, Index As Long, State As Boolean)

' Set a ListView to have CheckBoxes.
' IN: hWnd, hWnd to ListView control (retrieved with GetFocus function)
'     Index, ListIndex in ListView
'     State, set or clear checkbox (True/False)
' Known bugs:
' Version   Date   Who Comment
' 00.00.000 010301 CFO Created
'*****

    Dim lsLVITEM As String
    Dim llResult As Long
    Dim liStateValue As Integer

' Set state value
If State Then
    liStateValue = &H2000
Else
    liStateValue = &H1000
End If

' Set members "state" and "stateMask"
'.state = liStateValue
'.stateMask = LVIS_STATEIMAGEMASK
lsLVITEM = LongToBytes(0) & LongToBytes(0) & LongToBytes(0) & _
    LongToBytes(liStateValue) & LongToBytes(LVIS_STATEIMAGEMASK) &
-
    LongToBytes(0) & LongToBytes(0) & _
    LongToBytes(0) & LongToBytes(0) & LongToBytes(0)

    llResult = SendMessageString(hWnd, LVM_SETITEMSTATE, i - 1, lsLVITEM)

End Sub
Public Sub ClearAllCheckState(hWnd As Long)

' 00.00.000 010301 CFO Created
'*****

```

```
Dim IsLVITEM As String
Dim IIResult As Long
```

```
IsLVITEM = LongToBytes(0) & LongToBytes(0) & LongToBytes(0) & _
    LongToBytes(&H1000) & LongToBytes(LVIS_STATEIMAGEMASK) & _
    LongToBytes(0) & LongToBytes(0) & _
    LongToBytes(0) & LongToBytes(0) & LongToBytes(0)
```

```
IIResult = SendMessageString(hWnd, LVM_SETITEMSTATE, -1, IsLVITEM)
```

```
End Sub
```