



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO
PARA CONTROL Y SEGURIDAD EN TIEMPO REAL VÍA
TELÉFONO CELULAR”**

**TESIS DE GRADO PREVIA LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN**

Presentado por:

JORGE ROBERTO ALVARADO CADENA

CÉSAR OSWALDO ARÉVALO CORONEL

RIOBAMBA – ECUADOR

2010

Agradecimiento

Al Ing. Daniel Haro

Al Ing. Hugo Moreno

Director y miembro de TESIS,
por su ayuda y colaboración para
la realización de este trabajo

Dedicatoria

A nuestros padres y hermanos por su apoyo incondicional.

NOMBRE

FIRMA

FECHA

**Dr. Romeo Rodríguez
DECANO DE LA FACULTAD
INFORMÁTICA Y ELECTRÓNICA**

**Ing. Paúl Romero
DIRECTOR DE LA ESCUELA DE
INGENIERÍA ELECTRÓNICA**

**Ing. Daniel Haro
DIRECTOR TESIS**

**Ing. Hugo Moreno
MIEMBRO TRIBUNAL**

**Tlgo. Carlos Rodríguez
DIRECTOR CENTRO
DOCUMENTACIÓN**

NOTA DE LA TESIS

“Nosotros CÉSAR OSWALDO ARÉVALO CORONEL y JORGE ROBERTO ALVARADO CADENA somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO”.

CÉSAR OSWALDO
ARÉVALO CORONEL

JORGE ROBERTO
ALVARADO CADENA

ÍNDICE DE ABREVIATURAS

3G: Tercera Generación

API: Application Program Interface (Interfaz de programación de aplicaciones)

AT: Attention (Atención)

AWT: Abstract Windows Toolkit

CDC: Connected Device Configuration (Configuración de dispositivos con conexión)

CLDC: Connected Limited Device Configuration (Configuración de dispositivos con conexiones de red limitada)

CORBA: Common Object Request Broker Architecture (arquitectura común de intermediarios en peticiones a objetos)

CPU: Central Processing Unit (Unidad de proceso central)

CVM: Compact Virtual Machine. (Maquina virtual compacta)

EEPROM: Electrically Erasable Programmable Read Only Memory (Memoria de solo lectura programable y borrrable eléctricamente)

ETSI: European Telecommunications Standars Institute (Organización de estandarización de la industria de las telecomunicaciones de Europa)

FDMA: Frequency Division Multiple Access (Acceso Múltiple por División de Frecuencia)

GFC: Generic Connection Framework. (Marco general de conexión)

GPR: General Purpose Register. (Registros de Propósito General)

GPS: Global Positioning System (Sistema de posicionamiento mundial)

GPRS: General Packet Radio Service (Servicio general de paquetes vía radio)

GSM: Global System for Mobile (Sistema global para las comunicaciones)

IDS: Identification System (señal de identificación del sistema)

IP: Internet Protocol (Protocolo de Internet)

J2SE: Java Platform, Standard Edition

J2EE: Java Platform, Enterprise Edition

JCP: Java Community Process.

JIT: Just In Time (compilación al vuelo)

JME: Java Platform Micro Edition

JVM: Java Virtual Machine (Maquina virtual de Java).

JVMDI: Interfaz de depuración de la Máquina Virtual

LCD: Liquid Crystal Display (Pantalla de cristal líquido)

MIDP: Mobile Information Device Profile. (Perfil para dispositivos de información móvil)

MMS: Multimedia Messaging System (Servicio de mensajes multimedia)

MSSP: Master Synchronous Serial Port (Puerto Serie Síncrono Maestro)

MTSO: Oficina de Conmutación de Telefonía Móvil.

OC: Onda corta.

P2P: Peer to Peer (Red entre pares)

PDA: Personal Digital Assistant (Asistente digital personal)

PIC: Peripheral Interface Controller (Controlador de interfaz periférico)

PWM: Pulse Width Modulation (Modulación de Ancho de Impulsos)

RAM: Random Access Memory (Memoria de acceso aleatorio)

RISC: Reduced Instruction Set Computer (Computadores de juego de instrucciones reducido)

ROM: Read Only Memory (Memoria de solo lectura)

SFR: Special Function Register. (Registros de Funciones Especiales)

SMS: Short Message Service (Servicio de mensajes cortos)

TDMA: Time Division Multiple Access (Acceso múltiple por división de tiempo)

UMTS: Universal Mobile Telecommunications System (Sistema Universal de Telecomunicaciones Móviles)

USART: Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal).

VM: Virtual Machine (Máquina Virtual)

WAP: Wireless Application Protocol (Protocolo de aplicaciones inalámbricas)

WiFi: Wireless Fidelity.

WMA: Wireless Messaging API (API de mensajería inalámbrica)

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura II.1. Aplicaciones típicas de un sistema domótico | 21 |
| Figura II.2. Pines del PIC 16F628A | 26 |
| Figura II.3. Pines del PIC 16F877A | 28 |
| Figura II.4. Formato de transmisión asíncrona | 33 |
| Figura II.5. División de un área en celdas o células. | 40 |
| Figura II.6. Reutilización de frecuencias en celdas no adyacentes..... | 41 |
| Figura II.7. Típica torre de transmisión de telefonía celular | 44 |
| Figura II.8. Funcionamiento de la red celular..... | 45 |
| Figura II.9. Relación entre las APIs de la plataforma Java..... | 57 |
| Figura II.10. Algoritmo de verificación de clases | 60 |
| Figura II.11. Arquitectura del entorno de ejecución de JME..... | 66 |
| Figura II.12. Componentes del Wireless Messaging API..... | 71 |
| Figura II.13. El MessageConnection y su relación con el CFG | 74 |
| Figura III.14. Esquema detallado de dispositivos conectados al sistema | 76 |
| Figura III.15. Esquema general del sistema..... | 79 |
| Figura III.16. Diagramas de Flujo del Sistema..... | 81 |
| Figura III.17. Conector y pines de datos del nokia 5220..... | 82 |
| Figura III.18. Esquema de conexiones de teclado y lcd | 83 |
| Figura III.19. Etapa de comunicación..... | 84 |
| Figura III.20. Etapa de potencia..... | 85 |
| Figura III.21. Diagrama de Flujo del Funcionamiento del Software..... | 87 |
| Figura IV.22. Diseño de circuito impreso..... | 90 |
| Figura IV.23. Ubicación de elementos en la central del sistema. | 91 |

| | |
|--|-----|
| Figura IV.24. Interface software de usuario. | 92 |
| Figura IV.25. Mensaje de advertencia en el software celular..... | 93 |
| Figura IV.26. Ingreso de clave personal..... | 93 |
| Figura IV.27. Solicitud de confirmación para envío de SMS..... | 94 |
| Figura IV.28. Interface software de usuario. | 94 |
| Figura IV.29. Vivienda en que se instaló el sistema..... | 95 |
| Figura IV.30. Conexión entre el panel de control y la central..... | 96 |
| Figura IV.31. Central del sistema. | 96 |
| Figura IV.32. Panel de control..... | 97 |
| Figura IV.33. Contacto magnético..... | 97 |
| Figura IV.34. Sensor infrarrojo de movimiento. | 98 |
| Figura IV.35. Sirena..... | 98 |
| Figura IV.36. Iluminación conectada al sistema..... | 99 |
| Figura IV.37. Conexiones para cerradura eléctrica. | 99 |
| Figura IV.38. Central y panel de control. | 100 |
| Figura IV.39. Ubicación del panel de control..... | 100 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla II.1. Características del PIC 16F877A..... | 29 |
| Tabla II.2. Librerías de CDC | 63 |
| Tabla II.3. Librerías de configuración CLDC..... | 64 |
| Tabla II.4 Librerías del Foundation Profile. | 67 |
| Tabla II.5 Librerías del Personal Profile..... | 68 |
| Tabla II.6 Librerías del perfil MIDP..... | 69 |
| Tabla II.7. Contenido del paquete javax.wireless.messaging | 72 |
| Tabla III.8. Detalle de pines del bus de datos del Nokia 3220 | 83 |

ÍNDICE

CAPÍTULO I

| | |
|------------------------------------|----|
| 1.1. ANTECEDENTES | 14 |
| 1.2. JUSTIFICACIÓN | 15 |
| 1.3. OBJETIVOS | 16 |
| 1.3.1. OBJETIVO GENERAL | 16 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 16 |
| 1.4. HIPÓTESIS | 17 |

CAPÍTULO II

| | |
|--|----|
| 2. FUNDAMENTO TEÓRICO | 18 |
| 2.1. DOMÓTICA | 18 |
| 2.1.1. APLICACIONES DE LA DOMÓTICA | 21 |
| 2.1.2. DOMÓTICA COMO MEDIO DE SEGURIDAD | 23 |
| 2.2. MICROCONTROLADORES | 24 |
| 2.2.1. MICROCONTROLADOR 16F628A..... | 25 |
| 2.2.2. MICROCONTROLADOR PIC 16F877A | 27 |
| 2.3. COMUNICACIÓN SERIAL..... | 29 |
| 2.3.1. CONSIDERACIONES EN LA COMUNICACIÓN SERIE | 29 |
| 2.3.2. LÍNEAS O CANALES DE COMUNICACIÓN..... | 30 |
| 2.3.3. MODOS DE TRANSMISIÓN | 31 |
| 2.3.4. LA TRANSMISIÓN ASÍNCRONA | 32 |
| 2.3.5. COMANDOS AT..... | 35 |
| 2.4. TELEFONÍA CELULAR..... | 38 |
| 2.4.1. FUNCIONAMIENTO DE LA TELEFONÍA CELULAR..... | 39 |
| 2.4.2. TECNOLOGÍAS UTILIZADAS EN LOS TELÉFONOS CELULARES ... | 46 |
| 2.4.3. GPRS (General Packet Radio Service)..... | 47 |
| 2.4.4. INFLUENCIA EN LA SOCIEDAD | 48 |
| 2.5. JAVA | 49 |
| 2.5.1. FILOSOFÍA..... | 51 |
| 2.5.3. JAVA MICRO EDITION..... | 55 |
| 2.5.4. WIRELESS MESSAGING API..... | 70 |
| 2.5.5. THREAD O HILOS DE EJECUCIÓN | 75 |

CAPÍTULO III

| | |
|---------------------------------------|----|
| 3. DISEÑO DEL SISTEMA..... | 76 |
| 3.1. PLANTEAMIENTO..... | 76 |
| 3.2. REQUERIMIENTOS TECNOLÓGICOS..... | 78 |

| | | |
|-----------------|--------------------------------------|-----|
| 3.2.1. | HARDWARE..... | 78 |
| 3.2.2. | SOFTWARE..... | 79 |
| 3.3. | DISEÑO DE HARDWARE | 79 |
| 3.3.1. | SISTEMA DE ADQUISICIÓN DE DATOS..... | 81 |
| 3.3.2. | ETAPA DE COMUNICACIÓN | 83 |
| 3.3.3. | ETAPA DE CONTROL..... | 84 |
| 3.3.4. | ETAPA DE EJECUCIÓN | 85 |
| 3.3.5. | ETAPA DE POTENCIA | 85 |
| 3.4. | DISEÑO DE SOFTWARE..... | 86 |
| 3.4.1. | INTERFACE SOFTWARE DE USUARIO | 86 |
| 3.4.2. | LISTADO DE ACCIONES..... | 88 |
| CAPÍTULO IV | | |
| 4. | IMPLEMENTACIÓN DEL SISTEMA..... | 89 |
| 4.1. | IMPLEMENTACIÓN DEL HARDWARE..... | 90 |
| 4.1.1. | CIRCUITO IMPRESO..... | 90 |
| 4.1.2. | DISPOSICIÓN DE ELEMENTOS | 91 |
| 4.2. | IMPLEMENTACIÓN DEL SOFTWARE..... | 91 |
| 4.3. | INSTALACIÓN..... | 95 |
| 4.4. | PRUEBAS | 101 |
| CONCLUSIONES | | |
| RECOMENDACIONES | | |
| RESUMEN | | |
| BIBLIOGRAFÍA | | |

CAPÍTULO I

1.1.ANTECEDENTES

El control domótico ha estado restringido hasta hace poco solo a tareas pre-programadas y al control de ciertas tareas del hogar desde un computador lo cual no permite flexibilidad en el control de dichas tareas. Además de que las empresas que proveen dispositivos y realizan la instalación de este tipo de servicios lo hacen a costos muy elevados.

Mediante la utilización de la tecnología GPRS se mejora y actualiza GSM con servicios como:

- Servicio de mensajes multimedia (MMS)
- Mensajería instantánea
- Aplicaciones en red para dispositivos a través del protocolo WAP
- Servicios P2P utilizando el protocolo IP

- Servicio de mensajes cortos (SMS)
- Posibilidad de utilizar el dispositivo como módem USB

Posibilitando el envío y recepción de datos por medio del celular de manera rápida y económica.

La creciente inseguridad ciudadana, hace que cada día surjan nuevas formas de mejorar la seguridad de nuestra familia y hoy la tecnología nos da las herramientas necesarias. Al punto de permitir el control domótico a partir de dispositivos que son de uso común, y que proporcionan alta velocidad, con muchas prestaciones como es el área de cobertura, bajo costo, etc.

Dentro del hogar existen varios dispositivos que aceptan control domótico como son porteros eléctricos, sistema de iluminación, calefacción / aire acondicionado, sistema de aspersión, sistema de seguridad, etc.

1.2.JUSTIFICACIÓN

Con la implementación de este sistema se mejora la forma en que se realiza el control domótico al poder controlar siempre y poder conocer el estado del mismo por medio del teléfono celular, sin importar en donde se encuentre el usuario y con una interface sencilla.

El desarrollo de software para celular usando WMA (Wireless Messaging API) de Java Micro Edition proporciona la capacidad de ofrecer al usuario del sistema una interface sencilla pero eficaz para el control del sistema domótico y mediante la investigación profundizar los conocimientos en programación celular.

Hoy en día la tecnología provee un medio confiable y de amplia cobertura para el envío/recepción de datos sobre la red de telefonía celular, puesto además que el uso de teléfonos celulares en día se ha masificado.

Se proporciona una herramienta para brindar seguridad, comodidad y rapidez en el control domótico, prestando un servicio a favor del mejoramiento de la calidad de vida de la comunidad, y con un costo bajo.

1.3.OBJETIVOS

1.3.1. OBJETIVO GENERAL

Diseñar e implementar un sistema domótico para control y seguridad domiciliaria, en tiempo real vía teléfono celular.

1.3.2. OBJETIVOS ESPECÍFICOS

- Estudiar dispositivos sensores, actuadores y microcontroladores para el control de puertas, luces, y seguridad de una vivienda vía teléfono celular.
- Analizar y diseñar el software de usuario para la comunicación con el sistema de control de la vivienda.
- Diseñar e implementar el hardware del equipo que permita la interacción entre el teléfono celular y la interfaz de potencia que controle al menos cuatro dispositivos.

1.4.HIPÓTESIS

El diseño e implementación de un sistema de control vía teléfono celular permitirá ofrecer una alternativa en tiempo real para el control domótico.

CAPÍTULO II

2. FUNDAMENTO TEÓRICO

2.1.DOMÓTICA

La domótica puede definirse como la adopción, integración y aplicación de las nuevas tecnologías informáticas y comunicativas al hogar. Incluye principalmente el uso de electricidad, dispositivos electrónicos, sistemas informáticos y diferentes dispositivos de telecomunicaciones, incorporando la telefonía móvil e Internet. Algunas de sus principales características son: interacción, interrelación, facilidad de uso, teleoperación o manejo a distancia, fiabilidad, y capacidad de programación y actualización.

Tradicionalmente se dice que un edificio inteligente es aquel que descansa sobre los cuatro pilares siguientes:

Automatización de funciones:

Con lo que se pretende proporcionar la mayor autonomía funcional posible al edificio. Es decir, el edificio debe ser capaz de controlar, por sí mismo, todas las instalaciones que pueda incorporar para que se satisfagan las necesidades de los usuarios. Estas funciones son tales como seguridad, control de presencia, climatización, depuración de aire, abastecimiento de agua, ascensores, iluminación, sistemas anti-incendios, apertura automática de puertas, ventanas, toldos, persianas, etc.

Automatización de actividades:

Entendiéndose con ello que el edificio debería incorporar la infraestructura necesaria para dar soporte a la actividad que se vaya a desarrollar en el mismo. Así, debería incorporar todo tipo de tomas de alimentación, de transmisión de datos, audio, vídeo y de control, así como los dispositivos que sean necesarios para cualquier trabajador (fax, modem, etc.), teniendo en cuenta incluso el diseño ergonómico, desde su puesto físico de trabajo, hasta el sistema operativo de los sistemas de procesado que se vaya a utilizar. Todo ello unido e integrado por medio de redes locales y globales.

Telecomunicaciones avanzadas:

Entendiendo como tales aquellas que permiten transmitir cualquier tipo de información multimedia (audio, vídeo, datos, señales de control, etc...) de la que se pueda hacer uso en el edificio. Para conseguir esto hacen falta, como mínimo, una serie de sistemas: Cableado estructurado, control de servicios técnicos y seguridad, televisión en circuito cerrado, telefonía interior, intercomunicación, megafonía y otros servicios de valor añadido.

Hay que hacer hincapié en el hecho de que el sistema de telecomunicaciones de un edificio inteligente debe ser integral (es decir, debe permitir que por el mismo terminal se acceda a cualquier punto tanto interior como exterior al propio edificio) y ampliable a exigencias futuras, sin necesidad de realizar ningún tipo de modificación en la estructura del cableado.

Para conseguir todo esto son necesarias una homogenización del cableado del edificio en un solo sistema, flexibilidad máxima que posibilite añadir y modificar servicios sin cambiar el cableado y, finalmente, una estructuración y dimensionamiento del sistema para que se puedan aprovechar al máximo las ventajas que ofrezcan las futuras autopistas de la información.

Flexibilidad al cambio:

Éste es un aspecto fundamental en la “inteligencia” del edificio. Se trata de garantizar que el edificio sea capaz de satisfacer las necesidades de cualquier conjunto de usuarios diferentes que pueda albergar a lo largo de su vida útil.

Por ejemplo, cambiar una planta estructurada como una oficina abierta, tipo paisaje, a una serie de oficinas cerradas independientes, debe poderse hacer sin necesidad de modificar estas infraestructuras, ni de realizar ningún tipo de obras de albañilería.

Por supuesto, los puntos anteriores deben estar completamente integrados, formando parte de un ente común, en el cual se manifieste la “inteligencia” del edificio.

2.1.1. APLICACIONES DE LA DOMÓTICA



Figura II.1. Aplicaciones típicas de un sistema domótico

Los servicios o aplicaciones típicas que ofrece la domótica se muestran en la figura II.1 y pueden agrupar según cinco aspectos o ámbitos principales:

1. **Ahorro energético:** El ahorro energético no es algo tangible, sino un concepto al que se puede llegar de muchas maneras. En muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos sino una gestión eficiente de los mismos.

- Climatización: programación y zonificación.
- Gestión eléctrica:
 - Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado
 - Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida
- Uso de energías renovables

2. **Confort:** Conlleva todas las actuaciones que se puedan llevar a cabo que mejoren el confort en una vivienda. Dichas actuaciones pueden ser de carácter tanto pasivo, como activo o mixtas.

- Iluminación:
 - Apagado general de todas las luces de la vivienda
 - Automatización del apagado/ encendido 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 de fácil manejo
- Integración del portero al teléfono, o del videoportero al televisor
- Control vía Internet
- Gestión Multimedia y del ocio electrónicos
- Generación de macros y programas de forma sencilla para el usuario

3. **Seguridad:** Consiste en una red de seguridad encargada de proteger tanto los Bienes Patrimoniales y la seguridad personal.

- Simulación de presencia.
- Alarmas de Detección de incendio, fugas de gas, escapes de agua, concentración de monóxido en garajes.
- Alerta médica. Teleasistencia.
- Cerramiento de persianas puntual y seguro.
- Acceso a Cámaras IP

4. **Comunicaciones:** Son los sistemas o infraestructuras de comunicaciones que posee el hogar.
- Ubicuidad en el control tanto externo como interno, control remoto desde Internet, PC, mandos inalámbricos (p.ej. PDA con WiFi).
 - Transmisión de alarmas.
 - Intercomunicaciones.
5. **"Telegestión y Accesibilidad":** Diseño para todos, un diseño accesible para la diversidad humana, la inclusión social y la igualdad. Este enfoque constituye un reto Ético y creativo. Donde las personas con discapacidad reducida puedan acceder a estas tecnologías sin temor a un obstáculo del tipo de tecnología o arquitectura.

2.1.2. DOMÓTICA COMO MEDIO DE SEGURIDAD

La seguridad es uno de los factores fundamentales para quienes diseñan y adoptan los sistemas domóticos, y las innovaciones involucran varias posibilidades ya disponibles. Son destacables tres aspectos: los sensores y periféricos utilizados, el almacenamiento de la información —por ejemplo la de vídeo— y las posibilidades de control y ajuste del sistema. En el primero, además de las imágenes convencionales de las cámaras de vigilancia, se destacan los avances recientes en la detección de volúmenes, la discriminación térmica de los objetos y la percepción del movimiento. En el segundo resalta la existencia de dos nuevas posibilidades para reemplazar las clásicas cintas de VHS: el almacenamiento comprimido en discos duros, utilizando en ocasiones muy poco espacio, y la transmisión de imagen en tiempo real sin almacenamiento, al menos cuando no sea necesario. En el último aspecto se destaca la viabilidad de controlar el

sistema desde el hogar, desde cualquier sitio que posea conexión a Internet o donde exista cobertura de telefonía móvil, lo cual tiene serias implicaciones en términos de las escalas geográficas de acción.

Existen muchas aplicaciones domóticas interesantes e inquietantes para la seguridad. Por un lado, los sistemas de televigilancia han llegado a tal desarrollo que permiten ver y capturar imágenes de lo que ocurre en cualquier parte del mundo que disponga de algún tipo de conexión. Mediante el sistema *e-netcam*, además de visualizar, se pueden almacenar las imágenes en un disco duro y grabarlas posteriormente, si así se desea, en formato Cd-Rom tradicional. De otra parte, ya existe la denominada “Alarma Perro Electrónico”, que simula la presencia de alguien en casa cuando en realidad no hay nadie. Fundamentalmente el sistema funciona de la siguiente manera: un sensor de movimiento reconoce que hay alguien fuera de la casa —en la entrada, por ejemplo—, luego de un rato activa unos altavoces que emiten el sonido de un feroz perro ladrando y, posteriormente, encienden las luces de una habitación o de la sala. Efectivamente estas aplicaciones permiten simular que hay personas en casa para intentar disuadir a un posible intruso, a la vez que pueden contactar automáticamente al dueño y la policía. También ya es posible que un sistema, como el Izaxon Presence, informe sobre cuántas personas se encuentran en cierto momento en una vivienda.

2.2.MICROCONTROLADORES

El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de E/S, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de E/S a la CPU. El PIC utilizaba microcódigo simple almacenado en ROM para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador.

2.2.1. MICROCONTROLADOR 16F628A

Los PIC16F628A, pertenece a la familia de los PIC16CXX, posee 18 pines, de bajo costo con un procesador tipo RISC y segmentado, se basa en una arquitectura HARVARD.

Con estos recursos el PIC es capaz de ejecutar instrucciones solamente en un ciclo de instrucción. Con la estructura segmentada se pueden realizar simultáneamente las dos fases en que se descompone cada instrucción, ejecución de la instrucción y búsqueda de la siguiente.

La separación de los dos tipos de memoria son los pilares de la arquitectura Harvard, esto permite acceder en forma simultánea e independiente a la memoria de datos y a la de instrucciones. El tener memorias separadas permite que cada una tenga el ancho y tamaño más adecuado. Así en el PIC 16F628 el ancho de los datos es de un byte, mientras que la de las instrucciones es de 14 bits. La distribución de pines se puede observar en la figura II.2.

CARACTERÍSTICAS PRINCIPALES

- Conjunto reducido de instrucciones (RISC). Solamente 35 instrucciones que aprender a utilizar
- Oscilador interno de 4MHz
- Las instrucciones se ejecutan en un sólo ciclo de máquina excepto los saltos (*goto* y *call*), que requieren 2 ciclos. Aquí hay que especificar que un ciclo de máquina se lleva 4 ciclos de reloj, si se utiliza el reloj interno de 4MHz, los ciclos de máquina se realizarán con una frecuencia de 1MHz, es decir que cada instrucción se ejecutará en 1µs (microsegundo)
- Opera con una frecuencia de reloj de hasta 20 MHz (ciclo de máquina de 200 ns)
- Memoria de programa: 2048 locaciones de 14 bits
- Memoria de datos: Memoria RAM de 224 bytes (8 bits por registro)
- Memoria EEPROM: 128 bytes (8 bits por registro)
- Stack de 8 niveles
- 16 Terminales de I/O que soportan corrientes de hasta 25 mA
- 3 Temporizadores
- Módulos de comunicación serie, comparadores, PWM

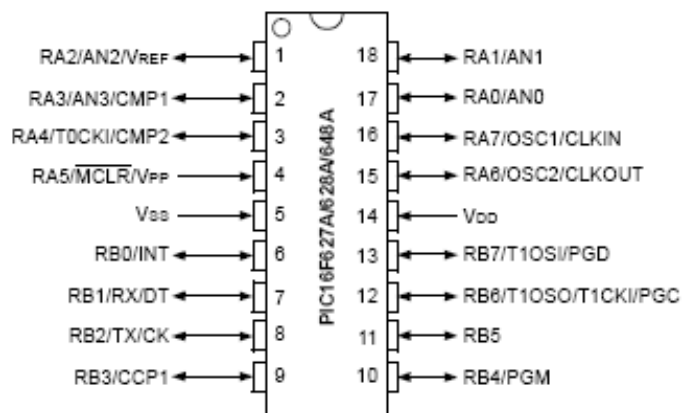


Figura II.2. Pines del PIC 16F628A

Otra característica de los PICs es el manejo de los bancos de registros. En línea general, los registros se clasifican como de uso general (GPR) y de uso específico o de funciones especiales (SFR).

- Los registros de uso general pueden ser usados directamente por el usuario, sin existir restricciones. Pueden servir para almacenar resultados que se reciben desde el registro W (acumulador), datos que provienen de las puertas de entradas, etc.
- Los registros de uso específicos no pueden ser usados directamente por el usuario. Estos registros controlan prácticamente todo el funcionamiento del microcontrolador, pues toda la configuración necesaria para funcionamiento del microcontrolador es hecho a través de algún tipo de SFR.

2.2.2. MICROCONTROLADOR PIC 16F877A

El modelo 16F877A posee varias características que hacen a este microcontrolador un dispositivo muy versátil, eficiente y práctico.

Algunas de estas características se muestran a continuación:

- Soporta modo de comunicación serial, posee dos pines para ello.
- Amplia memoria para datos y programa.
- Memoria reprogramable: La memoria en este PIC es la que se denomina FLASH; este tipo de memoria se puede borrar electrónicamente (esto corresponde a la "F" en el modelo).
- Set de instrucciones reducido (tipo RISC), pero con las instrucciones necesarias para facilitar su manejo.

Dispositivos periféricos:

- Timer0: Temporizador-contador de 8 bits con preescaler de 8 bits
- Timer1: Temporizador-contador de 16 bits con preescaler que puede incrementarse en modo sleep de forma externa por un cristal/clock.
- Timer2: Temporizador-contador de 8 bits con preescaler y postescaler.
- Dos módulos de Captura, Comparación, PWM (Modulación de Anchura de Impulsos).
- Conversor A/D de 10 bits.
- Puerto Serie Síncrono Master (MSSP) con SPI e I2C (Master/Slave).
- USART/SCI (Universal Synchronous Asynchronous Receiver Transmitter) con 9 bit.
- Puerta Paralela Esclava (PSP) solo en encapsulados con 40 pines

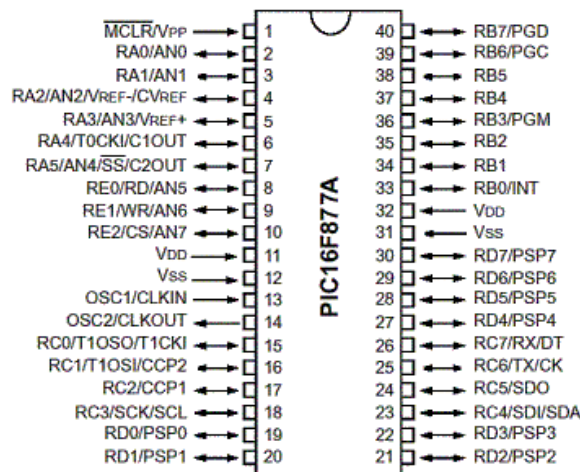


Figura II.3. Pines del PIC 16F877A

CARACTERÍSTICAS

En la tabla II.1.se pueden observar las características más relevantes del dispositivo, y su distribución de pines en la figura II.3.

| CARACTERÍSTICAS | 16F877 |
|--|------------------|
| Frecuencia máxima | DX-20MHz |
| Memoria de programa flash palabra de 14 bits | 8KB |
| Posiciones RAM de datos | 368 |
| Posiciones EEPROM de datos | 256 |
| Puertos E/S | A,B,C,D,E |
| Número de pines | 40 |
| Interrupciones | 14 |
| Timers | 3 |
| Módulos CCP | 2 |
| Comunicaciones Serie | MSSP, USART |
| Comunicaciones paralelo | PSP |
| Líneas de entrada de CAD de 10 bits | 8 |
| Juego de instrucciones | 35 Instrucciones |
| Longitud de la instrucción | 14 bits |
| Arquitectura | Harvard |
| CPU | Risc |
| Canales Pwm | 2 |
| Pila Harware | - |
| Ejecución En 1 Ciclo Máquina | - |

Tabla II.1. Características del PIC 16F877A

2.3.COMUNICACIÓN SERIAL

En telecomunicaciones y computación, la comunicación serial es el proceso de envío de datos de un bit por vez, secuencialmente, sobre un canal de comunicación o un bus de computadora. Contrasta con la comunicación paralela, donde todos los bits de cada símbolo (la más pequeña unidad de datos transmitida por vez) son enviados juntos.

La comunicación serial es utilizada en casi todas las comunicaciones y redes de computadoras, porque los costos de los cables y las dificultades de sincronización hacen a la comunicación paralela poco práctica.

2.3.1. CONSIDERACIONES EN LA COMUNICACIÓN SERIE

Cuando se transmite información a través de una línea serie es necesario utilizar un sistema de codificación que permita resolver los siguientes problemas:

1. **Sincronización de bits:** El receptor necesita saber donde comienza y donde termina cada bit en la señal recibida para efectuar el muestreo de la misma en el centro del intervalo de cada símbolo (bit para señales binarias).
2. **Sincronización del carácter:** La información serie se transmite por definición bit a bit, pero la misma tiene sentido en palabras o bytes.
3. **Sincronización del mensaje:** Es necesario conocer el inicio y fin de una cadena de caracteres por parte del receptor para, por ejemplo, detectar algún error en la comunicación de un mensaje.

2.3.2. LÍNEAS O CANALES DE COMUNICACIÓN

Se pueden establecer canales para la comunicación de acuerdo a tres técnicas, siempre tomando al microprocesador o microcontrolador como referencia (transmisor) y al periférico como destino (receptor):

- a. Simplex
- b. Semi duplex (Half duplex)
- c. Totalmente duplex (Full duplex)

Simplex: En ella la comunicación serie usa una dirección y una línea de comunicación.

Siempre existirá un transmisor y un receptor, no ambos.

La ventaja de este sistema consiste en que es necesario sólo un enlace a dos hilos.

La desventaja radica en que el extremo receptor no tiene ninguna forma de avisar al extremo transmisor sobre su estado y sobre la calidad de la información que se recibe.

Esta es la razón por la cual, generalmente, no se utiliza.

Semi duplex: La comunicación serie se establece a través de una sola línea, pero en ambos sentidos. En un momento el transmisor enviará información y en otro recibirá, por lo que no se puede transferir información en ambos sentidos de forma simultánea.

Este modo permite la transmisión desde el extremo receptor de la información, sobre el estado de dicho receptor y sobre la calidad de la información recibida por lo que permite así la realización de procedimientos de detección y corrección de errores.

Full duplex: Se utilizan dos líneas (una transmisora y otra receptora) y se transfiere información en ambos sentidos. La ventaja de este método es que se puede transmitir y recibir información de manera simultánea.

La mayoría de los dispositivos especializados para la comunicación pueden transferir información tanto en full duplex como en half duplex (el modo simplex es un caso especial dentro de half duplex).

2.3.3. MODOS DE TRANSMISIÓN

Existen dos modos básicos para realizar la transmisión de datos y son:

- Modo asíncrono.
- Modo síncrono.

Las transmisiones asíncronas son aquellas en que los bits que constituyen el código de un carácter se emiten con la ayuda de impulsos suplementarios que permiten mantener en sincronismo los dos extremos.

En las transmisiones síncronas los caracteres se transmiten consecutivamente, no existiendo ni bit de inicio ni bit de parada entre los caracteres, estando dividida la

corriente de caracteres en bloques, enviándose una secuencia de sincronización al inicio de cada bloque.

2.3.4. LA TRANSMISIÓN ASÍNCRONA

Cuando se opera en modo asíncrono no existe una línea de reloj común que establezca la duración de un bit y el caracter puede ser enviado en cualquier momento. Esto conlleva que cada dispositivo tiene su propio reloj y que previamente se ha acordado que ambos dispositivos transmitirán datos a la misma velocidad.

No obstante, en un sistema digital, un reloj es normalmente utilizado para sincronizar la transferencia de datos entre las diferentes partes del sistema. El reloj definirá el inicio y fin de cada unidad de información así como la velocidad de transmisión. Si no existe reloj común, algún modo debe ser utilizado para sincronizar el mensaje.

En realidad, la frecuencia con que el reloj muestrea la línea de comunicación es mucho mayor que la cadencia con que llegan los datos. Por ejemplo, si los datos están llegando a una cadencia de 2400 bps, el reloj examinará la línea unas 19200 veces por segundo, es decir, ocho veces la cadencia binaria. La gran rapidez con que el reloj muestrea la línea, permite al dispositivo receptor detectar una transmisión de 1 a 0 o de 0 a 1 muy rápidamente, y mantener así la mejor sincronización entre los dispositivos emisor y receptor.

El tiempo por bit en una línea en que se transfiere la información a 2400 bps es de unos 416 microsegundos ($1 \text{ seg}/2400$). Una frecuencia de muestreo de 2400 veces por segundo nos permitirá muestrear el principio o el final del bit. En ambos casos detectaremos el bit, sin embargo, no es extraño que la señal cambie ligeramente, y

permanezca la línea con una duración un poco más larga o más corta de lo normal. Por todo ello, una frecuencia de muestreo lenta no sería capaz de detectar el cambio de estado de la señal a su debido tiempo, y esto daría lugar a que la estación terminal no recibiera los bits correctamente.

Bit de inicio y bit de parada

En la transmisión asíncrona un carácter a transmitir es encuadrado con un indicador de inicio y fin de carácter, de la misma forma que se separa una palabra con una letra mayúscula y un espacio en una oración. La forma estándar de encuadrar un carácter es a través de un bit de inicio y un bit de parada.

Durante el intervalo de tiempo en que no son transferidos caracteres, el canal debe poseer un "1" lógico. Al bit de parada se le asigna también un "1". Al bit de inicio del carácter a transmitir se le asigna un "0". Por todo lo anterior, un cambio de nivel de "1" a "0" lógico le indicará al receptor que un nuevo carácter será transmitido. Todo esto se ilustra en la figura 2.4.

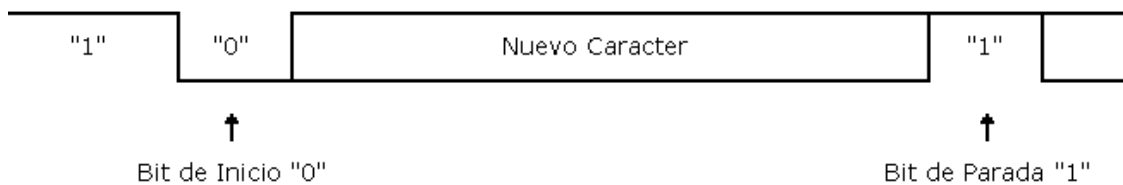


Figura II.4. Formato de transmisión asíncrona

Reglas de transmisión asíncrona

La transmisión asíncrona que vamos a ver es la definida por la norma RS232, en la que profundizaremos más adelante y que se basa en las siguientes reglas:

- a. Cuando no se envían datos por la línea, ésta se mantiene en estado alto (1).

- b. Cuando se desea transmitir un carácter, se envía primero un bit de inicio que pone la línea a estado bajo (0) durante el tiempo de un bit.
- c. Durante la transmisión, si la línea está a nivel bajo, se envía un 0 y si está a nivel alto se envía un 1.
- d. A continuación se envían todos los bits del mensaje a transmitir con los intervalos que marca el reloj de transmisión. Por convenio se transmiten entre 5 y 8 bits.
- e. Se envía primero el bit menos significativo, siendo el más significativo el último en enviarse.
- f. A continuación del último bit del mensaje se envía el bit (o los bits) del final que hace que la línea se ponga a 1 por lo menos durante el tiempo mínimo de un bit. Estos bits pueden ser un bit de paridad para detectar errores y el bit o bits de stop, que indican el fin de la transmisión de un carácter.

Los datos codificados por esta regla, pueden ser recibidos siguiendo los pasos siguientes:

- a. Esperar la transición 1 a 0 en la señal recibida.
- b. Activar el reloj con una frecuencia igual a la del transmisor.
- c. Muestrear la señal recibida al ritmo de ese reloj para formar el mensaje.
- d. Leer un bit más de la línea y comprobar si es 1 para confirmar que no ha habido error en la sincronización.

2.3.5. COMANDOS AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM.

Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías Microcomm y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.

Los comandos AT se denominan así por la abreviatura de attention.

Aunque la finalidad principal de los comandos AT es la comunicación con modems, la telefonía móvil GSM también ha adoptado como estandar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

Es claro que la implementación de los comandos AT corresponde a los dispositivos GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable de serie, canal Infrarrojos, Bluetooth, etc.

COMANDOS AT MÁS UTILIZADOS

Estos son los comandos más comunes en la mayoría de los modems y los que más se usan.

ATA

- 1) Se pone en modo respuesta y espera una señal portadora del modem remoto.
- 2) Espera S7 segundos y colgará si no se detecta portadora.

ATDnúmero

- 1) Descuelga y llama al número de teléfono solicitado.
- 2) Espera un tono de llamada antes de marcar.
 - 2.1) Si no se detecta ese tono en S6 segundos, el modem devuelve código de resultado "no dial tone"
 - 2.2) si se detecta el tono el modem espera S7 segundos
 - 2.2.1) si no establece conexión el modem vuelve al estado de comandos
 - 2.2.2) si se establece conexión el modem entra en el estado on-line.

ATE: Eco. Los comandos introducidos en el modem vuelven por eco al PC (por defecto).

ATH: Descuelga el teléfono

ATI: Revisa la ROM del modem (checksum)

ATL: Programa el volumen del altavoz

ATM: Programa conexión/desconexión del altavoz

ATO: Vuelve a estado on-line desde el estado de comandos.

ATS: Visualiza/cambia contenidos de los registros S

ATV: Envía códigos de resultado en palabras o números

ATW: Envía "códigos del progreso de la negociación"

ATX: Programa códigos de resultado

ATZ: Reset

AT&C: Programa detección de portadora

AT&D: Programa control de DTR

AT&K: Programa control de flujo

AT&W: Almacena perfil configuración del usuario

AT&Y: Especifica que perfil de configuración usuario de los almacenados se va a utilizar

1 Comandos generales

a) AT+CGMI: Identificación del fabricante

b) AT+CGSN: Obtener número de serie

c) AT+CIMI: Obtener el IMSI.

d) AT+CPAS: Leer estado del modem

2. Comandos del servicio de red

a) AT+CSQ: Obtener calidad de la señal

b) AT+COPS: Selección de un operador

c) AT+CREG: Registrarse en una red

d) AT+WOPN: Leer nombre del operador

3. Comandos de seguridad:

a) AT+CPIN: Introducir el PIN

b) AT+CPINC: Obtener el número de reintentos que quedan

c) AT+CPWD: Cambiar password

4. Comandos para la agenda de teléfonos

- a) AT+CPBR: Leer todas las entradas
- b) AT+CPBF: Encontrar una entrada
- c) AT+CPBW: Almacenar una entrada
- d) AT+CPBS: Buscar una entrada

5. Comandos para SMS

- a) AT+CPMS: Seleccionar lugar de almacenamiento de los SMS
- b) AT+CMGF: Seleccionar formato de los mensajes SMS
- c) AT+CMGR: Leer un mensaje SMS almacenado
- d) AT+CMGL: Listar los mensajes almacenados
- e) AT+CMGS: Enviar mensaje SMS
- f) AT+CMGW: Almacenar mensaje en memoria
- g) AT+CMSS: Enviar mensaje almacenado
- h) AT+CSCA: Establecer el Centro de mensajes a usar
- i) AT+ WMSC: Modificar el estado de un mensaje.

2.4.TELEFONÍA CELULAR

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

El teléfono móvil es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, cada una de las cuales es una célula, si bien existen redes telefónicas móviles satelitales. Su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de video e incluso GPS y reproductor mp3.

2.4.1. FUNCIONAMIENTO DE LA TELEFONÍA CELULAR

Los teléfonos celulares, por sofisticados que sean y luzcan, no dejan de ser radio transmisores personales.

Siendo un sistema de comunicación telefónica totalmente inalámbrica, los sonidos se convierten en señales electromagnéticas, que viajan a través del aire, siendo recibidas y transformadas nuevamente en mensaje a través de antenas repetidoras o vía satélite.

Para entender mejor cómo funcionan estos sofisticados aparatos puede ayudar compararlos con una radio de onda corta (OC) o con un walkie-talkie. Un radio OC es un aparato simple. Este permite que dos personas se comuniquen utilizando la misma frecuencia, así que sólo una persona puede hablar al tiempo.

Un teléfono celular es un dispositivo dual, esto quiere decir que utiliza una frecuencia para hablar, y una segunda frecuencia aparte para escuchar. Una radio OC tiene 40 canales. Un teléfono celular puede utilizar 1664 canales. Estos teléfonos también operan con “células” o “celdas” y pueden alternar la célula usada a medida que el teléfono es

desplazado. Las células le dan a los teléfonos un rango mucho mayor a los dispositivos que lo comparamos. Alguien que utiliza un teléfono celular, puede manejar a través de toda la ciudad y mantener la conversación todo el tiempo. Las células son las que dan a los teléfonos celulares un gran rango.

En un radio dual, los dos transmisores utilizan diferentes frecuencias, así que dos personas pueden hablar al mismo tiempo. Los teléfonos celulares son duales.

El teléfono celular estándar de la primera generación estableció un rango de frecuencias entre los 824 Megahertz y los 894 para las comunicaciones analógicas.

Para enfrentar la competencia y mantener los precios bajos, este estándar estableció el concepto de dos portadores en cada mercado, conocidos como portadores A y B. A cada portador se le da 832 frecuencias de voz, cada una con una amplitud de 30 Kilohertz. Un par de frecuencias (una para enviar y otra para recibir) son usadas para proveer un canal dual por teléfono. Las frecuencias de transmisión y recepción de cada canal de voz están separadas por 45 Megahertz. Cada portador también tiene 21 canales de datos para usar en otras actividades.

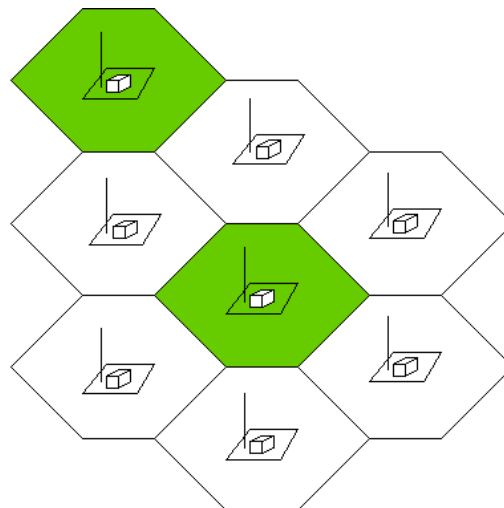


Figura II.5. División de un área en celdas o células.

La genialidad del teléfono celular reside en que una ciudad puede ser dividida en pequeñas "células" o "celdas", como se muestra en la figura II.5., que permiten extender la frecuencia por toda una ciudad. Esto es lo que permite que millones de usuarios utilicen el servicio en un territorio amplio sin tener problemas.

He aquí cómo funciona. Se puede dividir un área en células. Cada célula es típicamente de un tamaño de 10 millas cuadradas (unos 26 Km²). Las células se imaginan como unos hexágonos en un campo hexagonal grande.

Sin embargo, el tamaño de las células puede variar mucho dependiendo del lugar en que se encuentre. Las estaciones de base se separan entre 1 a 3 Km. en zonas urbanas, aunque pueden llegar a separarse por más de 35 Km en zonas rurales.

En zonas muy densamente pobladas o áreas con muchos obstáculos (como ser edificios altos), las células pueden concentrarse en distancias cada vez menores. Algunas tecnologías, como los PCS (Personal Communication Services), requieren células muy cercanas unas de otras debido a su alta frecuencia y bajo poder en el que operan.

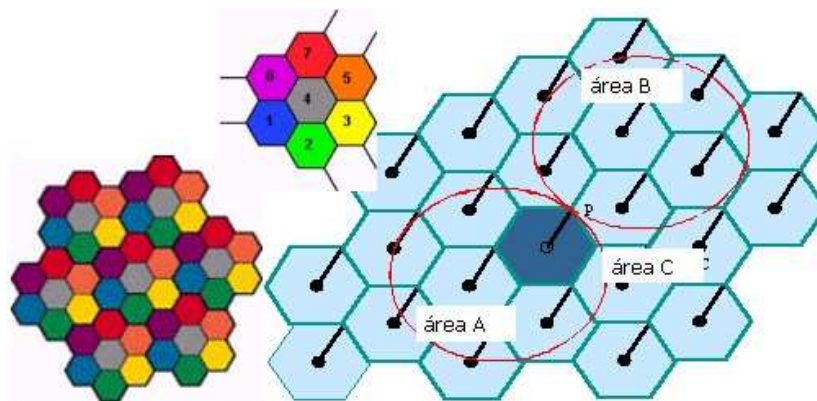


Figura II.6. Reutilización de frecuencias en celdas no adyacentes.

Los edificios pueden, a su vez, interferir con el envío de las señales entre las células que se encuentren más lejanas, por lo que algunos edificios tienen su propia "microcélula."

Los subterráneos son típicos escenarios donde una microcélula se hace necesaria.

Microcélulas pueden ser usadas para incrementar la capacidad general de la red en zonas densamente pobladas como ser los centros capitalinos.

Debido a que los teléfonos celulares y las estaciones de base utilizan transmisores de bajo poder, las mismas frecuencias pueden ser reutilizadas en células no adyacentes, un ejemplo de la reutilización de frecuencias se muestra en la figura II.6. cada frecuencia se representa con un color diferente.

Cada celda en un sistema análogo utiliza un séptimo de los canales de voz disponibles. Eso es, una celda, más las seis celdas que la rodean en un arreglo hexagonal, cada una utilizando un séptimo de los canales disponibles para que cada celda tenga un grupo único de frecuencias y no haya colisiones entre células adyacentes.

Esta configuración puede verse en forma gráfica, en la figura II.6., puede observarse un grupo de células numerado.

De esta forma, en un sistema analógico, en cualquier celda pueden hablar 59 personas en sus teléfonos celulares al mismo tiempo. Con la transmisión digital, el número de canales disponibles aumenta. Por ejemplo el sistema digital TDMA puede acarrear el triple de llamadas en cada celda, alrededor de 168 canales disponibles simultáneamente.

Cada célula tiene una estación base que consta de una torre y un pequeño edificio en donde se tiene el equipo de radio. Cada célula utiliza un séptimo de los 416 canales duales de voz. Dejando entonces a cada célula aproximadamente los 59 canales disponibles nombrados anteriormente.

Si bien los números pueden variar dependiendo de la tecnología usada en el lugar, las cantidades sirven para mostrar cómo funciona esta tecnología; que en caso de tratarse de una generación más moderna, puede de todas formas extrapolarse directamente.

Los teléfonos celulares poseen unos transmisores de bajo poder dentro de ellos. Muchos teléfonos celulares tienen 2 fuerzas de señal: 0.6 Watts y 3 Watts (como comparación, la mayoría de los radios de onda corta transmiten a 5 Watts). La estación base también transmite a bajo poder.

Los transmisores de bajo poder tienen 2 ventajas:

El consumo de energía del teléfono, que normalmente opera con baterías, es relativamente bajo. Esto significa que bajo poder requiere baterías pequeñas, y esto hace posible que existan teléfonos que caben en la mano. A su vez aumenta en forma considerable el tiempo en que se puede usar el teléfono entre carga y carga de la batería.

Las transmisiones de las estaciones base y de los teléfonos no alcanzan una distancia más allá de la célula. Es por esto que en la figura de arriba en cada celda se pueden utilizar las mismas frecuencias sin interferir unas con otras.

Las transmisiones de la base central y de los teléfonos en la misma celda no salen de ésta. Por lo tanto, cada celda puede reutilizar las mismas 59 frecuencias a través de la ciudad.

La tecnología celular requiere un gran número de estaciones base para ciudades de cualquier tamaño. Una ciudad típica grande puede tener cientos de torres emisoras, como la de la figura II.7. Pero debido a que hay tanta gente utilizando teléfonos celulares, los

costos se mantienen bajos para el usuario. Cada portador en cada ciudad tiene una oficina central llamada MTSO. Esta oficina maneja todas las conexiones telefónicas y estaciones base de la región.



Figura II.7. Típica torre de transmisión de telefonía celular

Cuando el usuario desea realizar una llamada, el teléfono celular envía un mensaje a la torre solicitando una conexión a un número de teléfono específico. Si la torre dispone de los suficientes recursos para permitir la comunicación, un dispositivo llamado "switch" conecta la señal del teléfono celular a un canal en la red de telefonía pública. La llamada en este momento toma un canal inalámbrico así como un canal en la red de telefonía pública que se mantendrán abiertos hasta que la llamada se concluya. En la figura II.8. se gráfica lo descrito anteriormente.

Digamos que usted tiene un celular, lo enciende, y alguien trata de llamarle. La MTSO recibe la llamada, y trata de encontrarlo. Desde los primeros sistemas la MTSO lo encontraba activando su teléfono (utilizando uno de los canales de control, ya que su teléfono se encuentra siempre escuchando) en cada célula de la región hasta que su teléfono respondiera. Entonces la estación base y el teléfono decidirán cuál de los 59 canales en su teléfono celular usará. Ahora estará conectado a la estación base y puede empezar a hablar y escuchar.

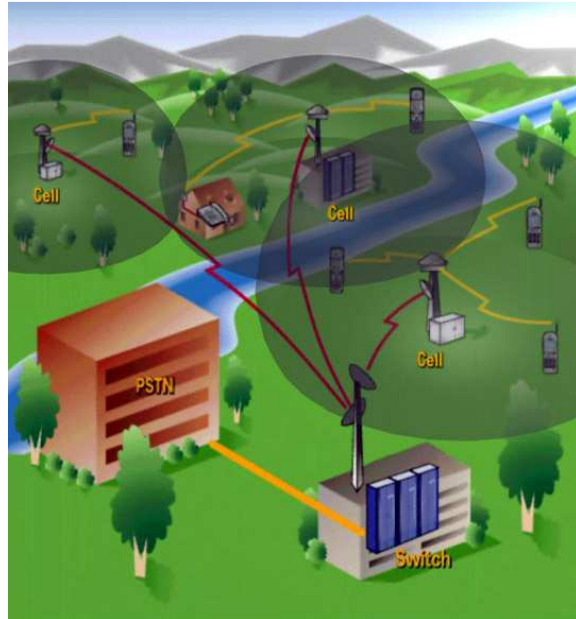


Figura II.8. Funcionamiento de la red celular

A medida que usted se mueva en la célula, la estación base notará que la fuerza de su señal disminuye. Entretanto, la estación base de la célula hacia la que se está moviendo (que está escuchando la señal) será capaz de notar que la señal se hace más fuerte.

Las dos estaciones base se coordinan a sí mismas a través del MTSO, y en algún punto su teléfono obtiene una señal que le indica que cambie de frecuencia. Este cambio hace que su teléfono mude su señal a otra célula.

En sistemas modernos los teléfonos esperan una señal de identificación del sistema (IDS) del canal de control cuando se encienden. El teléfono también transmite una propuesta de registro y la red mantiene unos datos acerca de su ubicación en una base de datos (de esta forma es que la MTSO sabe en qué célula se encuentra si quiere timbrar su teléfono). A medida que se mueve entre células, el teléfono detecta los cambios en la señal, los registra y compara para con los de la nueva célula cuando cambia de canal. Si el teléfono no puede hallar canales para escuchar se sabe que está fuera de rango y muestra un mensaje de "sin servicio".

Éste es, en forma bastante simplificada, el funcionamiento de la telefonía celular; abarcando desde el aspecto teórico en la división de las zonas geográficas en células, hasta el intercambio de ondas electro magnéticas necesario para establecer una sencilla comunicación entre dos teléfonos celulares. Si bien puede enfocarse el tema de manera mucho más técnica, deteniéndose más en aspectos de frecuencia y amplitud de las ondas por ejemplo, preferimos darle un enfoque más general, dando sí algunos datos técnicos específicos que nos parecieron de mayor relevancia para el entendimiento general del tema.

2.4.2. TECNOLOGÍAS UTILIZADAS EN LOS TELÉFONOS CELULARES

TECNOLOGÍAS DE ACCESO CELULAR

Las tecnologías utilizadas actualmente para la transmisión de información en las redes son denominadas de acceso múltiple, debido a que más de un usuario puede utilizar cada una de las celdas de información. Actualmente existen tres diferentes, que difieren en los métodos de acceso a las celdas:

GSM

Es un estándar mundial para teléfonos celulares. Llamado *Global System for Mobile communications* (Sistema Global para las comunicaciones móviles), formalmente conocida como *Group Special Mobile* (GSM, Grupo Especial Móvil). Fue creado por CEPT (organismo internacional que agrupa a las entidades responsables en la Administración Pública de cada país europeo de las políticas y la regulación de las comunicaciones, tanto postales como de telecomunicaciones), y posteriormente

desarrollado por ETSI (*European Telecommunications Standards Institute* – organización de estandarización de la industria de las telecomunicaciones de Europa con proyección mundial) para estandarizar la telefonía celular en Europa, luego adoptado por el resto del mundo. En el año 2001, el 70% de los usuarios de telefonía móvil en el mundo usaban GSM. Es un estándar abierto, no propietario y que se encuentra en desarrollo constante.

GSM emplea una combinación de TDMA y FDMA entre estaciones en un par de canales de radio de frecuencia duplex, con baja lupulización de frecuencia entre canales. Como se explicó anteriormente, TDMA se utiliza para información digital codificada, por lo que GSM es un sistema diseñado para utilizar señales digitales, así como también, canales de voz digitales, lo que permite un moderado nivel de seguridad.

Existen cuatro versiones principales, basadas en la banda: GSM-850, GSM-900, GSM-1800 y GSM-1900, diferenciándose cada una en la frecuencia de las bandas.

En GSM, las conexiones se pueden utilizar tanto a la voz, como a datos, lo que permitió el avance del envío y consumo de datos a través de los celulares. Los casos más comunes son las imágenes que se pueden enviar y recibir, y el uso de aplicaciones a través de los teléfonos móviles, tal es el caso de Internet.

Las implementaciones más veloces de GSM se denominan GPRS y EDGE, también denominadas generaciones intermedias, o 2.5G, que conducen a la tercera generación (3G), o UMTS.

2.4.3. GPRS (General Packet Radio Service)

Básicamente es una comunicación basada en paquetes de datos. En GSM, los intervalos de tiempo son asignados mediante una conexión conmutada, en tanto que en GPRS son asignados mediante un sistema basado en la necesidad a la conexión de paquetes. Es

decir, que si no se envía ningún dato por el usuario, las frecuencias quedan libres para ser utilizadas por otros usuarios. Los teléfonos GPRS por lo general utilizan un puerto bluetooth para la transferencia de datos.

2.4.4. INFLUENCIA EN LA SOCIEDAD

Las comunicaciones móviles contribuyen a la eficiencia de las compañías, tanto en logística, marketing como en las comunicaciones internas mas allá de eso el teléfono móvil ha probado ser un instrumento valioso para la pequeña empresa y sus dueños. Nuevos conceptos de servicios en el sector público han crecido alrededor de la telefonía móvil, por ejemplo, aquellos basados en SMS.

Todo lo que nos rodea, incluyendo el celular, dice mucho sobre cómo somos. Las publicidades gráficas muestran varias características para ayudar al consumidor a poner su propio toque personal. El objetivo será ver los diferentes usos que contienen los celulares y como son capaces de persuadirnos a través de ellos.

El celular es un elemento para comunicarse, pero con el avance de la tecnología nos da una comunicación que va más allá de esto. En las publicidades se hace hincapié en los diferentes usos que brinda el celular, más que en su objetivo principal: "el de comunicarse mediante un llamado telefónico". También apuntan a lo simbólico, en donde todo lo que el sujeto es, lo hace gracias a poseer un celular, "su" celular. Las posibilidades que brindan los teléfonos celulares son infinitas, y ya se puede considerarlo como un objeto de uso personal, ya que el sujeto se identifica con el celular.

El gran avance tecnológico en la telefonía celular, ha permitido un crecimiento, tanto en el diseño de los celulares (su peso, grosor, pantalla color, cantidad de líneas, etc.), como

en la innovación de accesorios disponibles para cada celular en particular. Por ejemplo: manos libres con radio que permite sintonizar el dial que desee el consumidor y a su vez la posibilidad de hablar por teléfono sin tener que interrumpir sus actividades normales.

Las empresas a través de usos y características de los teléfonos crean una nueva necesidad para el usuario. Algunas de ellas son:

- cámara de video fotográfica
- juegos
- mayor velocidad de conexión a Internet y descargas de la web; la persona puede enviar imágenes, mensajes o e-mails y también bajar rings tons, mp3, chat.
- resolución de pantalla
- GSM
- sonido polifónico
- memoria
- agenda
- alarma

2.5.JAVA

Java es una plataforma con lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el

bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

Java es el primer lenguaje que tiene la virtud de ser compilado e interpretado de forma simultánea. Cuando un programador realiza una aplicación o un applet en Java y lo compila, en realidad, el compilador no trabaja como un compilador de un lenguaje al uso. El compilador Java únicamente genera el denominado ByteCode.

Este código es un código intermedio entre el lenguaje máquina del procesador y Java. Evidentemente este código no es ejecutable por sí mismo en ninguna plataforma hardware, pues no se corresponde con el lenguaje de ninguno de los procesadores que actualmente se conocen (habrá que esperar a ver qué ocurre con los procesadores Java). Por lo tanto, para ejecutar una aplicación Java es necesario disponer de un mecanismo

que permita ejecutar el ByteCode. Este mecanismo es la denominada Máquina Virtual Java. En cada plataforma (Unix, Linux, Windows 95/NT, Macintosh, etc.) existe una máquina virtual específica. Así que cuando el ByteCode llega a la máquina virtual, ésta lo interpreta pasándolo a código máquina del procesador donde se esté trabajando, y ejecutando las instrucciones en lenguaje máquina que se deriven de la aplicación Java. De este modo, cuando el mismo ByteCode llega a diferentes plataformas, éste se ejecutará de forma correcta, pues en cada una de esas plataformas existirá la máquina virtual adecuada. Con este mecanismo se consigue la famosa multiplataforma de Java, que con sólo codificar una vez, podemos ejecutar en varias plataformas.

2.5.1. FILOSOFÍA

La plataforma Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine o OSGi respectivamente.

INDEPENDENCIA DE LA PLATAFORMA

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode), instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se

ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una recompilación dinámica en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias. El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el

entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

2.5.2. APIS

Sun define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus APIs (Application Program Interface) de forma que pertenezcan a cada una de las plataformas:

- Java ME (Java Platform, Micro Edition) o JME orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.
- Java SE (Java Platform, Standard Edition) o J2SE para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
- Java EE (Java Platform, Enterprise Edition) o J2EE orientada a entornos distribuidos empresariales o de Internet.

Las clases en las APIs de Java se organizan en grupos disjuntos llamados paquetes. Cada paquete contiene un conjunto de interfaces, clases y excepciones relacionadas. La información sobre los paquetes que ofrece cada plataforma puede encontrarse en la documentación de ésta.

El conjunto de las APIs es controlado por Sun Microsystems junto con otras entidades o personas a través del programa JCP (Java Community Process). Las compañías o individuos participantes del JCP pueden influir de forma activa en el diseño y desarrollo de las APIs, algo que ha sido motivo de controversia.

En 2004, IBM y BEA apoyaron públicamente la idea de crear una implementación de código abierto (open source) de Java, algo a lo que Sun se ha negado.

2.5.3. JAVA MICRO EDITION

La plataforma Java Micro Edition, o anteriormente Java 2 Micro Edition (J2ME), es una especificación de un subconjunto de la plataforma Java orientada a proveer una colección certificada de APIs de desarrollo de software para dispositivos con recursos restringidos. Está orientado a productos de consumo como PDAs, teléfonos móviles o electrodomésticos.

Java ME se ha convertido en una buena opción para crear juegos en teléfonos móviles debido a que se puede emular en un PC durante la fase de desarrollo y luego subirlos fácilmente al teléfono. Al utilizar tecnologías Java el desarrollo de aplicaciones o videojuegos con estas APIs resulta bastante económico de portar a otros dispositivos.

Las necesidades de los usuarios de telefonía móvil han cambiado mucho en estos últimos años y cada vez demandan más servicios y prestaciones por parte tanto de los terminales como de las compañías. Además el uso de esta tecnología depende del asentamiento en el mercado de otras, como GPRS, íntimamente asociada a JME y que no ha estado a nuestro alcance hasta hace poco. JME es la tecnología del futuro para la industria de los dispositivos móviles. Actualmente las compañías telefónicas y los fabricantes de móviles están implantando los protocolos y dispositivos necesarios para soportarla.

En la actualidad no es realista ver Java como un simple lenguaje de programación, si no como un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje Java es compilado a código intermedio interpretado por una Java Virtual Machine (JVM), por lo que el código ya compilado es independiente de la plataforma.

- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes `java.lang` y `java.io`.

Un claro ejemplo de éste último punto es que JME contiene una mínima parte de las APIs de Java. Esto es debido a que la edición estándar de APIs de Java ocupa 20 Mb, y los dispositivos pequeños disponen de una cantidad de memoria mucho más reducida. En concreto, JME usa 37 clases de la plataforma JSE provenientes de los paquetes `java.lang`, `java.io`, `java.util`. Esta parte de la API que se mantiene fija forma parte de lo que se denomina “configuración”. Otras diferencias con la plataforma JSE vienen dadas por el uso de una máquina virtual distinta de la clásica JVM denominada KVM. Esta KVM tiene unas restricciones que hacen que no posea todas las capacidades incluidas en la JVM.

Como vemos, JME representa una versión simplificada de JSE. Sun separó estas dos versiones ya que JME estaba pensada para dispositivos con limitaciones de proceso y capacidad gráfica. También separó JSE de JEE porque este último exigía unas características muy pesadas o especializadas de E/S, trabajo en red, etc. Por tanto, separó ambos productos por razones de eficiencia. Hoy, JEE es un superconjunto de JSE pues contiene toda la funcionalidad de éste y más características, así como JME es un subconjunto de JSE (excepto por el paquete `javax.microedition`) ya que, como se ha mencionado, contiene varias limitaciones con respecto a JSE, esta relación entre todas las APIs de Java la observamos en la figura II.9.

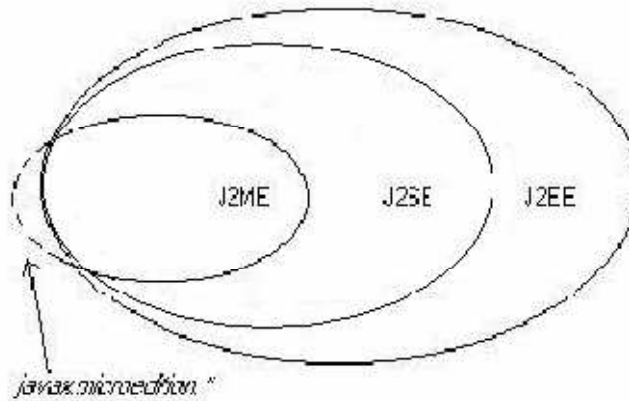


Figura II.9. Relación entre las APIs de la plataforma Java.

NOCIONES BÁSICAS DE JME

Ya hemos visto qué es Java Micro Edition y la hemos enmarcado dentro de la plataforma Java. En este apartado vamos a ver cuáles son los componentes que forman parte de esta tecnología.

- Por un lado tenemos una serie de máquinas virtuales Java con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.
- Configuraciones, que son un conjunto de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de características específicas.

Existen 2 configuraciones definidas en JME:

Connected Limited Device Configuration (CLDC) enfocada a dispositivos con restricciones de procesamiento y memoria, y **Connected Device Configuration (CDC)** enfocada a dispositivos con más recursos.

Perfiles, que son unas bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.

Un entorno de ejecución determinado de JME se compone entonces de una selección de:

- a) Máquina virtual.

- b) Configuración.
- c) Perfil.
- d) Paquetes Opcionales.

MÁQUINAS VIRTUALES JME

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (bytecode) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente. Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales. JME define varias JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos que, en algunos casos, suprimen algunas características con el fin de obtener una implementación menos exigente.

Ya hemos visto que existen 2 configuraciones CLDC y CDC, cada una con unas características propias. Como consecuencia, cada una requiere su propia máquina virtual. La VM (Virtual Machine) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM. Veremos a continuación las características principales de cada una de ellas:

KVM

Se corresponde con la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria, entre

40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM está escrita en lenguaje C, aproximadamente unas 24000 líneas de código, y fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- Alta portabilidad.
- Modulable.
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada. Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica Java Virtual Machine (JVM):

El verificador de clases estándar de Java es demasiado grande para la KVM. De hecho es más grande que la propia KVM y el consumo de memoria es excesivo, más de 100Kb para las aplicaciones típicas. Este verificador de clases es el encargado de rechazar las clases no válidas en tiempo de ejecución. Este mecanismo verifica los bytecodes de las clases Java realizando las siguientes comprobaciones:

- Ver que el código no sobrepase los límites de la pila de la VM.
- Comprobar que no se utilizan las variables locales antes de ser inicializadas.
- Comprobar que se respetan los campos, métodos y los modificadores de control de acceso a clases.

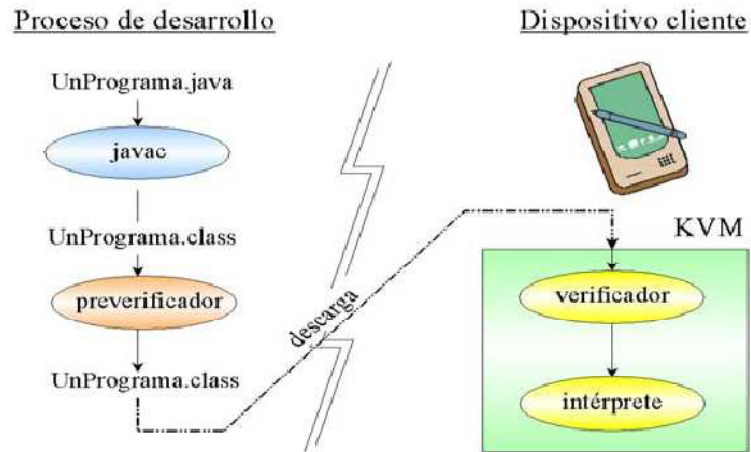


Figura II.10. Algoritmo de verificación de clases

Por esta razón los dispositivos que usen la configuración CLDC y KVM introducen un algoritmo de verificación de clases en dos pasos. Este proceso puede apreciarse gráficamente en la figura II.10.

La KVM puede ser compilada y probada en 3 plataformas distintas:

1. Solaris Operating Environment.
2. Windows
3. PalmOs

CVM

La CVM (Compact Virtual Machine) ha sido tomada como Máquina Virtual Java de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de JSE. Está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2Mb o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

1. Sistema de memoria avanzado.

2. Tiempo de espera bajo para el recolector de basura.
3. Separación completa de la VM del sistema de memoria.
4. Recolector de basura modularizado.
5. Portabilidad.
6. Rápida sincronización.
7. Ejecución de las clases Java fuera de la memoria de sólo lectura (ROM).
8. Soporte nativo de hilos.
9. Baja ocupación en memoria de las clases.
10. Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real.
11. Conversión de hilos Java a hilos nativos.
12. Soporte para todas las características de Java2 v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de Java (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI).

CONFIGURACIONES

Ya hemos mencionado algo anteriormente relacionado con las configuraciones.

Para tenerlo bien claro diremos que una configuración es el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un grupo de dispositivos. Éstas APIs describen las características básicas, comunes a todos los dispositivos:

- Características soportadas del lenguaje de programación Java.
- Características soportadas por la Máquina Virtual Java.
- Bibliotecas básicas de Java y APIs soportadas.

Como ya hemos visto con anterioridad, existen dos configuraciones en JME:

CLDC, orientada a dispositivos con limitaciones computacionales y de memoria y CDC, orientada a dispositivos con no tantas limitaciones. Ahora veremos un poco más en profundidad cada una de estas configuraciones.

Configuración de dispositivos con conexión, CDC

La CDC está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con internet, algunos electrodomésticos y sistemas de navegación en automóviles. CDC usa una Máquina Virtual Java similar en sus características a una de JSE, pero con limitaciones en el apartado gráfico y de memoria del dispositivo. Ésta Máquina Virtual es la que hemos visto como CVM (Compact Virtual Machine). La CDC está enfocada a dispositivos con las siguientes capacidades:

Procesador de 32 bits.

Disponer de 2 Mb o más de memoria total, incluyendo memoria RAM y ROM.

Poseer la funcionalidad completa de la Máquina Virtual Java.

Conectividad a algún tipo de red.

La CDC está basada en JSE v1.3 e incluye varios paquetes Java de la edición estándar. Las peculiaridades de la CDC están contenidas principalmente en el paquete `javax.microedition.io`, que incluye soporte para comunicaciones http y basadas en datagramas. La Tabla II.2 nos muestra las librerías incluidas en la CDC.

| Nombre de Paquete CDC | Descripción |
|-----------------------|--|
| java.io | Clases e interfaces estándar de E/S. |
| java.lang | Clases básicas del lenguaje. |
| Java.lang.ref | Clases de referencia. |
| Java.lang.reflect | Clases e interfaces de reflexión. |
| Java.math | Paquete de matemáticas. |
| Java.net | Clases e interfaces de red. |
| Java.security | Clases e interfaces de seguridad |
| java.security.cert | Clases de certificados de seguridad. |
| Java.text | Paquete de texto. |
| Java.util | Clases de utilidades estándar. |
| Java.util.jar | Clases y utilidades para archivos JAR. |
| Java.util.zip | Clases y utilidades para archivos ZIP y comprimidos. |
| Javax.microedition.io | Clases e interfaces para conexión genérica CDC. |

Tabla II.2. Librerías de CDC

Configuración de dispositivos limitados con conexión, CLDC

La CLDC está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de estos dispositivos son: teléfonos móviles, buscapersonas (pagers), PDAs, organizadores personales, etc.

Ya hemos dicho que CLDC está orientado a dispositivos con ciertas restricciones.

Algunas de estas restricciones vienen dadas por el uso de la KVM, necesaria al trabajar con la CLDC debido a su pequeño tamaño. Los dispositivos que usan CLDC deben cumplir los siguientes requisitos:

- Disponer entre 160 Kb y 512 Kb de memoria total disponible. Como mínimo se debe disponer de 128 Kb de memoria no volátil para la Máquina Virtual
- Java y las bibliotecas CLDC, y 32 Kb de memoria volátil para la Máquina Virtual en tiempo de ejecución.
- Procesador de 16 o 32 bits con al menos 25 Mhz de velocidad.

- Ofrecer bajo consumo, debido a que estos dispositivos trabajan con suministro de energía limitado, normalmente baterías.
- Tener conexión a algún tipo de red, normalmente sin cable, con conexión intermitente y ancho de banda limitado (unos 9600 bps).

La CLDC aporta las siguientes funcionalidades a los dispositivos:

Un subconjunto del lenguaje Java y todas las restricciones de su Máquina Virtual (KVM).

Un subconjunto de las bibliotecas Java del núcleo.

Soporte para E/S básica.

Soporte para acceso a redes.

Seguridad.

La Tabla II.3 nos muestra las librerías incluidas en la CLDC.

| Nombre de paquete CLDC | Descripción |
|------------------------|--|
| java.io | Clases y paquetes estándar de E/S. Subconjunto de J2SE. |
| java.lang | Clases e interfaces de la Máquina Virtual. Subconj. de J2SE. |
| java.util | Clases, interfaces y utilidades estándar. Subconj. de J2SE. |
| javax.microedition.io | Clases e interfaces de conexión genérica CLDC |

Tabla II.3. Librerías de configuración CLDC.

Un aspecto muy a tener en cuenta es la seguridad en CLDC. Esta configuración posee un modelo de seguridad sandbox al igual que ocurre con los applets. En cualquier caso, una determinada Configuración no se encarga del mantenimiento del ciclo de vida de la

aplicación, interfaces de usuario o manejo de eventos, sino que estas responsabilidades caen en manos de los perfiles.

PERFILES

Acabamos de decir que el perfil es el que define las APIs que controlan el ciclo de vida de la aplicación, interfaz de usuario, etc. Más concretamente, un perfil es un conjunto de APIs orientado a un ámbito de aplicación determinado. Los perfiles identifican un grupo de dispositivos por la funcionalidad que proporcionan (electrodomésticos, teléfonos móviles, etc.) y el tipo de aplicaciones que se ejecutarán en ellos. Las librerías de la interfaz gráfica son un componente muy importante en la definición de un perfil. Aquí nos podemos encontrar grandes diferencias entre interfaces, desde el menú textual de los teléfonos móviles hasta los táctiles de los PDAs.

El perfil establece unas APIs que definen las características de un dispositivo, mientras que la configuración hace lo propio con una familia de ellos. Esto hace que a la hora de construir una aplicación se cuente tanto con las APIs del perfil como de la configuración. Tenemos que tener en cuenta que un perfil siempre se construye sobre una configuración determinada. De este modo, podemos pensar en un perfil como un conjunto de APIs que dotan a una configuración de funcionalidad específica.

Ya hemos visto los conceptos necesarios para entender cómo es un entorno de ejecución en Java Micro Edition. Este entorno de ejecución se estructura en capas, una construida sobre la otra. Anteriormente vimos que para una configuración determinada se usaba una Máquina Virtual Java específica. Teníamos que con la configuración CDC usábamos la CVM y que con la configuración CLDC usábamos la KVM. Con los perfiles ocurre lo

mismo. Existen unos perfiles que construiremos sobre la configuración CDC y otros que construiremos sobre la CLDC.

Para la configuración CDC tenemos los siguientes perfiles:

- Foundation Profile.
- Personal Profile.
- RMI Profile.

Para la configuración CLDC tenemos los siguientes:

- PDA Profile.
- Mobile Information Device Profile (MIDP).

Todo esto lo podemos observar en la figura II.11.

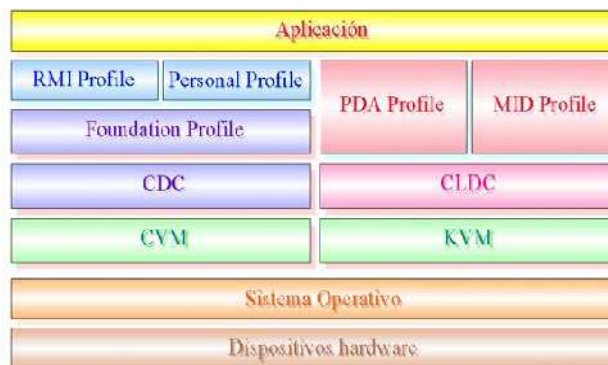


Figura II.11. Arquitectura del entorno de ejecución de JME.

Un perfil puede ser construido sobre cualquier otro. Sin embargo, una plataforma JME sólo puede contener una configuración. A continuación vamos a ver con detenimiento cada uno de estos perfiles:

Foundation Profile: Este perfil define una serie de APIs sobre la CDC orientadas a dispositivos que carecen de interfaz gráfica como, por ejemplo, decodificadores de

televisión digital. Este perfil incluye gran parte de los paquetes de la JSE, pero excluye totalmente los paquetes “java.awt” Abstract Windows Toolkit (AWT) y “java.swing” que conforman la interfaz gráfica de usuario (GUI) de JSE. Si una aplicación requiriera una GUI, entonces sería necesario un perfil adicional. Los paquetes que forman parte del Foundation Profile se muestran en la Tabla II.4.

| Paq. del Foundation Profile | Descripción |
|-----------------------------|---|
| java.lang | Soporte del lenguaje Java |
| java.util | Añade soporte completo para zip y otras funcionalidades (java.util.Timer) |
| java.net | Incluye sockets TCP/IP y conexiones HTTP |
| java.io | Clases Reader y Writer de J2SE |
| java.text | Incluye soporte para internacionalización |
| java.security | Incluye códigos y certificados |

Tabla II.4 Librerías del Foundation Profile.

Personal Profile: El Personal Profile es un subconjunto de la plataforma JSE v1.3, y proporciona un entorno con un completo soporte gráfico AWT.

El objetivo es el de dotar a la configuración CDC de una interfaz gráfica completa, con capacidades web y soporte de applets Java. Este perfil requiere una implementación del Foundation Profile. La Tabla II.5 nos muestra los paquetes que conforman el Personal Profile v1.0.

RMI Profile: Este perfil requiere una implementación del Foundation Profile se construye encima de él. El perfil RMI soporta un subconjunto de las APIs JSE v1.3 RMI. Algunas características de estas APIs se han eliminado del perfil RMI debido a las limitaciones de cómputo y memoria de los dispositivos.

PDA Profile: El PDA Profile está construido sobre CLDC. Pretende abarcar PDAs de gama baja, tipo Palm, con una pantalla y algún tipo de puntero (ratón o lápiz) y una resolución de al menos 20000 pixels (al menos 200x100 pixels) con un factor 2:1. No es posible dar mucha más información porque en este momento este perfil se encuentra en fase de definición.

| Paq. del Personal Profile | Descripción |
|---------------------------|---|
| java.applet | Clases necesitadas para crear applets o que son usadas por ellos |
| java.awt | Clases para crear GUIs con AWT |
| java.awt.datatransfer | Clases e interfaces para transmitir datos entre aplicaciones |
| java.awt.event | Clases e interfaces para manejar eventos AWT |
| java.awt.font | Clases e interfaces para la manipulación de fuentes |
| java.awt.im | Clases e interfaces para definir métodos editores de entrada |
| java.awt.im.spi | Interfaces que añaden el desarrollo de métodos editores de entrada para cualquier entorno de ejecución Java |
| java.awt.image | Clases para crear y modificar imágenes |
| java.beans | Clases que soportan JavaBeans |
| javax.microedition.xlet | Interfaces que usa el Personal Profile para la comunicación. |

Tabla II.5 Librerías del Personal Profile

Mobile Information Device Profile (MIDP): Este perfil está construido sobre la configuración CLDC. Al igual que CLDC fue la primera configuración definida para J2ME, MIDP fue el primer perfil definido para esta plataforma.

Este perfil está orientado para dispositivos con las siguientes características:

- Reducida capacidad computacional y de memoria.
- Conectividad limitada (en torno a 9600 bps).
- Capacidad gráfica muy reducida (mínimo un display de 96x54 pixels monocromo).
- Entrada de datos alfanumérica reducida.

- 128 Kb de memoria no volátil para componentes MIDP.
- 8 Kb de memoria no volátil para datos persistentes de aplicaciones.
- 32 Kb de memoria volátil en tiempo de ejecución para la pila Java.

Los tipos de dispositivos que se adaptan a estas características son: teléfonos móviles, buscapersonas (pagers) o PDAs de gama baja con conectividad.

El perfil MIDP establece las capacidades del dispositivo, por lo tanto, especifica las APIs relacionadas con:

- La aplicación (semántica y control de la aplicación MIDP).
- Interfaz de usuario.
- Almacenamiento persistente.
- Trabajo en red.
- Temporizadores.

En la Tabla II.6 podemos ver cuáles son los paquetes que están incluidos en el perfil MIDP.

| Paquetes del MIDP | Descripción |
|---------------------------|---|
| javax.microedition.lcdui | Clases e interfaces para GUIs |
| javax.microedition.rms | Record Management Storage. Soporte para el almacenamiento persistente del dispositivo |
| javax.microedition.midlet | Clases de definición de la aplicación |
| javax.microedition.io | Clases e interfaces de conexión genérica |
| java.io | Clases e interfaces de E/S básica |
| java.lang | Clases e interfaces de la Máquina Virtual |
| java.util estándar | Clases e interfaces de utilidades |

Tabla II.6 Librerías del perfil MIDP.

Las aplicaciones que realizamos utilizando MIDP reciben el nombre de MIDlets (por simpatía con Applets). Decimos así que un MIDlet es una aplicación Java realizada con

el perfil MIDP sobre la configuración CLDC. En los temas siguientes nos centraremos en la creación de estos MIDlets ya que es un punto de referencia para cualquier programador de JME. Además, desde un punto de vista práctico MIDP es el único perfil actualmente disponible.

2.5.4. WIRELESS MESSAGING API

Originalmente presentado en la Java Community Process como JSR 120, el WMA 1.1 ha sido extendido y actualizado como WMA 2.0 en la JSR 205.

El Wireless Messaging (WMA) provee una interface común que puede ser utilizado para habilitar una aplicación basada en Mobile Information Device Profile (MIDP) para enviar y recibir mensajes cortos de texto y mensajes binarios, así como mensajes multimedia. Estos mensajes típicamente son partes de un sistema de almacenamiento y envío de mensajes, tales como Short Messaging Service (SMS) y el Multimedia Messaging Service (MMS)

WMA apunta a teléfonos celulares y otros dispositivos que pueden enviar y recibir mensajes en forma inalámbrica. Es un API genérico para envío no solamente de mensajes binarios o de texto sino mensajes multipartes, generalmente usado para transmisión de mensajes multimedia.

La manera que los mensajes son enviados depende de la base de transporte, o portadora, tales como GSM SMS, GSM CBS, CMDA SMS, o MMS. El formato del mensaje y el transporte son definidos por el estándar respectivo, sin embargo la mayor parte del WMA hace tales detalles transparente a la aplicación. Es importante notar que aunque el transporte de SMS y MMS son actualmente administrados de forma diferente en la red, y que el MMS no es solo un medio para transmitir grandes paquetes de SMS. WMA no

establece límites para el tamaño del mensaje u otras restricciones, sin embargo la capa de transporte lo hace como se podrá notar más adelante.

WMA está basado en el Generis Conecction Framework (GFC). Es definido como un paquete opcional del JME; es decir contiene APIs especializados que pueden ser agregados a una pila de software basado en una configuración estándar. El común denominador es el Connected Limited Device Configuration (CLDC). Debido a que el Connected Device Configuration (CDC) es un superconjunto del (CLDC). WMA puede ser incluido en ambas pilas (stacks) CDLC y CDC.

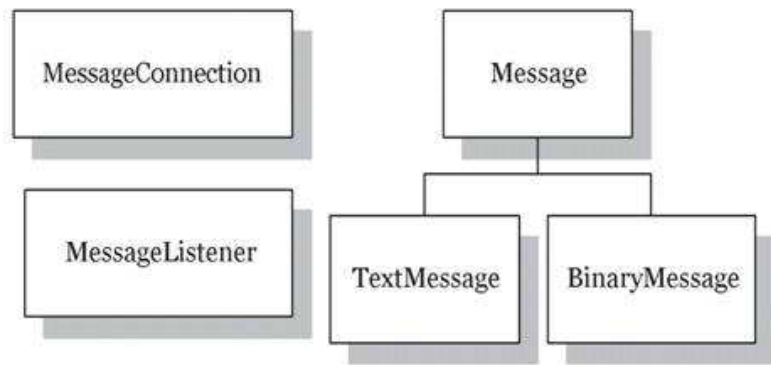


Figura II.12. Componentes del Wíreless Messaging API

Todos los componentes del WMA están contenidos in un solo paquete, javax.wireless.messaging, el cual define todas las interfaces para envío y recepción inalámbrica de mensajes de texto y binario, se observa este paquete en la figura II.12. En la tabla II.7. se describe el contenido de este paquete.

| Interface | Descripción | Métodos |
|-------------------|---|--|
| Message | Base Message interface, del cual subinterfaces (tales como TextMessage y BinaryMessage) son derivados | getAddress(), getTimestamp(), setAddress() |
| BinaryMessage | Subinterface del Message que provee métodos para activar y tomar una carga binaria | getPayloadData(), setPayloadData() |
| TextMessage | Subinterface de Message para acivar y tomar cargas de texto | getPayloadText(), setPayloadText() |
| MessageConnection | Subinterface del GCF Connection, el cual provee una fábrica de Mensajes, y métodos para enviar y recibir Mensajes | newMessage(), receive(), send(), setMessageListener(), numberOfSegments() |
| MessageListener | Define la interface del escuchador para implementar notificación asíncrona de objetos Message | notifyIncomingMessage() |

Tabla II.7. Contenido del paquete javax.wireless.messaging

Seguidamente una breve descripción de cada una de las interfaces

La Interface Message

La interface javax.wireless.messaging.Message es la base de todos los tipos de mensajes que se pueden manejar utilizando el WMA, un mensaje es lo que es enviado y recibido, producido y consumido.

En algunos aspectos, un mensaje es similar a un datagrama: tiene direcciones de origen y destino, una carga, y una manera de enviar y bloquear. El WMA provee funcionalidad adicional, tales como soporte para mensajes binarios y de texto y una interface “escuchadora” para recibir mensajes asincrónicamente.

El WMA define dos interfaces BinaryMessage y TextMessage, y la especificación es extensible para habilitar el soporte de tipos adicionales de mensajes.

La Interface BinaryMessage

La subinterface BinaryMessage representa un mensaje con carga binaria, y declara métodos para activar y tomar mensajes binarios. Métodos generales para activar y tomar la dirección de un mensaje y tomar su tiempo de grabación.

La Interface TextMessage

La subinterface TextMessage representa un mensaje con una carga de texto, tal como el mensaje de texto basado en SMS. Provee métodos para activar y tomar cargas de texto (instancias o String). Antes que el mensaje de texto sea enviado o recibido, la implementación por debajo es responsable de la codificación y decodificación apropiada, del String o del formato apropiado, por ejemplo el GSM 7-bit o UCS-2. Métodos generales para activar y capturar la dirección de un mensaje y tomar su tiempo de grabación son inherentes a Message.

La Interface MessageConnection

La Interface MessageConnection es una subinterface del Generic Connection Framework javax.microedition.io.Connection. En la parte superior de la figura II.13. se puede observar el CFG, y en la parte inferior como la interface MessageConnection se relaciona con el resto del WMA.

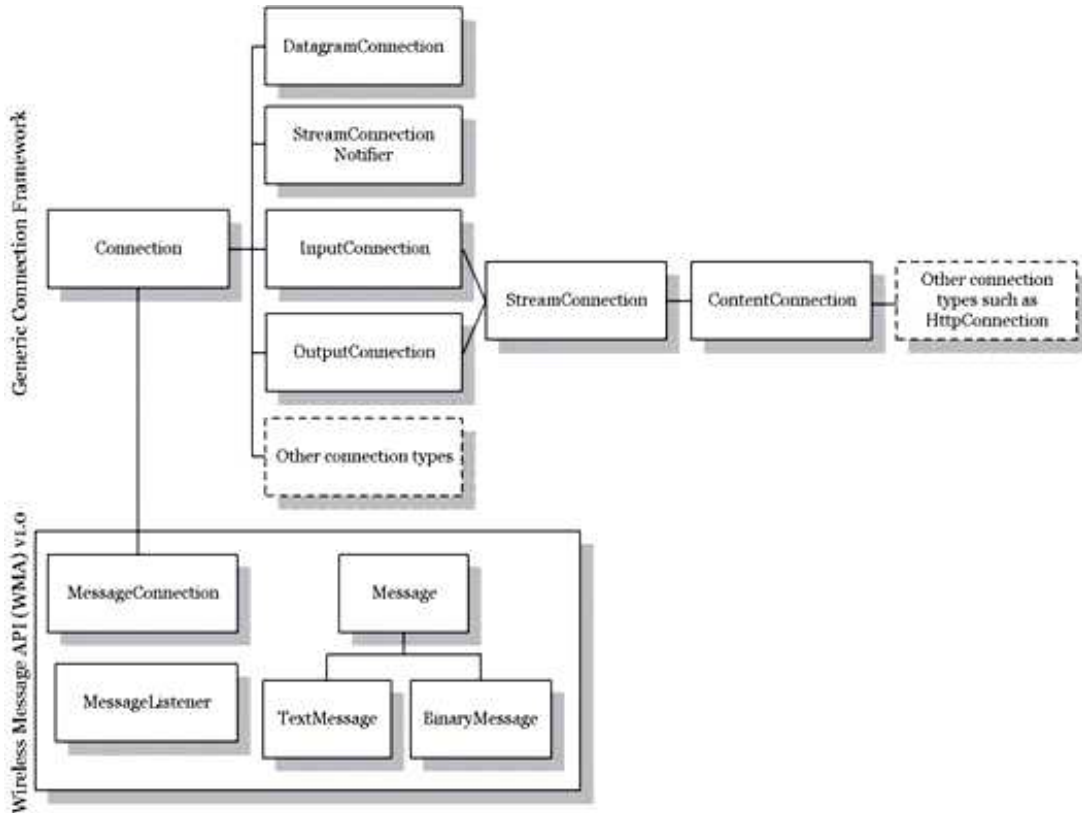


Figura II.13. El MessageConnection y su relación con el CFG

La Interface MessageListener

El MessageListener implementa el patrón de diseño listener para recibir objetos mensajes asincrónicamente; esto es, sin bloqueo mientras se aguardan mensajes. Esta interface define un método simple: notifyIncomingMessage() es invocado por la plataforma cada vez que un mensaje es recibido. Para registrar los mensajes, utiliza el método MessageConnection.setListener(). Debido a la implementación de algunas plataformas, pueden ser tratadas en un solo thread, la cantidad de procesamiento con el notifyIncomingMessage() debería ser mantenidos al mínimo.

2.5.5. THREAD O HILOS DE EJECUCIÓN

Un hilo de ejecución, es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente). Los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

Un hilo es básicamente una tarea que puede ser ejecutada en paralelo con otra tarea.

CAPÍTULO III

3. DISEÑO DEL SISTEMA

3.1. PLANTEAMIENTO

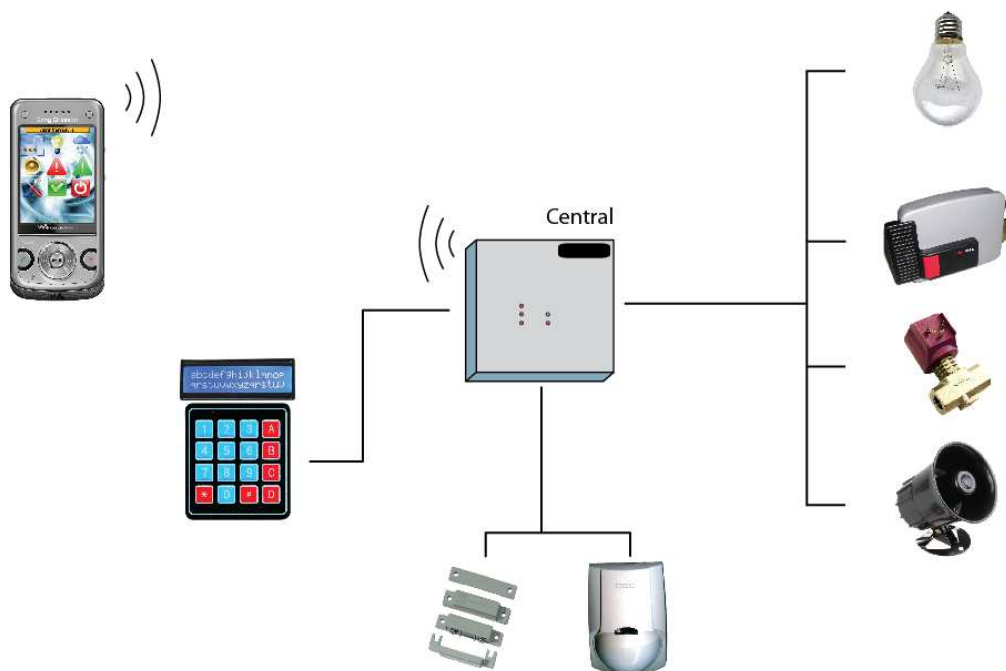


Figura III.14. Esquema detallado de dispositivos a conectar al sistema

Con la implementación de este sistema lo que se busca es brindar una herramienta práctica que fusione el control de los distintos dispositivos de un hogar con algunas herramientas de seguridad, a través de un dispositivo accesible y de uso común como es un teléfono celular, permitiendo el manejo a distancia de estos dispositivos.

La figura III.15 se muestra el esquema detallado de los dispositivos que se busca que el sistema controle, por un lado, están los encargados de emitir las ordenes y controlar el funcionamiento de los dispositivos, con el envío de instrucciones por sus respectivos medios, remotamente en el caso del teléfono celular y un teclado y pantalla conectado directamente a la central.

Todas la instrucciones ejecutadas por el usuario ya sea mediante el dispositivo móvil (teléfono celular) y por otra parte la pantalla y teclado del sistema llegan a la central donde son analizadas y en caso de ser correctas, éste notificará a los usuarios de los problemas suscitados, caso contrario ordenará ejecutar la acción peticionada. Todas la instrucciones parten de la central, esta enciende o apaga los diferentes dispositivos que van a funcionar, dependiendo de la utilidad que se le va a dar al dispositivo, estos son:

- Cerradura Eléctrica
- Iluminación

Estos mencionados anteriormente son considerados dispositivos de un hogar, mientras que en lo respecta a seguridad son:

- Sirena
- Sensores magnéticos
- Sensor de movimiento

Todos estos, con excepción de los sensores, son aparatos que necesitan de flujo eléctrico para activarse; mientras tanto los sensores se encargan de recopilación de datos, es decir están a cargo de la lectura de estados, o los llamados niveles lógicos, en este caso o cero “1” o ”0” (activado o desactivado), permitiendo realizar las acciones respectivas con esas lecturas, en este caso la activación de la sirena y la notificación mediante mensaje de texto a un teléfono celular previamente registrado.

El manejo de estos dispositivos permite al usuario mantener el control y el conocimiento de lo que sucede en su domicilio tanto de la seguridad como de los dispositivos que puedan ser controlados remotamente, que cada vez son mayores en número.

3.2.REQUERIMIENTOS TECNOLÓGICOS

Para determinar los requerimientos del sistema detallamos a continuación las necesidades tecnológicas del mismo:

3.2.1. HARDWARE

- Micontrolador Pic 16F628A
- Micronrolador Pic 16F877A
- LCD 2x16
- Teclado Matricial 4x4
- Detector Digital PIR
- Contactos Magnéticos
- Relés
- Sirena Digital
- Modem GSM (Nokia 3220)

- Cable de Comunicación con el modem DKU-5
- Cristal a 4 MHz
- Bus de datos direccional 74LS245
- Oscilador de Períodos Altos CD4060

3.2.2. SOFTWARE

- Sistema Operativo Windows XP
- NetBeans IDE 6.5.1
- MicroCode Studio 2.3.0.0
- WinPic800 3.55g
- Proteus 7.2
- Microsoft Office 2007

3.3.DISEÑO DE HARDWARE

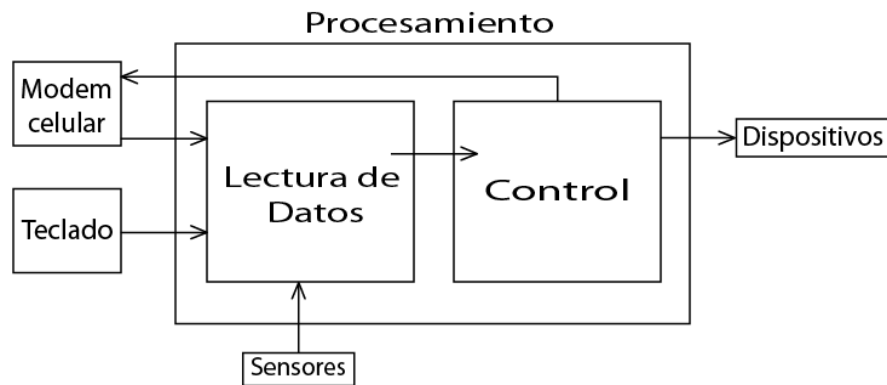


Figura III.15. Esquema general del sistema.

A breves rasgos el flujo de información de los diferentes dispositivos se observa en la figura III.15. y se detalla a continuación:

El sistema estaría controlado por una unidad remota que en este caso es un teléfono celular, con un menú sencillo y funcional programado en java ME. Y también un mando central a cargo de un teclado y una pantalla, que se encuentra en el lugar mismo donde se ejecuta el sistema, para el manejo local del sistema, por ende a los dispositivos. Estos son los encargados de ordenar y poner en funcionamiento los dispositivos según sean las instrucciones.

A mas de que el usuario ordena hacer un tipo de acción según la conveniencia de éste (mando central mando a distancia), por otro lado las lecturas de datos por parte de los sensores son también datos u órdenes que el mismo sistema debe conocer para realizar acciones pertinentes predeterminadas. Todo lo respectivo con el ingreso o lectura de datos, tendrán que procesarse para compararse y dar acción a instrucciones o notificados de eventos o problemas a la persona que maneja el sistema o usuario. Caso contrario simplemente se realizan las acciones respectivas. Estas notificaciones se la mostrarán en la pantalla en el caso del mando central o por medio de mensajes en el otro caso del teléfono celular o mando móvil.

La figura III.16 muestra el diagrama de flujo básico del funcionamiento del sistema, tanto en lo que refiere al control de los dispositivos como del sistema de seguridad, ambos están incluidos en la central, pero presentan un funcionamiento independiente, sin embargo ambos pueden ser controlados como ya se describió anteriormente por cualquier de los dos dispositivos de control.

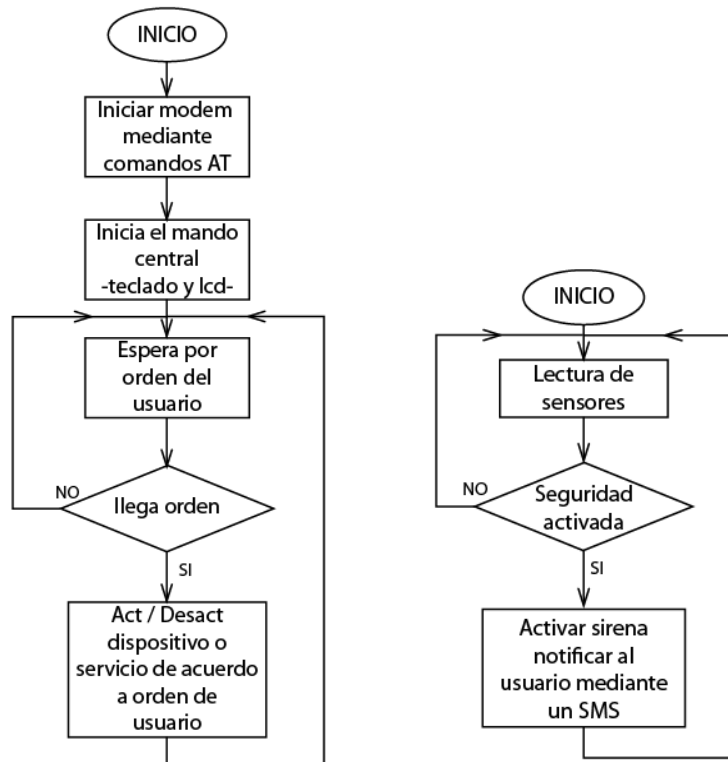


Figura III.16. Diagramas de Flujo del Sistema

3.3.1. SISTEMA DE ADQUISICIÓN DE DATOS

El ingreso de datos al sistema para ejecutar las instrucciones se lo realiza tomando en cuenta los dos dispositivos que el usuario manipula: teléfono celular y el mando central formado por un teclado y una pantalla.

Teléfono celular:

En primer lugar tenemos el teléfono celular (Nokia 3220) que será considerado como nuestro modem, un mando o receptor a distancia, por la alta cobertura que brinda la empresa que dota el servicio de telefonía móvil, ya que por medio del servicio de mensajería la comunicación o transmisión de datos se realiza. El aparato emisor de datos puede ser cualquier tipo de teléfono celular que permita la ejecución de aplicaciones

programadas en JME y contenga el permiso de transferencia de SMS. Mediante una aplicación que corre en el dispositivo que usa programación java ME, por medio de comandos AT, conviértelos pedidos del usuario en este dispositivo de un menú grafico, en instrucciones listas a transmitirse y a ser leídas y comprendidas por el sistema; los datos obtenidos se transmiten y llegan a un receptor denotado anteriormente y estos a su vez ingresan al sistema mediante un cable que es propio de la marca del teléfono en este caso el (DKU-5).

De este cable de datos (DKU-5), utilizamos solo los pines de transmisión (pin 7), recepción (pin 6) y tierra de datos (pin 8).

La comunicación con el celular o modem GSM se lo realiza mediante el cable de datos CA-42, mediante comunicación serial asíncrona a 9600 baudios, con 8 bits de datos, 1 bit de parada, y sin bit de paridad. La descripción de los pines del bus de datos del modem celular se detalla en la tabla III.1. y gráficamente se observa en la figura III.17.

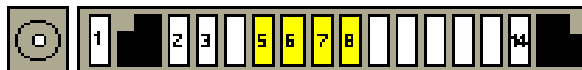


Figura III.17. Conector y pines de datos del nokia 5220

Mando central

Mientras tanto en la segunda parte tenemos un teclado matricial de 4 x 4 botones y una pantalla LCD de 16 x 2 caracteres, ubicados como central donde está el sistema. Aquí se muestra la opción que se fue escogida y la visualización de las instrucciones respectivamente. Dicha etapa se muestra en su diagrama esquemático en la figura III.18.

| PIN NUMBER | PIN NAME | DESCRIPCIÓN |
|------------|----------------|---|
| 1 | Vin | Charger input |
| 2 | GND | Charger ground |
| 3 | ACI | Auto-Connect-Ignition |
| 4 | V Out | |
| 5 | USB Vbus | Also act as USB power detection? |
| 6 | FBus Rx/USB D+ | USB exists at least in Nokia 2280, 3586, 7610 |
| 7 | FBus Tx/USB D- | USB exists at least in Nokia 2280, 3586, 7610 |
| 8 | GND | Data GND |
| 9 | X Mic- | Audio in - Ext. Mic input negativ |
| 10 | X Mic+ | Audio in - Ext. Mic input positiv |
| 11 | HS Ear L- | Audio out - Ext. Audio out - left, negativ |
| 12 | HS Ear L+ | Audio out - Ext. Audio out - left, positiv |
| 13 | HS Ear R- | Audio out - Ext. audio out - right, negativ |
| 14 | HS Ear R+ | Audio out - Ext. audio out - right, positiv |
| | GND | shield GND in cavities |

Tabla III.8. Detalle de pines del bus de datos del Nokia 3220

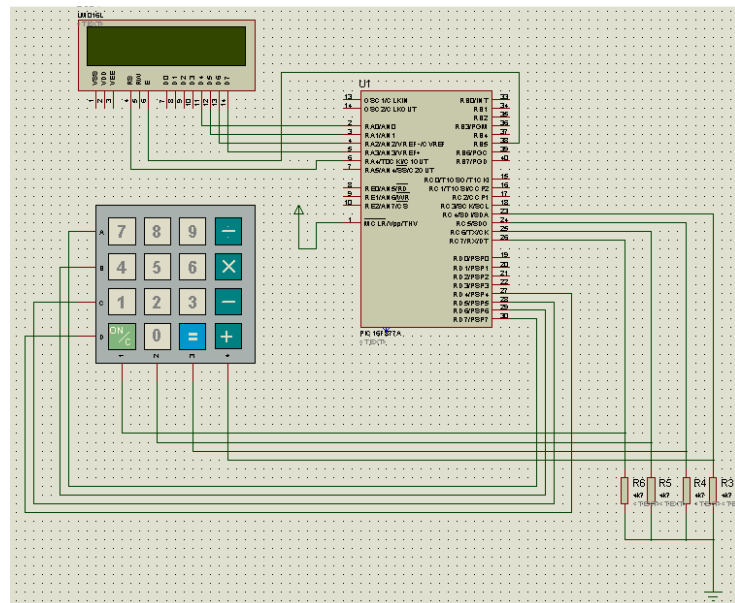


Figura III.18. Esquema de conexiones de teclado y lcd

3.3.2. ETAPA DE COMUNICACIÓN

Una parte muy importante entre estas etapas es la de comunicación y las delicada a la de lectura de datos que tiene que tratarse de una forma más cuidadosa, ya que estas son

leídas en un mismo puerto, debe ser separada de alguna manera para que ninguna interfiera o mezclar una con la otra. El elemento que permite esta tarea es 74LS245 que es un distribuidor de señales con seguimiento de dirección.

Cerrando todo el proceso las mismas instrucciones que llegan del teléfono celular son las mismas que llegan del pic 16F877A y así estas pueden ser ejecutadas en cualquier instante ya por el dispositivo final en este caso es el pic 16F628A. Esta etapa se muestra en la figura III.19.

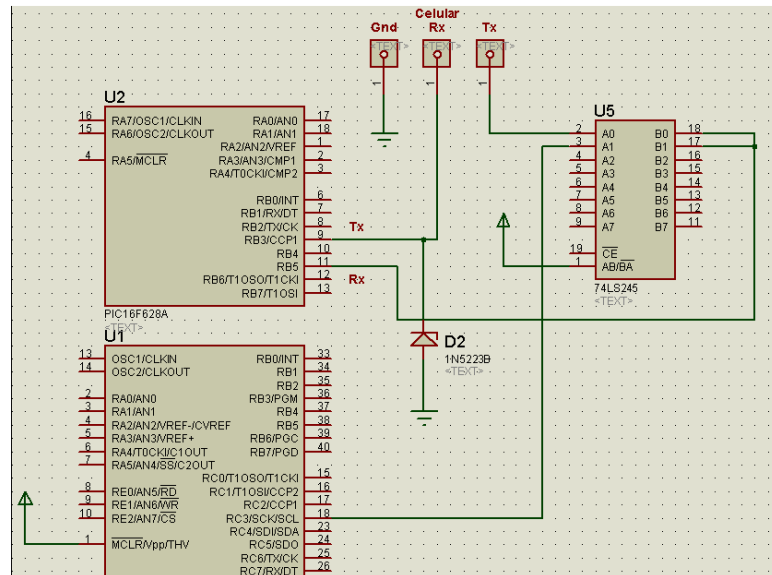


Figura III.19. Etapa de comunicación.

3.3.3. ETAPA DE CONTROL

Esta etapa se distingue el pic 18F877A se encarga del manejo de algunos dispositivos:

- Teclado(entrada y salida)
- LCD(entrada)
- Sensores de Movimiento y de Contacto(Entrada)

3.3.4. ETAPA DE EJECUCIÓN

La información que a petición del usuario fueron enviadas, llegan a esta etapa y aquí son reconocidas y procesadas, ya efectuando la instrucción pero en una etapa binaria aún.

También es la encargada de realizar alguna tareas de corrección o de información al usuario es decir realiza un envío de notificaciones al interesado que en este caso el usuario sobre el estado de los diferentes dispositivos.

3.3.5. ETAPA DE POTENCIA

Ya las instrucciones son binarias y tenues, estas deben ser magnificadas o manejadas con niveles de voltaje que los dispositivos eléctricos necesitan, es decir niveles de voltaje alterno y de 110 voltios.

Por medio de un transistor (2N3904 o Tip 32C), que toma la función de interruptor permitiendo la activación y desconexión, de un relé que soporte el voltaje de contacto de 5 Voltios o de 12 Voltios de corriente continua, a 110 V de corriente alterna, con el fin mencionado anteriormente que es de controlar el encendido de los dispositivos eléctricos, cuyo esquema se muestra en la figura III.20.

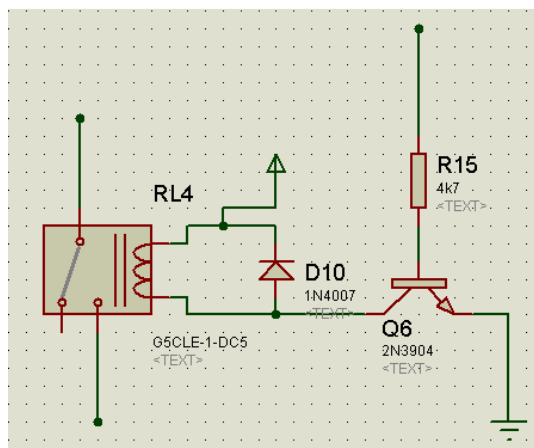


Figura III.20. Etapa de potencia

3.4.DISEÑO DE SOFTWARE

La aplicación celular la desarrollamos bajo la plataforma Java Micro Edition, la cual es compatible con la mayor parte de los teléfonos celulares, además de poseer la funcionalidad para el correcto manejo de las distintas características que poseen los celulares actualmente como son envío – recepción de mensajes de texto, además de permitirnos dar a la aplicación una interface sencilla y funcional. Y particularmente el manejo de SMS mediante Java la realizamos con la API denominada Wireless Messaging, que ya fue detallada.

3.4.1. INTERFACE SOFTWARE DE USUARIO

La idea fundamental en el concepto de interfaz es el de mediación, entre hombre y máquina. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza. Esto implica, además, que se trata de un sistema de traducción, hemos optado por proveer a la aplicación de un menú gráfico, donde cada tarea posee un botón propio con su respectiva imagen y texto acorde a cada tarea a controlar, la navegación por el menú se la realiza con las teclas de dirección, para escoger las acciones se una la tecla OK.

Se han tomado en cuenta diversos factores para el diseño de la interfaz:

Velocidad de Aprendizaje.- Se pretende que la persona aprenda a usar el sistema lo más pronto posible.

Velocidad de Respuesta.- El tiempo necesario para realizar una operación en el sistema.

Tasa de errores.- Porcentaje de errores que comete el usuario.

Retención.- Cuánto recuerda el usuario sobre el uso del sistema en un período de tiempo.

Satisfacción.- Se refiere a que el usuario esté a gusto con el sistema.

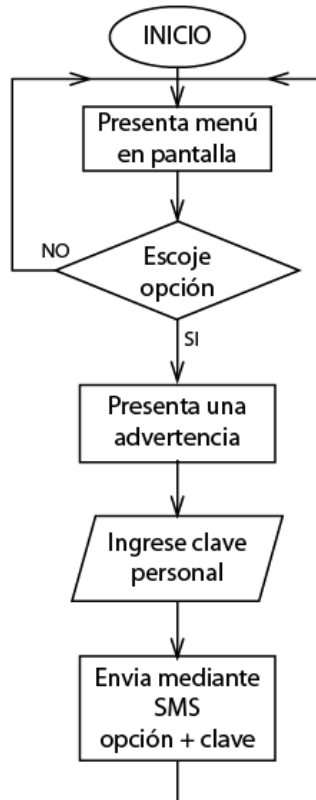


Figura III.21. Diagrama de Flujo del Funcionamiento del Software

La figura III.21. muestra el diagrama de flujo del funcionamiento que tendrá el software para el usuario en el teléfono celular, el cual pretende ser lo más simple posible, con un manejo intuitivo del menú que se presenta en pantalla, el usuario escoge la opción adecuada, se presenta una ventana de advertencia del dispositivo que se va a activar, a continuación se debe ingresar una clave personal, como un tipo de autenticación. Luego el software automáticamente envía el SMS con la información de la clave y la opción seleccionada por el usuario, cabe indicar que el usuario no conoce el formato de dicha información, para evitar que pueda controlar el sistema sin el software que lo acompaña.

Todo esto se ha realizado con el fin de garantizar al usuario mayor comodidad, rapidez y evitar posibles errores.

3.4.2. LISTADO DE ACCIONES

- Activar la cerradura eléctrica de la vivienda.
- Encender o apagar las luces de la vivienda que se hayan conectado a la central.
- Activar el regadío del jardín de la vivienda.
- Botón de pánico o emergencia que activa la sirena del sistema de seguridad.
- Botón mediante el que se desactiva el sistema de seguridad. Al desactivar la seguridad no se recibirán mensajes de alerta ni sonará la sirena al activarse los sensores magnéticos o de movimiento.
- Activar el sistema de seguridad, activada la seguridad sonará la sirena y se recibirán alertas mediante SMS al ser activado alguno de los sensores.
- El estado del sistema se puede conocer al escoger el botón asignado para esta tarea.
- Salir de la aplicación celular.

CAPÍTULO IV

4. IMPLEMENTACIÓN DEL SISTEMA

Los dispositivos de control o mando son el teléfono celular con su aplicación en java y un mando central constituido por un teclado y lcd ubicado en el interior de la vivienda para dar flexibilidad al control del sistema. Los dispositivos controlados por el sistema son los sensores magnéticos e infrarrojo y la sirena en lo que corresponde a seguridad, y además del control de luces, cerradura eléctrica y regadío de la vivienda. Cada uno de los dispositivos puede ser controlado desde el teléfono móvil o desde el mando central ubicado en la vivienda.

El sistema de seguridad consta de sensores magnéticos y sensor infrarrojo que al ser activados y cuando se encuentre el sistema de seguridad encendido, además de activarse la sirena ubicada en la vivienda se enviará un SMS al dueño de la vivienda informándole que ha ocurrido una violación a la seguridad de su domicilio. Además la sirena puede ser activada al pulsar un solo botón desde el mando central o la aplicación java, en caso de emergencia, es el llamado botón de pánico.

Además se incluyó el uso de una clave personal que es solicitada antes de activar o desactivar alguno de los dispositivos desde la aplicación java, y desde el mando central al desactivar o apagar el sistema de seguridad.

4.1.IMPLEMENTACIÓN DEL HARDWARE

4.1.1. CIRCUITO IMPRESO

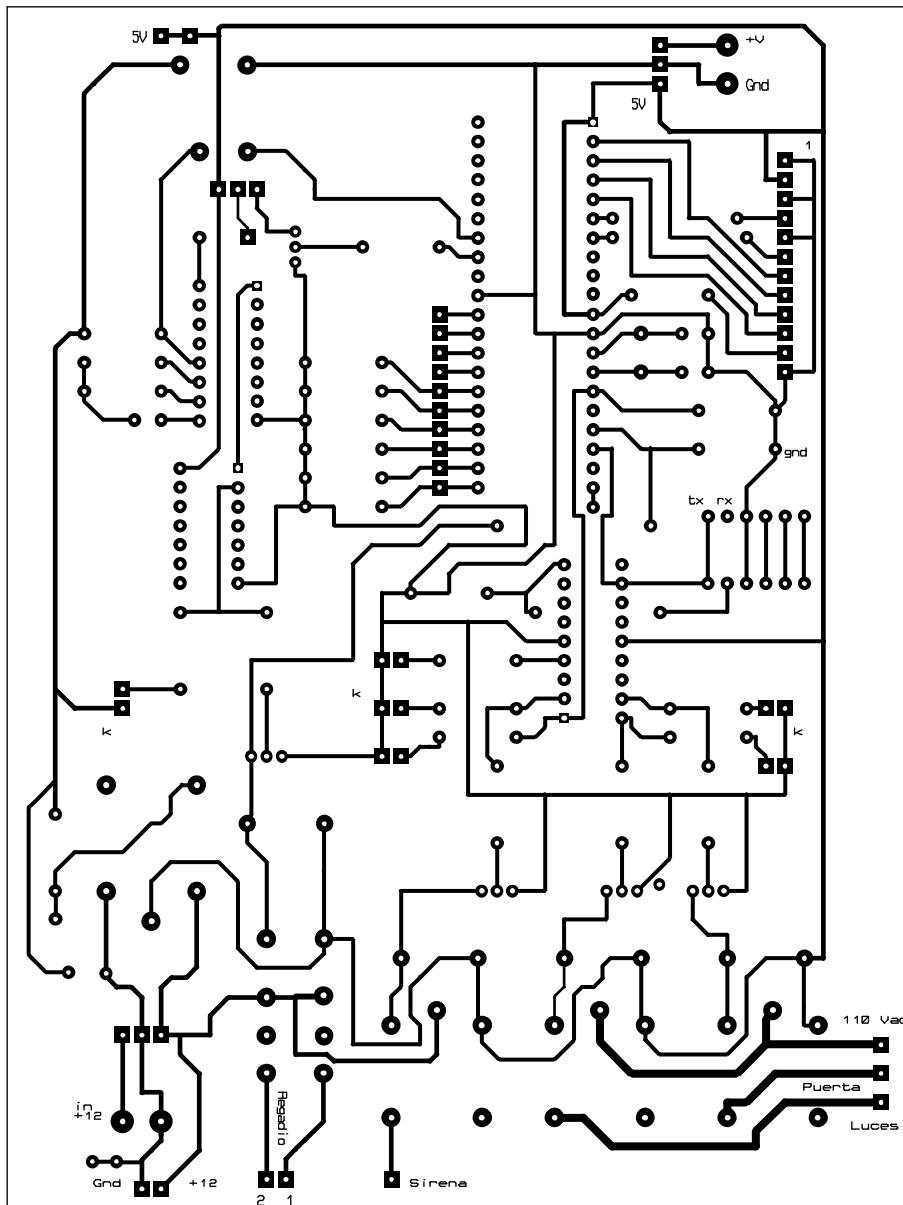


Figura IV.22. Diseño de circuito impreso

El diseño del circuito impreso de la figura IV.22. se lo realizó mediante un software de diseño y simulación electrónica, para obtener un resultado más confiable.

4.1.2. DISPOSICIÓN DE ELEMENTOS

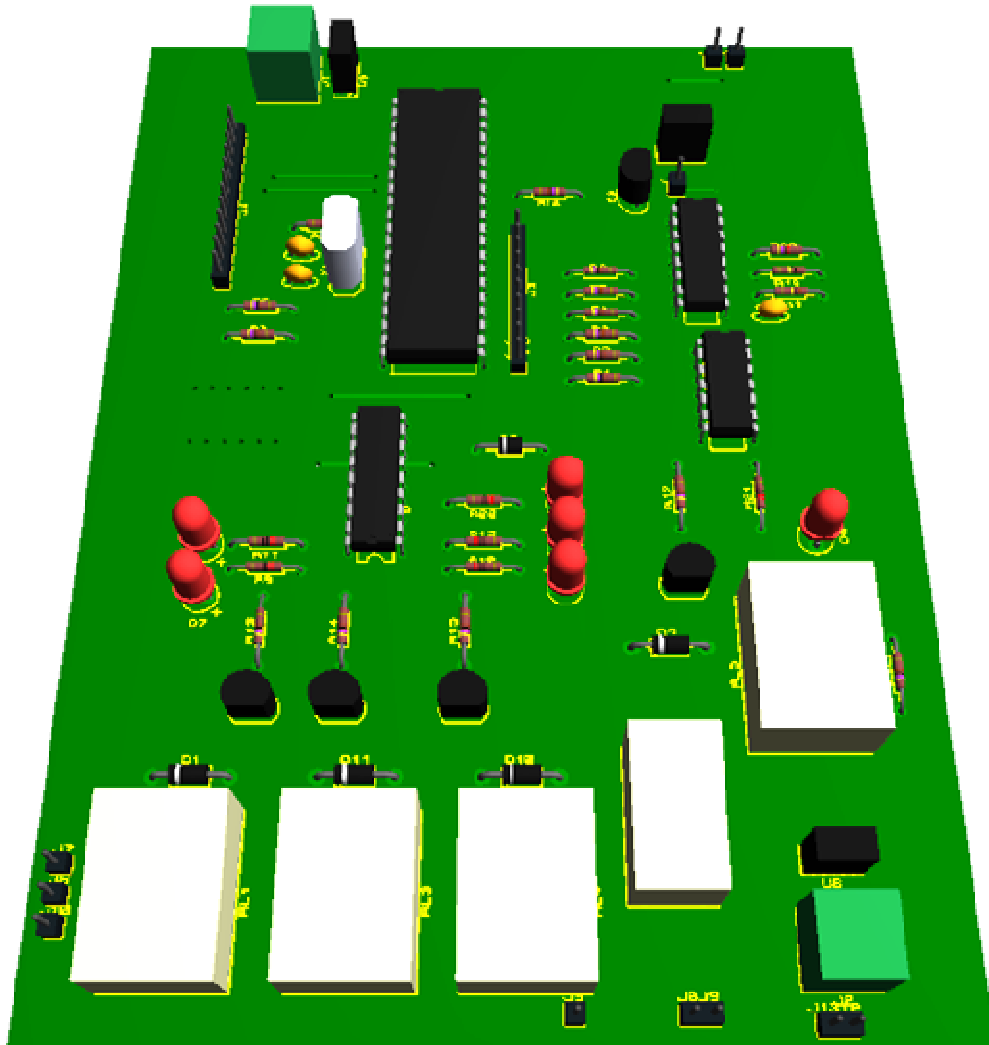


Figura IV.23. Ubicación de elementos en la central del sistema.

4.2.IMPLEMENTACIÓN DEL SOFTWARE

El software para el teléfono móvil está provisto de todas las opciones que se requieren para mantener un completo control del sistema y poder conocer su estado en cualquier

instante. A través de un menú gráfico, de la figura IV.24. el usuario controla fácilmente el dispositivo de su preferencia.

Dentro de la clase principal se lanza la ejecución de un Thread, como ya se explico anteriormente un hilo permite la ejecución de varios procesos al mismo tiempo concurrentemente, este nos permite que el menú de la aplicación se comporte como un subproceso de la aplicación, este se encuentra en constante lectura de las teclas que se presionan.



Figura IV.24. Interface software de usuario.

En Java, un thread puede estar en cuatro estados posibles.

Ejecutándose: está ejecutándose.

Preparado: está preparado para pasar al estado de ejecución.

Suspendido: en espera de algún evento.

Terminado: se ha finalizado la ejecución.

Al software está provisto de los controles necesarios para evitar errores y dificultad en su manejo, como son la presentación de mensajes de advertencia, se muestra en la figura IV.25., que detalla la función o dispositivo a activar o desactivar antes de que dicha acción tenga efecto sobre cualquiera de los dispositivos conectados al sistema domótico.



Figura IV.25. Mensaje de advertencia en el software celular

Como ya se había dicho anteriormente el software de usuario solicita el ingreso de una clave personal para realizar cualquiera de las funciones de que dispone el sistema, esto es para prevenir el uso del software por una persona no autorizada. La pantalla para ingreso de clave personal se muestra en la figura IV.26.

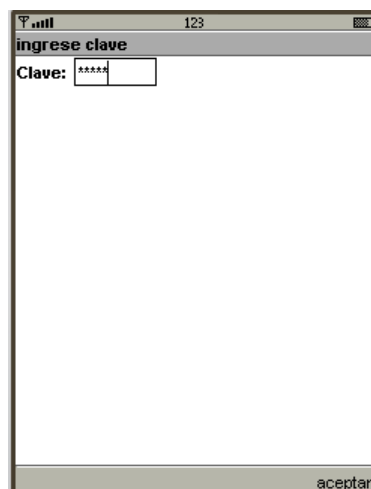


Figura IV.26. Ingreso de clave personal

Además como parte de las seguridades propias de los teléfonos móviles con soporte Java es que antes que cualquier aplicación pueda acceder a la red de datos o red celular así como antes de enviar cada SMS se solicita la confirmación por parte del usuario para realizar dicho envío o conexión. Dicha confirmación se muestra en la figura IV.27.

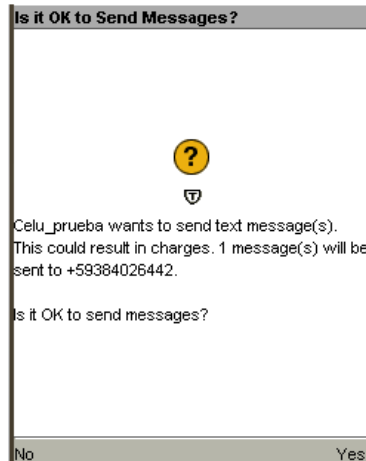


Figura IV.27. Solicitud de confirmación para envío de SMS



Figura IV.28. Interface software de usuario.

La figura IV.28. muestra el mensaje de advertencia que el software presentará antes de solicitar a la central que le informe al usuario sobre el estado de sus dispositivos y de su sistema de seguridad.

4.3.INSTALACIÓN

Para la instalación del sistema se analizó la vivienda para poder ubicar los dispositivos de manera adecuada, tomando en cuenta los requerimientos de seguridad y comodidad de sus habitantes. La figura IV.29 muestra la vivienda en que se instaló el sistema domótico.



Figura IV.29. Vivienda en que se instaló el sistema.

Se buscó que la ubicación del panel de control preste facilidad para el acceso al usuario, ya que con este se pueden controlar todos los dispositivos conectados al sistema y el sistema de seguridad. Para la ubicación de la central se buscó un lugar que preste la facilidad para el cableado de todos los dispositivos, ya que, este es el cerebro del sistema y de este salen todas las conexiones, es por esto que la ubicación de este debe ser estratégica. La figura IV.30. muestra la ubicación de la central y del panel de control.

La figura IV.31. muestra la central del sistema con todas sus conexiones, en la parte derecha esta la fuente de poder que lo alimenta con voltajes de 12Vcd y 5Vcd, en la parte central el modem celular. Puesto que posee su propia fuente de alimentación la central posee una sola toma de corriente. En la parte inferior se observa la bornera donde

se conectan los distintos dispositivos como son: iluminación, cerradura eléctrica, sirena, salida temporizada, y alimentación para módem.



Figura IV.30. Conexión entre el panel de control y la central.

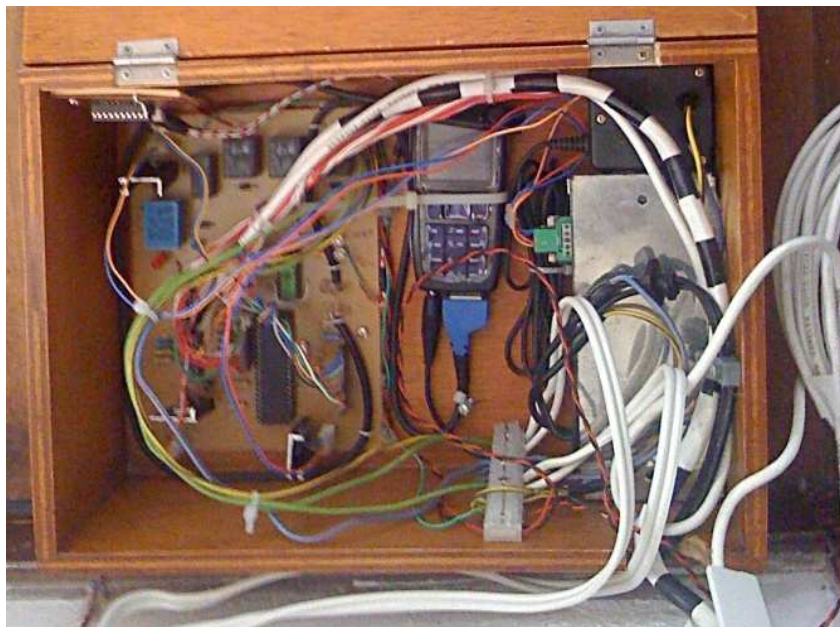


Figura IV.31. Central del sistema.

El mando central es el que se indica en la figura IV.32. se encuentra dentro de la vivienda para poder ofrecer al usuario la posibilidad de controlar todos los dispositivos

del sistema incluido el sistema de seguridad, su funcionamiento se explica detalladamente en el manual de usuario en el anexo 1.



Figura IV.32. Panel de control.

Los dispositivos que conforman el sistema de seguridad son: contacto magnético para puertas, figura IV.33, sensor infrarrojo de movimiento, figura IV.34, y sirena, figura IV.35, los sensores alertan sobre un intruso en la vivienda mediante la sirena y un SMS al usuario, solo en caso que la seguridad del domicilio este activada.



Figura IV.33. Contacto magnético.



Figura IV.34. Sensor infrarrojo de movimiento.



Figura IV.35. Sirena.

La parte de control de dispositivos se aplicó a la iluminación y cerradura eléctrica, que son solo una pequeña muestra de la cantidad de dispositivos que se podrían controlar. Para la iluminación la central se encarga de proporcionar el voltaje de 110Vca para alimentar los focos, en lo que respecta a la cerradura eléctrica, la central solamente funciona como interruptor. Estos dispositivos los observamos en la figura IV.36. y la figura IV.37.



Figura IV.36. Iluminación conectada al sistema.



Figura IV.37. Conexiones para cerradura eléctrica.

En la figura IV.38. se muestra como está ubicado el panel de control con respecto a la central, la ubicación que se escogió responde a prestar comodidad para realizar las conexiones.



Figura IV.38. Central y panel de control.

La figura IV.39. muestra la puerta de acceso a la vivienda y junto a esta el panel de control, de manera que preste la facilidad para el control de todos los dispositivos.



Figura IV.39. Ubicación del panel de control.

4.4.PRUEBAS

El sistema esta implementado y en funcionamiento óptimo, para llegar a este resultado se realizó una serie de pruebas, de hardware, software, dispositivos y conexiones.

Las pruebas que se realizó en lo referente a la central o circuito principal del sistema son:

- Niveles de voltaje apropiados.
- Eliminar posibles errores en la comunicación serial entre los microcontroladores y con el modem celular.
- Depurar errores de programación de los microcontroladores evitando comportamiento inadecuado.
- Se añadieron leds indicadores para poder analizar por etapas posibles fallos en el circuito.
- Comprobar la ejecución de los comandos AT por el microcontrolador en el módem celular.
- Simular una intrusión en el domicilio para comprobar el envío de los mensajes de advertencia al dueño del domicilio.

Pruebas de software:

- Comprobar el envío de los mensajes desde el software java y con el formato adecuado.
- Pruebas de funcionamiento del menú del programa en lo referente a la navegación dentro del mismo y la facilidad para ubicar las diferentes opciones.

- Probar la funcionalidad de cada una de las opciones y su control sobre el sistema domótico.

Pruebas de dispositivos:

- Pruebas de funcionamiento correcto de los dispositivos, independientemente del sistema domótico para saber sobre sus voltajes de funcionamiento en el caso de iluminación y cerradura eléctrica. Tiempos de respuesta, alcance, voltajes de alimentación para sensor infrarrojo, magnético y sirena.
- Comprobación de funcionamiento del mando central y visualización en su lcd.
- Pruebas sobre el sistema de seguridad para depurar la programación evitando falsas alarmas.

Pruebas de conexión:

- Probar que las conexiones no sean causa de mal funcionamiento de ninguno de los componentes del sistema es importante para evitar posibles complicaciones y demoras localizando alguna falla en caso de presentarse.

CONCLUSIONES

- Mediante el uso de programación JME, comandos AT, microcontroladores, y el uso de la red celular mediante SMS se ha logrado dar una alternativa económica para generar servicios de control domótico y seguridad domiciliaria, utilizando una interface amigable.
- Existen cada día más dispositivos dentro de un hogar a los que se puede aplicar control domótico, en el desarrollo del presente trabajo se ha utilizado los más básicos servicios de un hogar como son luces, puertas y sistema de seguridad, para dar una idea del alcance que tiene el sistema implementado.
- Se han analizado los distintos dispositivos de un hogar y se ha aplicado control domótico a cuatro que son: luces, cerradura eléctrica, sistema de seguridad, conformado por sensores de movimiento, magnéticos de puertas y sirena.
- Se añadió un mando central para poder controlar los dispositivos y la seguridad del domicilio sin necesidad del software celular, en caso que no exista necesidad de hacerlo desde el teléfono celular.
- El sistema si bien utiliza la red celular existente y el servicio de SMS de las compañías de telefonía, por ser independiente de estas, no conlleva ningún costo adicional y amplía la gama de servicios actualmente disponible en el área de los teléfonos celulares.

- Se ha diseñado el software de usuario, bajo la plataforma Java Micro Edition de Java, se instaló en un teléfono celular que soporte dicha tecnología y se probó su funcionamiento para controlar efectivamente los distintos dispositivos que se conectaron a nuestro sistema.

RECOMENDACIONES

- Evitar conexiones erróneas, puesto que también controlamos dispositivos de corriente alterna a 110 voltios, que podrían causar corto circuito y dañar los microcontroladores y demás elementos.
- Para la aplicación Java se recomienda el uso de un celular que permita dar privilegios a las aplicaciones, como son para acceder a la red celular y permitir el envío de mensajes de texto desde la aplicación, de lo contrario el celular bloqueará el intento de la aplicación de enviar los comandos para el sistema.

RESUMEN

Se ha diseñado, implementado e instalado un sistema domótico para control y seguridad en tiempo real vía teléfono celular, proporcionando una herramienta de control de cerradura eléctrica, iluminación y sistema de seguridad conformado por sensor magnético de puertas, sensor infrarrojo de movimiento y sirena.

El teléfono celular para controlar el sistema utiliza un software diseñado para dicha tarea, presenta una interface sencilla para controlar por completo el sistema. La central se encarga de notificar al usuario mediante un mensaje de texto si ha existido alguna violación de la seguridad.

Se utilizó la plataforma Java para el desarrollo de la aplicación celular que permite el envío de mensajes de texto de forma transparente para el usuario, el mensaje de texto es recibido por la central del sistema mediante un modem GSM, el microcontrolador PIC16F628A se encarga de leer dicho mensaje y verificar si posee la sintaxis correcta para activar o desactivar el dispositivo indicado. Se añadió un panel de control conformado por un teclado matricial y un LCD el cual es controlado mediante el PIC16F877A. El sistema brinda seguridad, rapidez y comodidad en el control domótico. Se obtuvo un sistema que provee comodidad al usuario y un bajo costo de operación en relación con el beneficio de poder controlar los dispositivos conectados al sistema domótico y la seguridad de su domicilio, desde su propio teléfono celular.

Se ha logrado reducir el tiempo de respuesta ante una posible intrusión en el domicilio hasta en un 50%. El control domótico toma alrededor de 8 segundos sin importar donde se encuentre el usuario.

Se recomienda la instalación de este sistema por adecuarse fácilmente a las necesidades de seguridad y control domótico.

SUMMARY

It has been designed, implemented and installed an home automation system for security and control in real-time via cell phone, providing a tool to control electric locks, lighting and security system comprised of magnetic door sensor, infrared motion sensor and siren.

The cell phone to control the system uses software designed for this task; it presents a simple interface to control the entire system. The central is responsible for notifying to the user via text message if there has been any violation of security.

We used the Java platform for developing mobile application that allows text messages sent transparently to the user, the text message is received by the central system via a GSM modem, the PIC16F628A microcontroller is responsible for reading the message and verify if it has the correct syntax to enable or disable the device indicated. A control panel was added consisting of a matrix keyboard and an LCD which is controlled by the PIC16F877A. The system provides security, speed and convenience in automation control. The result was a system that provides user comfort and low operating cost in relation to the benefit of controlling devices connected to home automation and security system for your home, from his own cell phone.

It has reduced the response time to a possible intrusion in the home as much as 50%. The automation control takes about 8 seconds no matter where user is located.

Installation of this system is recommended because easily adapts to the needs of security and home automation control.

ANEXOS

ANEXO 1

MANUAL DE USUARIO

Introducción

Los sistemas que en la actualidad se usan, tienen un alto costo en mantenimiento y monitoreo, con alertas a terceros que dependiendo de los problemas que pudieren presentarse, los usuarios beneficiarios son los últimos en ser notificados de las alertas sobre su seguridad o de sus bienes.

El DOMFOR-TECH es un sistema sencillo con un bajo costo de mantenimiento, fácil y práctico de usar, para el usuario final, permitiéndole de una manera cómoda conocer y controlar el estado de los dispositivos y sistema de seguridad de su domicilio, sin importar el lugar donde el usuario se encuentre por medio de un instrumento económico y altamente difundido, como es un teléfono celular (que soporte aplicaciones java).

Con este dispositivo podemos realizar las respectivas acciones que el usuario considere ejecutar, y a la vez obtener información del estado de estas.

La ventaja primordial del sistema domótico y de seguridad DOMFOR-TECH, es que el principal interesado sobre la seguridad y comodidad de su domicilio tiene la posibilidad de estar enterado instantáneamente de lo que ocurre en su domicilio, permitiéndole tomar las medidas pertinentes a su elección, ya sea notificar a las autoridades, grupos barriales de seguridad, vecinos, etc. Para deslindar tardanzas por encontrarse a mucha distancia que evitaría una oportuna atención.

El DOMFOR-TECH permite comodidad ya que permite usar su teléfono celular como un control remoto para manejar diferentes equipos eléctricos de su hogar sin la necesidad de estar presente cerca de los mismos y es mas apagarlos en caso de que estén funcionando, teniendo en todo momento información del estado de estos.

En cuanto al costo de operación, es relativamente bajo, ya que por una orden o por una notificación entre el sistema y el usuario, representa un simple mensaje de texto cuyo

costo depende de la operadora celular a la que este afiliado el usuario, ya que es su elección, ya sea por preferencia del usuario o por prestaciones.

En lo referente a la parte de operación, el uso de un móvil con la aplicación java del sistema, muestra un agradable y práctico menú gráfico, fácil de entender y manipular, pero sin dejar de lado lo que es primordial, lo que se refiere a la seguridad con clave personal para realizar operaciones, evitando problemas y posibles violaciones en el sistema.

El uso adecuado de las instrucciones a continuación mostradas le permitirán a usted conocer el método de manejo y reacción del DOMFOR-TECH, para evitar contrariedades y problemas.

Funcionalidades del DOMFOR-TECH

Suministra una comunicación de parte y parte en un costo representativamente bajo, como es el de un mensaje de texto, que dependiendo de la operadora, sus costo es sumamente bajo. A más de eso el hecho de poder controlar 4 dispositivos domésticos, a distancia, con comodidad y con las funcionalidades que estos elementos presten.

Elementos del DOMFOR-TECH

Elementos a Controlar

Dentro de lo que se relaciona con el control domótico o sea de los dispositivos a controlar, el sistema DOMFOR-TECH permite el control de la iluminación, es decir manejar el encendido y apagado de estos elementos que en este caso vendrían a ser focos, luminarias, fluorescentes, etc.

Otro de los dispositivos que se controla es la cerradura eléctrica, en este caso permite accionar el interruptor para únicamente manejar el desbloqueo de la puerta.

Y por ultimo tenemos la sirena, que no es más que un dispositivo que permite alertar a la gente alrededor sobre una posible violación o invasión de la propiedad, en el caso de un intruso.

Elementos de Seguridad

Mientras tanto con lo que refiere a la seguridad poseemos la sirena mencionada anteriormente un sensor de movimiento infrarrojo y sensores magnéticos, que deberán ser colocados estratégicamente por una persona con los conocimientos necesarios. Estos dispositivos permiten que se alerte en caso de invasiones en el área protegida, con el uso de la sirena. Estos también permiten que se verifique que nadie ajeno, esté dentro del lugar que se requiera vigilar (sensor de movimiento infrarrojo) y controlar que todos los accesos estén correctamente asegurados.

Modos de Funcionamiento

Existen tres modos de funcionamiento, cuando se activa el sistema de seguridad, cuando este está en reposo y cuando existe una alerta en el sistema.

| |
|--|
| Los elementos a controlar siempre estarán en funcionamiento en cual quiere que fuere el modo en el que se encuentre. |
|--|

Modo Activo de Seguridad

Todos los dispositivos de seguridad (con excepción de la sirena) están funcionando, los sensores están pendientes de cualquier anomalía en el domicilio. A la mínima alteración de algo en este caso se activara las alertas, que en nuestro caso es la activación de la sirena y notifica al usuario mediante un mensaje de texto.

Modo Desactivado de Seguridad

En este modo el DOMFOR-TECH se encuentra en reposo, es decir no están en funcionamiento lo relacionado con la seguridad es decir los sensores magnéticos y el de movimiento.

Modo Alerta

Es el que activa la sirena, y notifica al usuario mediante mensajes de texto que la seguridad ha sido violada, ya que hay un intruso en el domicilio o que alguien intenta ingresar al mismo.

Modo de Uso del DOMFOR-TECH

El DOMFOR-TECH tiene dos modos de ser controlado, primero, mediante el panel central, que es la caja de control ubicada dentro del domicilio mediante el cual se puede controlar todos los dispositivos conectados al sistema, y por otro lado, mediante el mando inalámbrico (teléfono celular) que con la aplicación realiza las mismas operaciones como el panel central. Estos dos permiten controlar, activar o desactivar cualquiera de los dispositivos mencionados anteriormente.

Panel central

Posee 3 partes, el visor, la señal de activación y el teclado

El Visor:

Muestra en palabras las operaciones que está ejecutando el DOMFOR-TECH, para que el usuario identifique claramente el dispositivo activado o desactivado.

Las instrucciones que se muestran son:

PORTERO ELÉCTRICO: Nos indica que el cerrojo eléctrico ha sido activado, por lo tanto la puerta está lista para ser abierta.

LUCES: Indica que los focos, fluorescentes, etc. han sido activados o prendidos.

SEGURIDAD ACTIVADA: muestra que la seguridad está en funcionamiento es decir en **modo activo de seguridad**.

REGADÍO: anuncia que se ha ejecutado un dispositivo eléctrico, que está programado para que funcione un tiempo determinado (20 min de funcionamiento).

SEGURIDAD DESACTIVADA: indica que el sistema está en **modo desactivo de seguridad**.

INGRESO CLAVE DE SEGURIDAD: Aviso de que la clave debe ser ingresada por el usuario para constatar la identidad del usuario y ejecución de las instrucciones deseadas.

CLAVE CORRECTA: Recalca que la clave ingresada fue correcta y que se ha detenido la alerta del **modo de seguridad** que fué puesta en ejecución.

CLAVE INCORRECTA: Indica que ocurrió un error en el ingreso de la clave de seguridad, en este caso deberá ser nuevamente ingresada, pero cuando el mensaje de **ingreso de clave de seguridad** se muestre nuevamente en pantalla.

Señal de Activación:

Es una luz que indica que el **modo activo de seguridad** está en marcha, cuando esta está encendida.



Teclado:

Este es el encargado de dar las órdenes, es el control central que consta de un conjunto de botones, que están clasificados en tres grupos.

Numéricos: Simplemente se usan para el ingreso de la clave, cuando el sistema muestre en el visor el mensaje de **ingreso de clave de seguridad**. La clave de seguridad consta de 5 caracteres numéricos que por cuestiones de seguridad estos no serán mostrados en el visor.

Letras: Estos botones ejecutarán las instrucciones que vienen remarcadas junto a estos:

A Portero Eléctrico:

Basta con presionar una sola vez el botón y se activará la cerradura eléctrica.

B Luces

Al igual que el anterior basta con presionar una sola vez, y los focos o cualquier dispositivo de iluminación se encenderán, y de la misma forma, presionando nuevamente este las apagará.

C Regadío

Presionando este botón, activa un dispositivo que funcionará un cierto tiempo (20 min) y luego se desactivará automáticamente.

D Sirena

Al presionar este botón, se activa el **modo alerta**. Ya que este es un botón de emergencia.

Símbolos: Son los encargados del manejo de lo que es la seguridad.

* Ingresa al **modo activo de seguridad**, al ser pulsado este el usuario tiene un tiempo de espera, hasta que se pueda abandonar el domicilio, antes que los sensores empiecen a funcionar (15 seg.).

Desactiva la seguridad cuando está activado el **modo activo de seguridad**, o por alguna razón se encuentra en el **modo alerta**, pero este viene seguido del **ingreso de clave de seguridad**, para que esta instrucción sea ejecutada, ya que en caso contrario esta seguirá en marcha. El usuario ya ha sido notificado mediante un mensaje SMS al teléfono celular.

Control Inalámbrico

Este es el mencionado teléfono celular que en este caso controlará todas las instrucciones que posee el panel de control, con ligeras modificaciones.

Todas las instrucciones que se quieran ejecutar desde el teléfono celular, necesitan del ingreso de la clave de seguridad, esto para evitar contratiempos por pérdida, falsos usuarios, errores de manipulación, etc.

El menú que muestra las opciones se muestra a continuación:



Cada grafico representa las opciones del sistema:



Portero Eléctrico:

Basta con presionar una sola vez el botón arriba indicado y se activará la cerradura eléctrica.



Luces

Al igual que el anterior basta con presionar una sola vez, y los focos o cualquier dispositivo de iluminación se encendera, y de la misma forma, presionando nuevamente este las apagara.



Regadío

Presionando este botón, activa un dispositivo que funcionará un cierto tiempo (20 min) y luego se desactivará automáticamente.



Sirena

Al presionar este botón, se activa el **modo alerta**. Ya que este es un botón de emergencia.



Ingresa al **modo activo de seguridad**, al ser pulsado este el usuario tiene un tiempo de espera, hasta que se pueda abandonar el domicilio, antes que los sensores empiecen a funcionar (15 seg.).



Desactiva la seguridad cuando está activado el **modo activo de seguridad**, o por alguna razón se encuentra en el **modo alerta**, pero este viene seguido del **ingreso de clave de seguridad**, para que esta instrucción sea ejecutada, ya que en caso contrario esta seguirá en marcha. El usuario ya ha sido notificado mediante un mensaje SMS al teléfono celular.



Permite obtener un estado de todos los dispositivos en el sistema, es decir cual está en ejecución y cuales están apagados



La opción de salir de la aplicación, será permitida con escoger este botón.

ANEXO 2

CÓDIGO FUENTE

Se han creado tres clases dentro de la aplicación celular:

- Clase Principal: contiene los métodos básicos para la ejecución de la aplicación, startApp, pauseApp, destroApp.
- Clase Sprite: presenta los métodos para el manejo gráfico de pantalla.
- Clase M_grafic: aquí se realiza la mayor parte de los procesos de la aplicación.

CLASE PRINCIPAL

```
import java.lang.Thread.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Principal extends MIDlet implements CommandListener {

    Display pantalla;
    M_grafic tablero;
    Form config;
    Command aceptar;
    TextField clave;

    public Principal() {
        pantalla=Display.getDisplay(this);
        tablero = new M_grafic(this);
        config =new Form("ingrese clave");
        aceptar = new Command("aceptar",Command.OK,1);
        clave = new TextField("Clave:
        ", "",5,TextField.NUMERIC|TextField.PASSWORD);
        config.addCommand(aceptar);
        config.append(clave);
    }

    public void startApp() {
        tablero.ejec=true;
        new Thread(tablero).start();
        pantalla.setCurrent(tablero);
    }

    public void pauseApp() {
```

```

    }

    public void destroyApp(boolean unconditional) {

    }

    public void salir(){
        tablero.ejec=false;
        destroyApp(false);
        notifyDestroyed();
    }

    public void commandAction(Command c, Displayable d) {
        String data;
        data=clave.getString();

        if(c == aceptar){

            if (data.equals("")){
                config.setTitle("Campo vacio");
                config.setCommandListener(this);
                pantalla.setCurrent(config);

            }else{
                new Thread(tablero).start();
                pantalla.setCurrent(tablero);
                tablero.ejec=true;
            }
        }
    }

    public void ingre_clave(){

        pantalla=Display.getDisplay(this);
        tablero.ejec=false;
        config.setCommandListener(this);
        pantalla.setCurrent(config);

    }
}

```

CLASE M_GRAFIC

Envío De Mensajes

```
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.TextMessage;

public MessageConnection conn;

try{
    conn = (MessageConnection)
    Connector.open("sms://+59384026442");
} catch (Exception e){}

public void enviosms(String mensaje) {
    try{
        TextMessage tmsg = (TextMessage)
        conn.newMessage(conn.TEXT_MESSAGE);
        tmsg.setPayloadText(mensaje);
        conn.send(tmsg);
    }
    catch(Exception e){ }
}

try{
    conn.close();
} catch (Exception e){}
```

Aquí se define un método que crea y envía un mensaje de texto (TextMessage). Se ha añadido el paquete MessageConnection, el cual define todas las interfaces para envío y recepción inalámbrica de mensajes de texto y binario, y la sub-interface TextMessage. Se puede verificar que la conexión es un cliente, al crear la conexión se ha especificado la dirección de destino. Antes de enviar el mensaje de texto el método carga el mensaje a enviar, que ha sido tomada por la implementación, llamando el método setPayloadText().

Manejo Del Teclado

```
if ((keyStates & LEFT_PRESSED) != 0)
    if(posm!=1) posm-=1;
    else posm=9;
```

```

if ((keyStates & RIGHT_PRESSED) != 0)
    if(posm!=9) posm+=1;
    else posm=1;

if ((keyStates & UP_PRESSED) != 0)
    if((posm!=1)&(posm!=2)&(posm!=3)) posm-=3;
    else
        posm+=6;

if ((keyStates & DOWN_PRESSED) != 0)
    if((posm!=7)&(posm!=8)&(posm!=9)) posm+=3;
    else
        posm-=6;

```

En el código anterior se controla la pulsación de las teclas de dirección o cursores, keyStates devolverá un valor diferente de 0 al ser pulsada una tecla, además se identifica que tecla que fue pulsada (arriba, abajo, izquierda o derecha) para realizar la acción correspondiente en el menú gráfico.

```

if (((keyStates)& (FIRE_PRESSED))!=0){
    switch(posm){

        case 1:{
            dibu_alerta(1);
            ((Principal)midlet).ingre_clave();
            stop();
            data=((Principal)midlet).clave.getString();
            enviosms(data+"A");//abrir cerradura
            ((Principal)midlet).clave.setString("");
            break;
        }

        case 2:{
            dibu_alerta(2);
            ((Principal)midlet).ingre_clave();
            stop();
            data=((Principal)midlet).clave.getString();
            enviosms(data+"B");//encender luces
            ((Principal)midlet).clave.setString("");
            break;
        }

        case 3:{
            dibu_alerta(4);
            ((Principal)midlet).ingre_clave();
            stop();

```

```

        data=((Principal)midlet).clave.getString();
        enviosms(data+"C");//encender regadio
        ((Principal)midlet).clave.setString("");
        break;

    }
    case 4:{
        dibu_alerta(6);
        ((Principal)midlet).ingre_clave();
        stop();
        data=((Principal)midlet).clave.getString();
        enviosms(data+"F");//activar sirena
        ((Principal)midlet).clave.setString("");
        break;

    }
    case 5:{
        dibu_alerta(7);
        ((Principal)midlet).ingre_clave();
        stop();
        data=((Principal)midlet).clave.getString();
        enviosms(data+"G");//desactivar seguridades
        ((Principal)midlet).clave.setString("");
        break;
    }

    case 6:{
        dibu_alerta(9);
        ((Principal)midlet).ingre_clave();
        stop();
        data=((Principal)midlet).clave.getString();
        enviosms(data+"H");//activar seguridades
        ((Principal)midlet).clave.setString("");
        //enviosms("seguridad on");
        break;

    }

    case 7:{

        break;

    }

    case 8:{
        dibu_alerta(5);
        ((Principal)midlet).ingre_clave();
        stop();
        data=((Principal)midlet).clave.getString();
        enviosms(data+"E");//controlar estado

```



```
        ((Principal)midlet).clave.setString("");
        break;
    }

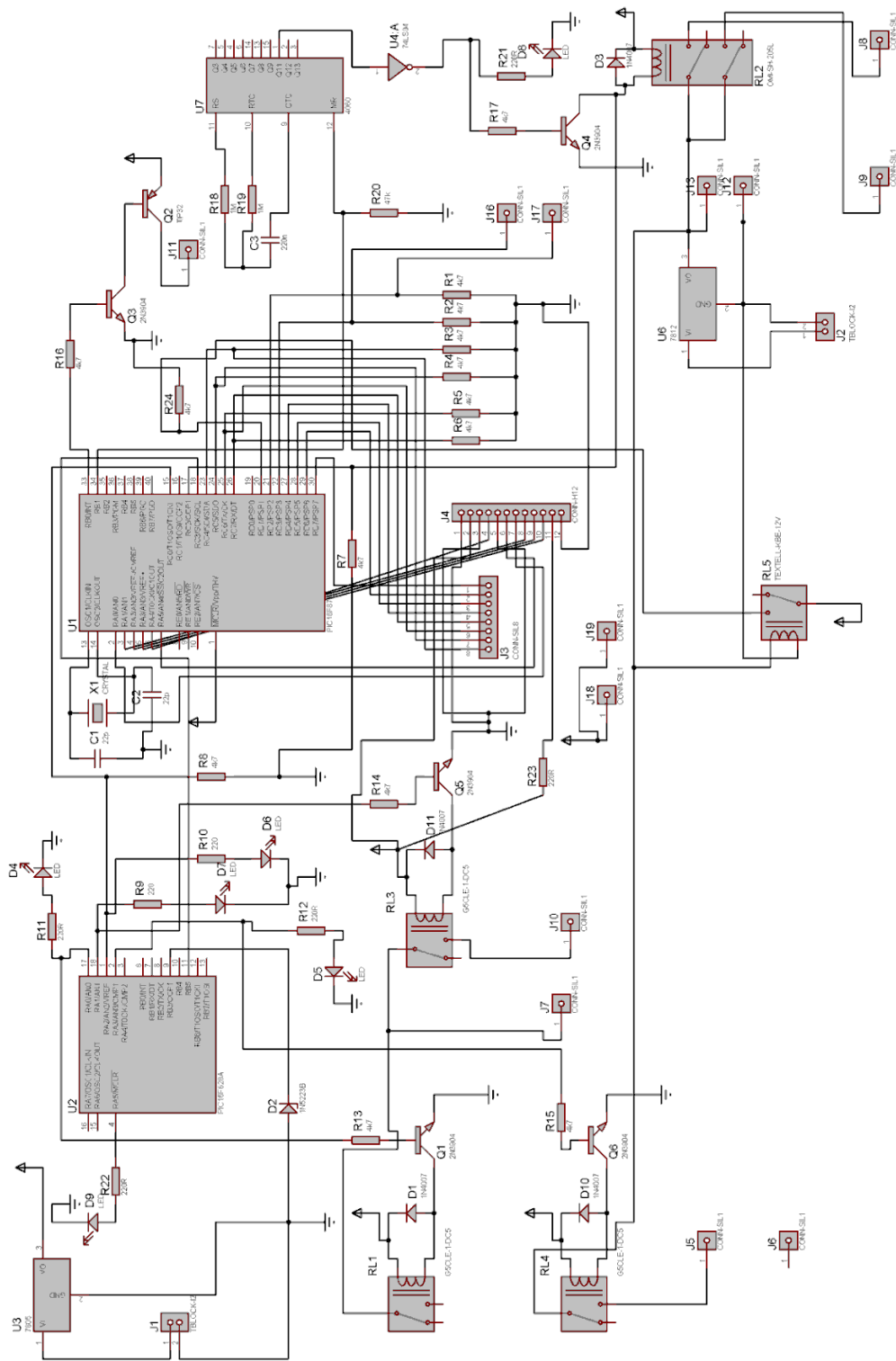
    case 9:{
        try{
            conn.close();
        }catch (Exception e){}

        ((Principal)midlet).salir();
    }
}
```

El código que precede detecta la pulsación de la tecla FIRE o tecla OK, además determina la posición actual del cursor dentro del menú del usuario para realizar el envío del mensaje de texto apropiado a cada una de las opciones, además de obtener la cadena de caracteres correspondientes a la clave personal del usuario.

ANEXO 3

DIAGRAMA ESQUEMÁTICO



BIBLIOGRAFÍA

1. GÁLVEZ, Sergio y ORTEGA, Lucas. Java a Tope: J2ME (Java 2 Micro Edition). Edición electrónica. Malaga – España. Universidad de Malaga. 188p.
2. JORGE, Patricia y FROUFE, Agustín. (2004). J2ME. Java 2 MicroEdition. Manual de usuario y tutorial. 3ed. Madrid – España. RA-MA. 592p.
3. REYES, Carlos A. (2006). Microcontroladores PIC Programación en Basic. 2ed. Quito – Ecuador. Automasis. 211p.

BIBLIOGRAFÍA DE INTERNET

1. ACTIVEXPERTS SOFTWARE B.V. - Nokia GSM AT Command Set.
<http://www.activexperts.com/xmstoolkit/atcommands/nokia/>
(20 – 11 – 2009)
2. CHAPARRO, Jeffer. - Domótica: La Mutación de la Vivienda.
[http://www.ub.es/geocrit/sn/sn-146\(136\).htm](http://www.ub.es/geocrit/sn/sn-146(136).htm)
(12 – 01 – 2010)
3. FORUM.NOKIA.COM. - AT commnads.
http://wiki.forum.nokia.com/index.php/AT_Commands
(15 – 11 – 2009)
4. GARCÍA, Carlos. J2ME, Java Wireless Message API (WMA).
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=wma>
(25 – 11 – 2009)

5. INZAURRALDE, Martín y otros. Telefonía Celular.

<http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular.shtml>

(10 – 01 – 2010)

6. MERLYNCK@NETSCAPE.NET. - Microcontroladores.

<http://www.monografias.com/trabajos12/microco/microco.shtml>

(10 – 01 – 2010)

7. MICROENGINEERING LABS, INC. - PicBasicPro en Castellano.

http://www.todopic.com.ar/pbp_sp.html

(10 – 11 – 2009)

8. NATIONAL INSTRUMENTS CORPORATION. - Comunicación Serial: Conceptos Generales.

<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>

(21 – 02 – 2010)

9. PINOUTS.RU TEAM. - Nokia Pop-port pinout and wiring.

http://pinouts.ru/CellularPhones-Nokia/nokia_pop_pinout.shtml

(20 – 11 – 2009)

10. TOBOSO, Emilio. - La comunicación serie.

<http://perso.wanadoo.es/pictob/comserie.htm>

(18 – 02 – 2010)