



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MEXICO**

**PROGRAMA DE MAESTRIA Y DOCTORADO
EN INGENIERIA**

FACULTAD DE INGENIERIA

**SISTEMA DE RECONOCIMIENTO DE
VOZ CON DISPOSITIVOS MOVILES,
APLICADO A CASAS INTELIGENTES.**

T E S I S

QUE PARA OBTAR POR EL GRADO DE:
MAESTRO EN INGENIERIA.
CAMPO DE CONOCIMIENTO: ELECTRICA.
CAMPO DISIPLINARIO: TELECOMUNICACIONES.
P R E S E N T A:

LOVERA GRANADOS FABIAN.



TUTOR:
DR. BOHUMIL PSENICKA.

AÑO:
JUNIO 2010



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. García Ugalde Francisco.

Secretario: Dr. Orduña Bustamante Felipe.

Vocal: Dr. Psenicka Bohumil.

1^{er} Suplente: Dr. Gómez Castellanos Javier.

2^{do} Suplente: M. I. Escobar Salguero Larry.

Lugar o lugares donde se realizó la tesis:
CIUDAD UNIVERSITARIA, MEXICO DF.

TUTOR DE TESIS:
DR. BOHUMIL PSENICKA.



FIRMA

A mi madre...

Agradecimientos:

A todos mis amigos, profesores y familiares que me han apoyado y enseñado tantas cosas a lo largo de mi Vida. A todas esas luces que alumbran el camino cuando hay oscuridad.

Agradecimientos especiales a la UNAM, por haberme otorgado una beca para la realización de estudios de maestría. Especialmente al Dr. Bohumil Psenicka y al M.I. Omar Nieto Crisóstomo por el apoyo brindado.

Agradecimientos especiales la Universidad de Valencia, España. Por haberme otorgado una beca de investigación. Especialmente al Dr. Jesús Soret Medel por facilitar los materiales necesarios para la implantación de la red domótica.

Índice.

1. INTRODUCCION	1
2. RED DOMOTICA	3
2.1 Elementos de red	3
2.1.1 Sensores	3
2.1.2 Actuadores	3
2.1.3 Unidades de control	4
2.1.4 Pasarelas	4
2.1.5 Red	4
2.2 Clasificación de los sistemas domóticos	4
2.2.1 Tipología de un sistema	5
2.2.1.1 Sistemas centralizados	5
2.2.1.2 Sistemas descentralizados	5
2.2.1.3 Sistemas distribuidos (Híbridos)	6
2.2.2 Topología de un sistema domótico	7
2.2.3 Medios de transmisión	7
2.3 Protocolos	8
2.3.1 Tipos de protocolos	8
2.3.2 Interoperabilidad	8
2.3.3 Ejemplos de sistemas domóticas (protocolos)	9
2.3.3.1 Lonworks	9
2.3.3.2 X-10	10
2.3.3.3 Konnex	10
3. SISTEMA EIB/KNX	12
3.1 Medios de transmisión	12
3.2 Modos de configuración	13
3.3 Topología	14
3.4 Direccionamiento	16
3.4.1 Direcciones físicas	16
3.4.2 Acoplador	17
3.4.3 Direcciones de grupo	18
3.5 Formato de trama (Telegrama) KNX	19
3.6 Transmisión de trama	22
3.6.1 Control de acceso al medio	23
4. MODULO DE RECONOCIMIENTO DE VOZ	25
4.1 Reconocimiento de palabras aisladas	25
4.1.1 Etapa de entrenamiento	26
4.1.2 Etapa de reconocimiento	26
4.2 Bloque de preprocesamiento	27
4.2.1 Filtro paso bajas	27

4.2.2	Filtro preénfasis	28
4.2.3	Ruido ambiental	28
4.2.4	Detección de inicio y fin	28
4.2.5	Ventaneo de Hamming	29
4.3	Bloque de entrenamiento	29
4.3.1	Análisis Mel Cepstral	30
4.3.2	Modelos ocultos de Markov (HMM)	32
4.3.2.1	Procesos de Markov en tiempo discreto	32
4.3.2.2	Definición de los modelos ocultos de Markov	33
4.4	Bloque de reconocimiento	34
5.	INTEGRACIÓN DEL SISTEMA DOMÓTICO Y EL MOTOR DE RECONOCIMIENTO DE VOZ CON DISPOSITIVOS MÓVILES	36
5.1	Modelo cliente servidor	36
5.2	El cliente	38
5.3	El servidor	43
5.3.1	Servidor Web y base de datos	43
5.3.1.1	Sistema de administración	45
5.3.2	Motor de reconocimiento de voz	48
5.3.2.1	Entrenamiento	49
5.3.2.2	Reconocimiento	50
5.3.3	Servidor domótico	51
5.3.3.1	Instalación del servidor domótico en Windows	52
5.4	Funcionamiento del sistema	56
5.4.1	Autenticar	56
5.4.2	Entrenar	56
5.4.3	Reconocimiento	57
5.4.4	Ejecutar comando	58
6.	PRUEBAS Y CONCLUSIONES	61
	APENDICE A: PROGRAMACION DE DISPOSITIVOS INTELIGENTES	65
	GLOSARIO DE TERMINOS	78

1. INTRODUCCIÓN.

La era moderna se caracteriza por el advenimiento de innovaciones tecnológicas que marcan la pauta del crecimiento económico de las sociedades, tal es el caso de las **computadoras**, **telefonía celular** e **Internet**. En esta investigación se utilizan esas tres grandes innovaciones más las llamadas redes domóticas (redes de casas inteligentes) para realizar una aplicación útil para las personas.

El objetivo de esta investigación es desarrollar un sistema capaz de ejecutar comandos de voz dentro de un entorno automatizado, en este caso, las llamadas “casas inteligentes”. Con el propósito de añadir valor agregado a aplicaciones domóticas, introduciendo mayor integración entre las máquinas y las personas sabiendo de antemano que el lenguaje hablado es la manera más natural de comunicación de los seres humanos.

En la figura 1.1 podemos observar el modelo utilizado, este corresponde al llamado modelo cliente – servidor, que consta de un teléfono celular (cliente) y una PC (servidor), el cliente lleva una aplicación desarrollada en la plataforma Java Micro Edition (J2ME) con la cual el usuario puede enviar una señal de voz por medio de Internet al servidor, quien la procesara y reconocerá el comando a ejecutar dentro de la red domótica o el servidor.

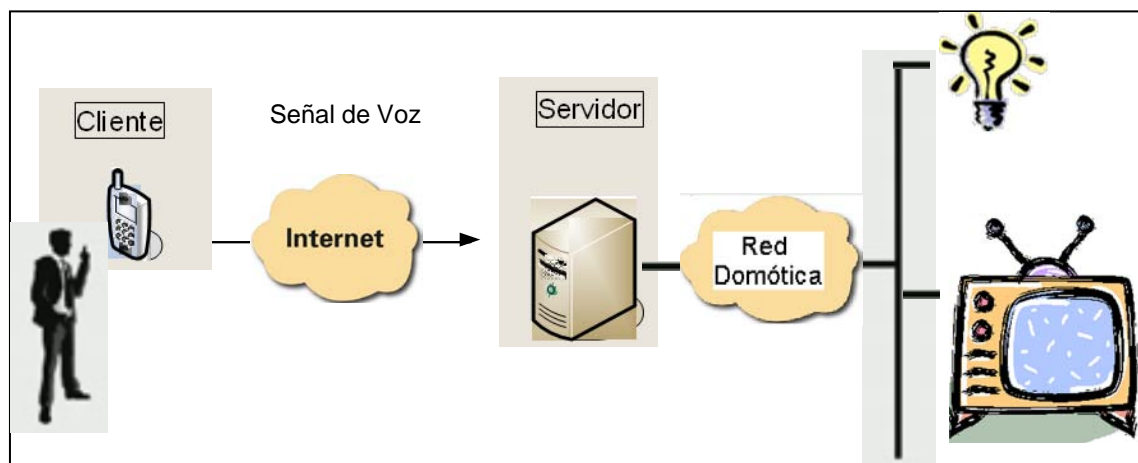


Figura 1.1 Sistema de reconocimiento de voz aplicado a redes domóticas.

La automatización de procesos actualmente es un área bastante importante, donde un Ingeniero en Telecomunicaciones puede intervenir en la entrega de información entre los dispositivos de comunicación, como son las redes de datos que intercomunican a dichos dispositivos. En esta investigación se trata de **redes domóticas** las cuales conciernen al intercambio de información entre dispositivos llamados “inteligentes”, dentro de una vivienda o edificio como pueden ser: controles de luces, calefacción, puertas, alarmas, etc. con el propósito de automatizar casas y edificios.

Se desarrolló un motor de reconocimiento de voz de palabras aisladas dependiente del locutor, el cual es capaz de recibir como entrada señales de voz utilizando algún dispositivo móvil (celular, PDA, Pocket PC, etc.), una vez capturada dicha señal se envía al servidor

quien deberá procesarla (reconocedor de voz) y disparar un proceso dentro de la casa, como puede ser prender o apagar luces, accionar alarmas, poner música, etc.

En la actualidad existen sistemas un tanto similares, la diferencia entre estos es la manera en que se obtiene la señal de voz, en algunos sistemas la voz es obtenida por medio de un micrófono de ambiente, en el sistema propuesto la señal es obtenida por medio de un dispositivo de mano (celular, PDA, Pocket PC, etc.), lo cual permite mayor movilidad del usuario y obtención un tanto clara de la señal.

Además de poder ejecutar comandos por medio de reconocimiento de voz, el sistema diseñado es capaz de disparar procesos en el servidor por medio de interacción de listas de comandos. Donde el usuario navega por plantillas de comandos y por medio de clicks elige la acción a ejecutar. Esto es un complemento que extiende en gran medida el número de acciones a ejecutar y da mayor comodidad.

2. RED DOMÓTICA.

El concepto de *domótica* tiene su origen en Francia con la palabra “domotique”, debido a la necesidad de referirse al conjunto de materiales y nuevas técnicas de automatización que podían emplearse en el hogar. Etimológicamente proviene del latín “domus” que significa *casa* y “tica” que se refiere a *automática*, es decir, “casa automatizada” [1].

De este modo entenderemos como red domótica aquel sistema de dispositivos interconectados por medio de protocolos de comunicación, quienes tienen la finalidad de realizar acciones automatizadas dentro de una vivienda. Por ejemplo encender/apagar luces, abrir/cerrar puertas, accionar alarmas, controlar la calefacción, etc.

La domótica básicamente consta de cuatro campos: confort, seguridad, ahorro de energía y comunicaciones. La aplicación desarrollada en este trabajo entra en los campos de confort y comunicación, ya que se trata de una interfaz entre el usuario y los dispositivos de red.

2.1 Elementos de red

Básicamente, una red domótica está compuesta de una serie de dispositivos que detectan un cambio de estado de una variable y transmiten esta información para que otros dispositivos puedan actuar en consecuencia, en función de las reglas o normas establecidas por el usuario del sistema. Cualquier dispositivo domótico se puede clasificar dentro de uno de los siguientes grupos: sensores, actuadores, unidades de control, pasarelas y redes [2].

2.1.1 Sensores

Son elementos físicos que sirven de entrada al sistema y realizan una función de conversión, transformando una variable física (temperatura, presión, etc.) en otra diferente que suele ser una señal eléctrica. Según su tipo se pueden clasificar en:

- Analógicos: transmiten señal continua, con valores dentro de un intervalo mínimo y máximo. Por ejemplo: sensores de temperatura, luminosidad, consumo eléctrico, etc.
- Digitales: transmiten únicamente un conjunto finito de valores, con señales ON/OFF. Por ejemplo: sensores de presencia, movimiento, ruido, presencia de gas, existencia de alguna fuga de agua, etc.

2.1.2 Actuadores

Se consideran salidas del sistema y permiten ejecutar las ordenes obtenidas mediante distintas entradas al sistema (sensores, unidades de control, etc.), actuando sobre los equipos terminales. La gran variedad de actuadores se pueden clasificar a grandes rasgos en tres tipos [2]:

- Electromecánicos: motores, electroválvulas, cerraduras eléctricas, rieles, conectores, etc.
- Acústicos: alarmas, bocinas, etc.
- Luminosos: paneles, monitores, etc.

2.1.3 Unidades de control

Es el elemento donde reside la mayor parte de la “inteligencia” del sistema. Esta encargado de recibir, procesar y tratar la información según el programa o algoritmo preestablecido, y comunicarlo, cuando proceda, a los actuadores correspondientes.

Gracias a la evolución de la electrónica, algunos captadores y actuadores han llegado a ser autónomos al incorporar la función del procesador. En definitiva es lo que caracteriza la arquitectura del sistema, como veremos mas adelante.

En función del sistema domótico la unidad de control tendrá una serie de características diferenciadas. Así, en sistemas distribuidos (como el sistema Konnex (KNX) [7]), la unidad de control se reparte en distintos elementos una vez realizada la programación de los dispositivos mediante una computadora. Existen otros sistemas cuya unidad de control es un ordenador o un sistema basado en microprocesadores que funge como servidor central del sistema (sistemas centralizados).

2.1.4 Pasarelas

Las pasarelas de comunicación están revolucionando el mercado de la domótica. La función de estos dispositivos es interconectar diferentes redes de una vivienda o edificio (control domótico, telefonía, televisión y tecnologías de la información), a una red publica de datos, como por ejemplo Internet, efectuando en su caso, la adaptación y traducción entre diferentes protocolos.

En esta investigación se utiliza un servidor domótico de distribución libre llamado “eibcontrol” [3]. Este funge como pasarela entre el llamado Internet protocol y el sistema domótico KNX.

2.1.5 Red

Es un sistema de circulación y transmisión de algún elemento, que une distintos puntos, lugares geográficos o topográficos. Por ejemplo, red eléctrica, de agua, telefónica, de comunicación, etc.

Se puede definir una red como [1]: Una interconexión de nodos (agentes, dispositivos...) que intercambien información o recursos, o también, un conjunto de elementos independientes interconectados. Para que una red funciones son necesarios toda una serie de elementos, señales y reglamentos conocidos como protocolos de comunicación.

2.2 Clasificación de los sistemas domóticos.

Los sistemas domóticos pueden clasificarse de varias formas en función de la tipología, de la topología y de los medios de transmisión.

2.2.1 Tipología de un sistema

Según la forma en que la red una los distintos puntos o lugares dispondremos de lo que se suele denominar arquitectura de control de la red. Puede ser de varios tipos:

- Sistemas centralizados.
- Sistemas descentralizados.
- Sistemas distribuidos.

2.2.1.1 Sistemas centralizados

Los sistemas centralizados son aquellos que se caracterizan por tener un único nodo que recibe toda la información de las entradas, la procesa y envía a las salidas las órdenes de acción correspondientes. (Están unidos a un nodo central que dispone las funciones de control y mando). En la figura 2.1 podemos observar un sistema centralizado.

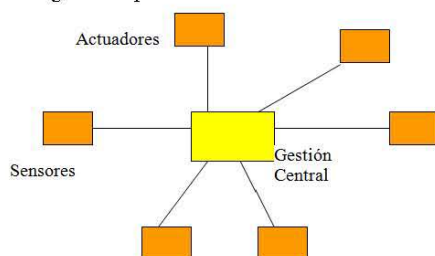


Figura 2.1. Sistema centralizado.

Las ventajas de los sistemas centralizados son:

- Los elementos sensores y actuadores son de tipo universal.
- Coste reducido o moderado.
- Fácil uso y formación.
- Instalación sencilla.

Los inconvenientes son:

- Cableado significativo.
- Sistema dependiente del funcionamiento óptimo de la central.
- Modularidad difícil.
- Reducida ampliabilidad.

2.2.1.2 Sistemas descentralizados

En los sistemas descentralizados, todos los nodos actúan de forma independiente unos de otros. Comparten la misma línea de comunicación y cada uno de ellos dispone de funciones de control y mando. En la figura 2.2 podemos observar un sistema descentralizado.

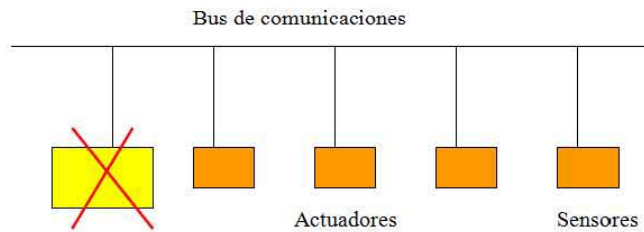


Figura 2.2. Sistema descentralizado.

Es necesario, en estos entornos, un protocolo de comunicaciones para que todos los elementos produzcan una acción coordinada.

Las ventajas de los sistemas descentralizados son los siguientes:

- Seguridad de funcionamiento en caso de que un dispositivo falle.
- Fácil de rediseñar.
- Cableado reducido.
- Fiabilidad de productos.
- Ampliable fácilmente.

Los inconvenientes son:

- Sensores no universales y limitados a la oferta.
- Coste elevado de los dispositivos.
- Mas próximos a “Edificios inteligentes” que a “casas inteligentes”.
- Complejidad de programación.

2.2.1.3 Sistemas distribuidos (Híbridos)

Los sistemas distribuidos combinan las tipologías centralizada y descentralizada. La inteligencia del sistema esta localizada en cada uno de los nodos de control y cada nodo tiene acceso físico directo a una serie limitada de elementos de red. En la figura 2.3 podemos observar un sistema distribuido.

Es necesario, al igual que en el de los sistemas descentralizados, un protocolo de comunicaciones para que todos los módulos produzcan una acción coordinada.

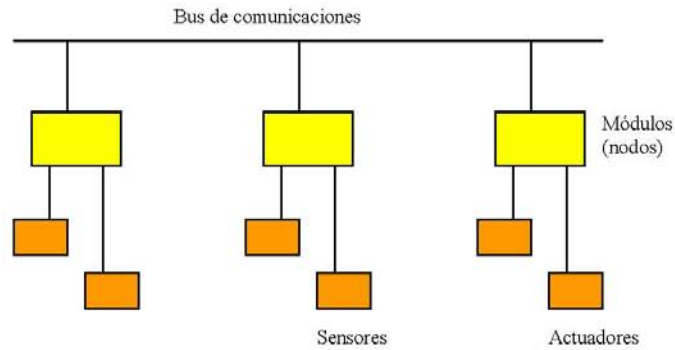


Figura 2.3. Sistema distribuido.

Las ventajas de los sistemas distribuidos son:

- Seguridad de funcionamiento.
- Posibilidad de rediseño de la red.
- Ampliable fácilmente.
- Sensores y actuadores de tipo universal (económicos y de gran oferta).
- Coste moderado.
- Cableado moderado.

Como único inconveniente destacamos el hecho de que requiere programación o configuración.

2.2.2 Topología de un sistema domótico

Otro aspecto que caracteriza a un sistema es su topología, es decir, la organización física y lógica de los "nodos" de la red.

Estrella: los dispositivos de entrada (sensores) y los de salida (actuadores) van cableados hasta la central de gestión desde donde se efectúa el tratamiento de los datos del conjunto.

Anillo: los nodos se conectan en un bucle cerrado y los datos se transmiten de nodo a nodo alrededor del bucle, siempre en la misma dirección.

Bus: todos los elementos del sistema (sensores, actuadores y nodos) están ligados sobre una línea que describe el conjunto o una parte de la red.

Malla: existen diferentes nodos que permiten el envío de los datos por distintos caminos. Cada nodo puede enviar y recibir mensajes, además de tener la capacidad de reenviar mensajes de sus vecinos.

2.2.3 Medios de transmisión

Para que los diferentes dispositivos de la red se comuniquen e intercambien información entre sí, los medios que se utilizan principalmente son:

- Sistemas que usan en todo o parte señales que se acoplan y transmiten por la instalación eléctrica de baja tensión tales como sistemas de corrientes portadoras (PLC) por sus siglas en ingles.
- Sistemas que utilizan en todo o en parte señales transmitidas por cables específicos para dicha función, tales como cables de par trenzado, paralelo, coaxial o fibra óptica.
- Sistemas que usan señales radiadas, tales como ondas de infrarrojo, radiofrecuencia o ultrasonidos.

Un sistema domótico puede combinar varios de los sistemas anteriores, debiendo cumplir los requisitos aplicables en cada parte del sistema.

2.3 Protocolos.

Se denomina protocolo de comunicación el conjunto de reglas que permiten el intercambio de información (¿Quién empieza la comunicación? ¿Cómo se comunican? ¿Con quien? ¿Con que prioridad?). E indica el formato de las tramas para la transmisión de datos entre dos dispositivos de sistemas descentralizados o distribuidos, además de las técnicas de acceso al medio para los dispositivos.

Algunas de las características que se definen en un protocolo son:

- El método de compresión de datos, si está incluido.
- El modo en que el dispositivo que envía el mensaje indica que ha terminado de enviarlo.
- El modo en el que el receptor indica que ha terminado de recibir el mensaje.
- El modo en que se comunica a los dispositivos que ha habido una colisión de tramas.

2.3.1 Tipos de protocolos

- **Privados o patentados:** protocolo desarrollado por una compañía para uso exclusivo de sus clientes; no son públicas sus características y prestaciones.
- **Abiertos:** es aquel que es conocido por todas las empresas y los usuarios disponen de la información y documentación necesaria para su implementación (por ejemplo, Ethernet, TCP/IP...). Normalmente están respaldados por grupos de empresas que se asocian para el desarrollo del protocolo.
- **Normalizados o estandarizados:** este concepto solo se utiliza para los protocolos abiertos. Cuando el grupo de interés de un determinado sistema ofrece a un organismo normalizador reconocido su protocolo y este lo estudia y decide publicar bajo una norma, se dice que un determinado protocolo está normalizado, lo que produce una garantía a los consumidores.

2.3.2 Interoperabilidad

La interoperabilidad es una propiedad, que se define como la capacidad de integrar productos de distintos fabricantes en sistemas flexibles y funcionales sin necesidad de desarrollar hardware, software o herramientas a la medida.

Productos inter operables permiten a los diseñadores de cada proyecto utilizar el mejor dispositivo para cada sistema o sub-sistema sin necesidad de utilizar una línea entera de productos de un mismo fabricante además de disminuir el costo de los dispositivos debido a mayor oferta de estos.

2.3.3 Ejemplos de sistemas domóticas (protocolos)

A continuación se describirán de manera general los principales sistemas domóticos disponibles, actualmente en el mercado que se asocian a distintos protocolos de comunicación.

Dentro de los protocolos utilizados para la automatización de casas destacan los sistemas: Lonworks, X-10 y Konnex [4].

2.3.3.1 Lonworks

La tecnología lonworks es una plataforma creada por la empresa norteamericana Echelon en el año de 1988, actualmente cuenta con mas de 4 000 fabricantes repartidos por todo el mundo. Este protocolo esta ratificado como estándar por diversos organismos de estandarización internacional como ANSI, IEEE, CEN y EN, en enero del 2009, además, esta ratificado por las normas ISO en la norma ISO/IEC 14908.1. [5]

El estándar LonWorks se basa en una arquitectura distribuida basada en un protocolo de comunicación llamado Lontalk que puede ser transmitido por cable de par trenzado y/o línea de potencia (PLC) principalmente. Las topologías en el medio cableado pueden ser bus, anillo o topología libre, la velocidad estándar de comunicación en par trenzado es de 78, 130 bps, el tipo de cable por el que transmitir esta información es variado y diverso, dependiendo de la longitud de la red empleada, ya que se pueden hacer redes de hasta 2 700 metros en bus sin repetidor.

Las bases del estándar LonWorks se basó en conceptos muy simples:

- Los sistemas de control en vivienda y edificios son fundamentalmente idénticos, independientemente de la aplicación final.
- Un sistema de control distribuido es significativamente más potente, flexible y ampliable que un sistema centralizado.
- Reconfiguración de funciones sin tocar la instalación.
- Conectividad fácil y robusta a Internet.

El estándar LonWorks proporciona una solución a los múltiples problemas de diseño, construcción, instalación y mantenimiento en viviendas y edificios, su tamaño puede variar de 1 a 32, 000 dispositivos.

Es importante que desarrollar productos basados en LonWorks no es barato, debido a los precios con que ECHELON comercializa las herramientas de desarrollo y al pago de royalties o “créditos” a pagar [4].

2.3.3.2 Protocolo X-10

X-10 es uno de los protocolos más antiguos que se están usando en aplicaciones domóticas. Fue diseñado en Escocia entre los años 1976 y 1978 con el objetivo de transmitir datos por las líneas de baja tensión a muy baja velocidad (60 bps en EEUU y 50 bps en Europa) y costes muy bajos. Al usar ondas portadoras sobre las líneas eléctricas de la vivienda no es necesario instalar nuevos cables para que se intercambien información entre diferentes dispositivos X-10.

El protocolo X-10, en si, no es propietario, es decir, cualquier fabricante puede producir dispositivos y ofrecerlos en su catálogo.

Gracias a su madurez (mas de 20 años en el mercado) y la tecnología empleada, los productos X-10 tienen un precio bastante competitivo, por lo que es líder en el mercado residencial y de pequeñas empresas (incluso realizados por usuarios finales o electricistas sin conocimientos de automatización).

Existen tres tipos de dispositivos X-10; los que solo pueden transmitir órdenes, los que solo pueden recibirlas y los que pueden enviar/recibir estas.

Los transmisores pueden direccionar hasta 256 receptores. En una misma instalación puede haber varios receptores configurados con la misma dirección, todos los realizaran la función preasignada cuando un transmisor envíe una trama con esta dirección.

Los dispositivos bidireccionales, tienen la capacidad de responder y confirmar la correcta realización de una orden, lo cual puede ser muy útil cuando el sistema X-10 esta conectado a una PC que muestre el estado actual de la red domótica.

2.3.3.3 Konnex

Este protocolo fue elegido para la implantación de la red domótica utilizada en esta investigación y en la siguiente sección explicaremos en detalle este protocolo. En la tabla 2.1 podemos observar la comparación de los tres protocolos mas utilizados.

Por mencionar otros protocolos existentes aplicados a redes domóticas tenemos: BUSing, Domonet, SimonVOX.2, CEBus, etc. y protocolos inalámbricos que son usados en automatización de viviendas tenemos: ZigBee, Wifi, HomeRF, Bluetooth, etc.

Tecnología	Medios de transmisión	Velocidad de transmisión (depende del medio)	Distancia máxima al dispositivo (depende del medio)
Konnex	- Par trenzado - Ondas portadoras - Ethernet - Radio	- 9 600 [bps] - 1 200/2 400 [bps] - 2.4 [Kbps]	- 1 000 [m] - 600 [m]
Lonworks	- Par trenzado - Ondas portadoras - Ethernet - Radio	- 78 [Kbps] – 1.28 [Mbps] - 2. 54 [Kbps]	- 500 – 2 700 [m]

	- Coaxial - Fibra óptica		
X-10	- Ondas portadoras	- 60 [bps] en EU - 50 [bps] en Europa	- 185 [m ²]

Tabla 2.1. Protocolos más utilizados en casas y edificios inteligentes.

3. SISTEMA EIB/KNX.

El estándar Konnex (KNX) es la iniciativa de tres asociaciones europeas BCI (Francia): promocionaba el sistema Batibus; EIB Asociación (Bélgica): promocionaba el sistema EIB y European Home Systems Association (Holanda): promocionaba el sistema EHS. Al unirse estas tres nace la asociación KNX con sede en Bruselas, Bélgica. Dado que el sistema EIB es compatible con el KNX la mayor parte de los dispositivos llevan ambos logos y en la literatura existente se puede hacer referencia a estos como el mismo sistema.

El objetivo de esta asociación es crear un único estándar europeo que sea capaz de competir en calidad, prestaciones y precios con otros sistemas norteamericanos como Lonworks o CEBus.

A finales de 2003 el estándar KNX fue aprobado por el CENELEC (Comité Europeo de Normalización Electrotécnica) como norma europea (EN 50090) para domótica e inmótica. También el CEN ha aprobado el estándar KNX, en concreto bajo los números EN 13321-1 (Medios + Protocolo) y EN 13321-2 (KNXnet/IP). A finales de 2006 se aprobó el KNX como norma internacional bajo el número de norma ISO/IEC 14543-3. En 2007 la tecnología KNX alcanzó también el status de norma experimental en China (GB/Z 20965).

El estándar KNX se ha convertido en el primer sistema normalizado a nivel mundial para la comunicación de control de viviendas y edificios que es totalmente libre de royalties para los miembros de la asociación y es totalmente independiente de cualquier plataforma tecnológica específica de hardware / software.

La asociación KNX se contaba en sus orígenes con 9 miembros: desde entonces el número de miembros ha aumentado hasta sobrepasar los 125 (a mediados de 2008), incluyendo empresas que anteriormente no pertenecían a ninguna de las asociaciones existentes. Dichas empresas representan más del 80% del mercado europeo de las instalaciones y los electrodomésticos. Puede consultarse en todo momento la lista actualizada de miembros visitando www.knx.org.

La tecnología KNX es un sistema abierto, destinado a la aplicación de soluciones compatibles, flexibles y rentables. Su uso es apropiado tanto para instalaciones simples y limitadas como para el sector de edificio completo.

3.1 Medios de transmisión

Gracias a la flexibilidad de la tecnología KNX, cualquier instalación puede ser fácilmente adaptable a las necesidades cambiantes del usuario.

Los medios soportados en el sistema son: par trenzado (TP1 y TP2), ondas portadoras sobre la red eléctrica en dos frecuencias (PL100 y PL132), radio frecuencia (RF) y Ethernet ("KNX IP"). Asimismo, utilizando las pasarelas correspondientes, es posible la transmisión de telegramas KNX a otros medios, por ejemplo fibra óptica. El medio soportado debe estar visible en la etiqueta de cada producto. En la tabla 3.1 podemos observar distintas áreas de aplicación de los medios de transmisión más utilizados.

Medio	Transmisión vía	Áreas preferidas de aplicación
Par trenzado	Bus de control independiente	Nuevas instalaciones y grandes renovaciones – nivel máximo de fiabilidad de la transmisión
Power line	Red existente	En lugares donde no se necesita un cable de control adicional y hay disponible cable de 230 V
Radio Frecuencia	Radio	En lugares donde no se desea o no se puede instalar cableado.

Tabla 3.1. Áreas de aplicación de los distintos medios de transmisión.

3.2 Modos de configuración

El estándar actual contempla dos tipos de configuración, según lo que indique la etiqueta de cada producto, el mismo puede configurarse (Uniones lógicas y parámetros) vía:

1. **Modo de instalación System (S-Mode):** la planificación de la instalación, así como su configuración es realizada por profesionales a través de una PC con una aplicación software especialmente diseñada para este fin.
Este modo es idóneo para proyectistas e instaladores KNX certificados y, sobre todo, para grandes instalaciones. La asociación KNX vende el software llamado ETS en distintas versiones, con el cual se realizan la mayoría de las configuraciones de los dispositivos.
2. **Modo de instalación sencilla (E-Mode):** los dispositivos son programados en fábrica para realizar la función concreta. Aún así deben ser configurados algunos detalles de la instalación. La configuración no se realiza mediante PC, sino a través de un controlador central o mediante micro interruptores alojados en el mismo dispositivo.
Este modo esta pensado para el instalador con conocimientos básicos sobre la tecnología Bus. Los productos compatibles con E-Mode tienen habitualmente una funcionalidad limitada y están concebidos para instalaciones de tamaño medio.
3. **Modo de instalación Automático (A-Mode):** cada dispositivo tiene la habilidad de programación dinámica. Con la filosofía plug & play ni el instalador ni el usuario tienen que configurar el dispositivo. Este modo esta especialmente indicado para ser usado en electrodomésticos, equipos de entretenimiento (consolas, set-top boxes, etc.) y proveedores de servicios.

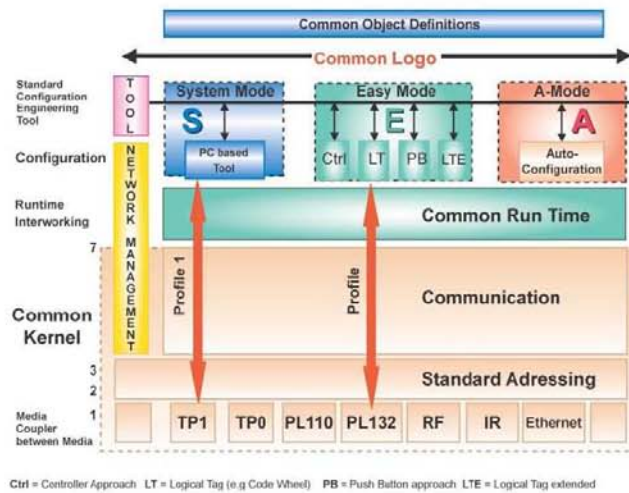


Figura 3.1. Modelo KNX [8].

En la figura 3.1 podemos observar la arquitectura del estándar KNX. Independientemente del modo de configuración elegido, cada componente de bus dispone de su propia inteligencia, por lo que no resulta necesaria una unidad central de control (por ejemplo una PC) que sirva como servidor central. Por lo tanto, el KNX puede ser utilizado tanto para pequeñas instalaciones (viviendas) como en proyectos mucho más grandes (hoteles, edificios, etc.).

3.3 Topología

La **línea** es la unidad mínima de instalación (ver figura 3.2). En ella se pueden conectar hasta 64 dispositivos (dependiendo de la capacidad de la fuente de alimentación y de la carga máxima producida por los dispositivos existentes).

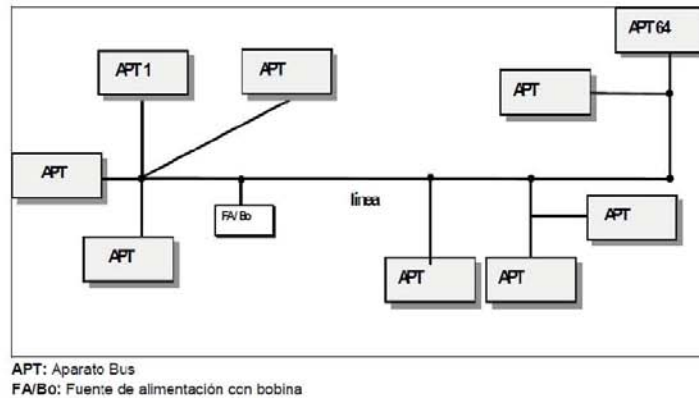


Figura 3.2 Topología – línea [6].

Si se quieren conectar más componentes al bus, se deberá instalar una nueva línea, que se acoplará, junto con la primera, a una línea principal mediante un acoplador de línea. Se pueden agregar hasta 15 líneas en la línea principal, construyendo un área (ver figura 3.3). De este modo en un área se pueden conectar hasta 960 dispositivos.

Cada línea, tanto principal como secundaria, debe tener su propia fuente de alimentación. Además, la línea principal puede tener conectados directamente hasta 64 dispositivos (incluyendo acopladores de línea), el número máximo de componentes Bus en la línea principal disminuye con el número de acopladores de línea en uso.

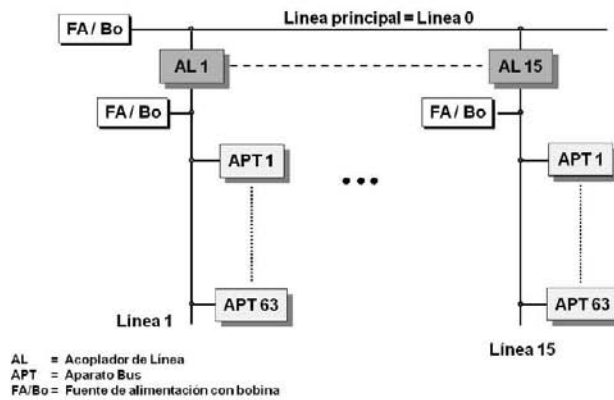


Figura 3.3. Topología – Área [6].

Cabe la posibilidad de unir hasta un total de 15 áreas distintas mediante los denominados *Acopladores de Área* (AA), para constituir el sistema completo (figura 3.4), que permitiría integrar hasta un máximo de 14 400 dispositivos.

También es posible situar componentes Bus en la línea de áreas. El número máximo de componentes Bus en la línea de áreas disminuye con el número de acopladores de áreas en uso.

Dividiendo la instalación KNX en líneas y áreas, la funcionalidad del sistema se incrementa considerablemente.

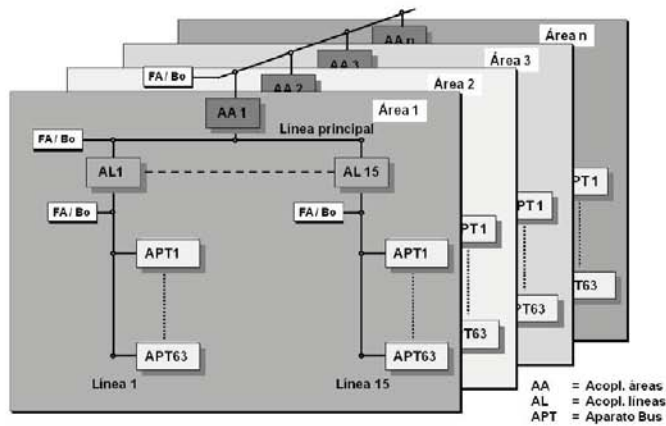


Figura 3.4. Topología – Varias áreas [6].

Para efectuar la conexión de dispositivos al bus, en cada línea se permite cualquier topología: árbol, estrella, bus o anillo, lo que facilita la instalación en viviendas y edificios. Únicamente no se permite cerrar los anillos entre líneas situadas topológicamente situadas en diferentes subredes. Es decir, no está permitido que un dispositivo este conectado a dos líneas a la vez.

3.4 Direccionamiento

Cada elemento de la instalación KNX queda perfectamente identificado, gracias al esquema de direccionamiento. Existen dos tipos de direcciones: direcciones físicas y direcciones de grupo.

3.4.1 Direcciones físicas

Las direcciones físicas identifican unívocamente cada dispositivo y corresponde con su localización en la topología global del sistema (área – línea secundaria- dispositivo). La dirección física consta de tres campos que en formato digital se representan separados por puntos análogamente con las direcciones IP:

- Área (4 bits). Identifica una de las 15 áreas.
A=1-15 señala las áreas funcionales 1-15.
A=0 corresponde a la dirección de la línea de áreas del sistema.
- Línea (4 bits). Identifica cada una de las 15 líneas en cada área.
L=1-15 señala las líneas 1-15 en las áreas definidas por A.
L=0 se reserva para identificar a la línea principal dentro del área.
- Dispositivo (8 bits). Identifica cada uno de los posibles dispositivos dentro de una línea.
C=1-255 señala los componentes de Bus en la línea definida por L.
C=0 se reserva para el acoplador de línea.

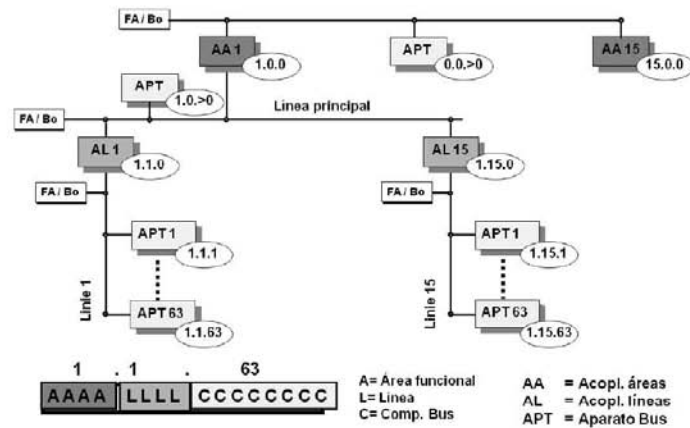


Figura 3.5. Direcciones físicas [6].

En la Figura 3.5 se muestra un ejemplo de direcciones físicas asignadas a los dispositivos de un sistema EIB. Tras la puesta en marcha, la dirección física se sigue utilizando para los siguientes fines:

- Diagnósticos, detección de errores, modificación de la instalación mediante reprogramación.
- Direccionamiento de objetos interfaces por medio de herramientas de puesta en marcha u otros dispositivos.

Importante: en el funcionamiento normal de la instalación, la dirección física no tiene ningún significado.

En la línea de áreas se conectan hasta 15 acopladores de área (AA), cuyas direcciones irán desde 1.0.0 hasta 15.0.0. Esta línea puede tener conectados dispositivos de Bus con direcciones 0.0.>0.

Cada área tiene una línea principal, con su fuente de alimentación, a la que se conectan los acopladores de línea (AL), con direcciones 1.1.0 a 1.15.0, y a cada línea secundaria conectada a un acoplador de línea pueden conectarse hasta 64 dispositivos.

3.4.2 Acoplador

Para la interconexión de diferentes líneas y diferentes áreas se emplea la **unidad de acoplamiento**. Este elemento es el mismo para los diferentes tipos de conexión, y dependiendo de la dirección física que se le asigne actuará como acoplador de línea, acoplador de área, o incluso repetidor dentro de una misma línea.

En el caso del acoplador de línea o de área, la unidad de acoplamiento actúa como encaminador (*router*). Cuando se asignan los parámetros, se proporciona al acoplador una tabla de filtros. Todos los telegramas de grupo recibidos se reenviarán si están registrados en la tabla de filtros. En la figura 3.6 podemos observar la función que realiza el acoplador.

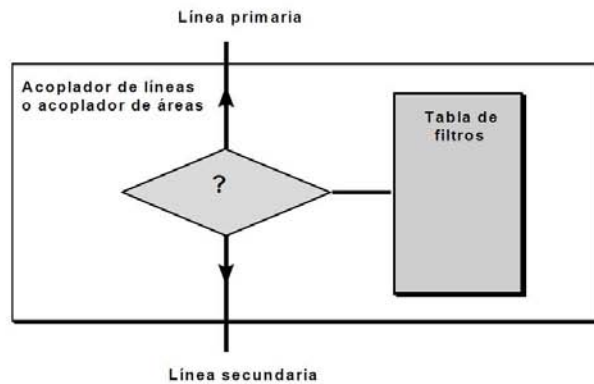


Figura 3.6. Acoplador [6].

De esta forma, cada línea funciona de forma independiente. Sólo se dejarán pasar a través del acoplador los telegramas de cruce de líneas y de cruce de áreas. Los LEDs amarillos del acoplador parpadean cuando se recibe un telegrama proveniente de las líneas de cada lado del acoplador.

Un acoplador también puede ser utilizado como amplificador (o repetidor) de línea para ampliar con un segmento mas de 64 componentes de Bus y una longitud de cable adicional de 1 [Km], en este caso, deja pasar todos los telegramas; no tiene tabla de filtros.

El acoplador de área, el acoplador de línea y el amplificador de línea son el mismo aparato. Las tareas que realizan dependen de dónde estén situados y de la correspondiente dirección física asignada (ver figura 3.7).

			El acoplador se utiliza	
			como	para
> 0	= 0	= 0	Acoplador de áreas	Línea de áreas/Línea principal
> 0	> 0	= 0	Acoplador de líneas	Línea principal/Línea secundaria
> 0	> 0	> 0	Amplificador de línea	Expansión de una línea

Figura 3.7. Modos de configuración del acoplador [2].

La dirección de un acoplador de Bus desprogramado es 15.15.255.

3.4.3 Direcciones de grupo

La comunicación entre los dispositivos en una instalación se lleva a cabo por medio de las direcciones de grupo. Las direcciones de grupo sirven para definir funciones específicas del sistema y son los que determinan las asociaciones de dispositivos en funcionamiento (y la comunicación entre sus objetos de aplicación).

Las direcciones de grupo asignan la correspondencia entre los elementos de entrada del sistema (sensores) y elementos de salida (actuadores).

Existen dos tipos de direccionamiento de grupo: de dos (grupo principal / subgrupo) y tres (grupo principal / grupo intermedio / subgrupo) niveles, dependiendo de las necesidades en la jerarquización de las funciones del sistema (ver figura 3.8).

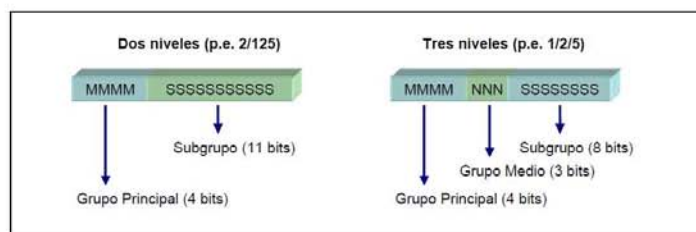


Figura 3.8. Niveles en las direcciones de grupo [2].

El encargado de diseñar el proyecto puede decidir cómo se usarán los niveles basándose, por ejemplo, en este esquema:

- Grupo Principal = planta
- Grupo intermedio = función (Ej.: iluminación, calefacción,...)
- Subgrupo = función de un componente o grupo de componentes (Ej.: conmutar lámpara cocina, conmutar luz ventana dormitorio, conmutar techo salón, regular techo salón...)

En el campo principal se pueden asignar valores de 1 a 13, los valores 14 y 15 no deben emplearse, ya que no se filtran por los acopladores TP1 y podrían afectar el funcionamiento del sistema [2]. En todos los campos la dirección "0" esta reservada para funciones propias del sistema.

Las direcciones de grupo, que asocian sensores con actuadores, se pueden asignar a cualquier dispositivo en cualquier línea (son independientes de las direcciones físicas), con las siguientes condiciones:

- El número de direcciones de grupo que pueden ser asignadas a los sensores o actuadores es variable y depende del tamaño de la memoria.
- Varios actuadores pueden tener la misma dirección de grupo, es decir, responden a un mismo telegrama.
- Los actuadores pueden responder a mas de una dirección de grupo (pueden estar direccionados o asociados a varios sensores simultáneamente).

3.5 Formato de trama (Telegrama) KNX

La trama KNX contiene siete campos, seis de control para conseguir una transmisión fiable de la información y un campo de datos que contiene el comando a ejecutar. En la figura 3.9 se muestra los campos y las longitudes de estos:

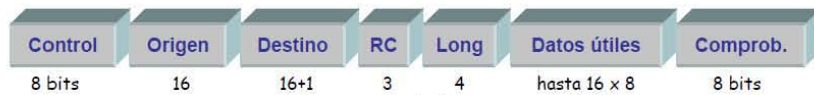


Figura 3.9. Formato de la trama KNX [2].

Campo de Control. En este campo se indica la prioridad de dicho telegrama según la función (alarma, servicios del sistema o servicios habituales). El bit de repetición (bit 3) se pone a cero en caso de repetirse algún envío a causa del no reconocimiento de alguno de los destinatarios. De este modo se evita que los dispositivos que ya hayan ejecutado la orden la vuelvan a repetir (ver figura 3.10).



Figura 3.10. Campo de control [2].

Campo de dirección origen: En este campo va contenida la dirección física (área / línea / dispositivo) de modo que se conozca el emisor del telegrama en las tareas de mantenimiento.

Campo de dirección destino: La dirección de destino puede ser de dos tipos, en función del valor que tome el bit de mayor peso de este campo (bit 17). Si tiene el valor “0”, se trata de una dirección física, y la trama se dirige únicamente a un dispositivo. Si tiene valor “1”, se trata de una dirección de grupo, y la trama se dirige a todos los dispositivos que tengan dicha dirección de grupo.

Campo RC (Contador de ruta): la trama transmitida por el aparato emisor contiene un contador de ruta cuyo valor inicial es de 6 (RC=6).

Cada acoplador disminuye en uno el contador de ruta y retransmite la trama siempre y cuando el valor sea mayor que 0. Esto para evitar inundaciones en la red de tramas erróneas que por alguna razón no puedan ser entregadas antes que pasen 6 acopladores.

El contador de ruta puede tener el valor de 7 que significa que los acopladores no modificaran el valor al pasar por ellos, en este caso se ignora la tabla de filtros y se pasa la trama a todos los acopladores de línea de todo el sistema.

Campo de longitud: en los cuatro bits de este campo se indica cuantos bytes contiene el campo de datos (0= 1 byte, 15=16 bytes).

Campo de datos útiles: Este campo lleva la información a ser entregada al dispositivo, esta se divide en dos partes el tipo de comando y los datos.

la transmisión y el dispositivo que recibe envía un acuse de recibo (ACK). De lo contrario se envía una trama de no reconocimiento (NAK) para que el emisor repita el envío. Si el dispositivo esta ocupado envía un código *Busy* para que el emisor reenvíe la trama. Ver figura 3.12.

D7	D6	D5	D4	D3	D2	D1	D0	
N	N	0	0	B	B	0	0	
1	1	0	0	0	0	0	0	BUSY (ocupado)
0	0	0	0	1	1	0	0	NAK (Recepción incorrecta)
1	1	0	0	1	1	0	0	ACK (recepción correcta)

Figura 3.12. Tipos de respuestas de reconocimiento [2].

3.6 Transmisión de trama

Los estados lógicos que puede tener un bit son "1" y "0". Durante el estado lógico "1" no circula corriente y durante el "0" circula corriente, como se muestra en la figura 3.13.

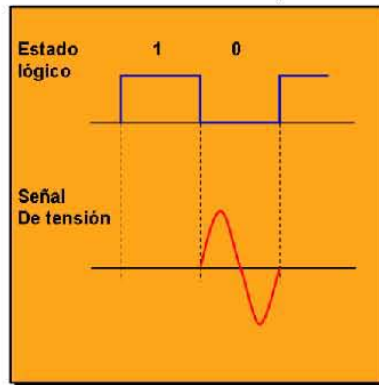


Figura 3.13. Estados lógicos [6].

Un telegrama se genera cuando se produce un evento en el bus, por ejemplo la activación de un interruptor o la detección de presencia. En este caso el componente buscara enviar un telegrama al bus.

En la figura 3.14 podemos observar el mecanismo de acceso al medio. El transmisor comprueba que el bus este desocupado por lo menos un tiempo t_1 y envía la trama, al finalizar la transmisión espera un intervalo de tiempo t_2 , la recepción de un acuse de recibo (ACK).



Figura 3.14. Envío de telegrama (trama) [6].

Si la recepción es incorrecta, es decir, no se recibe acuse de recibo (o bien se recibe NAK, no reconocimiento), la transmisión se reintenta hasta tres veces. Todos los dispositivos direccionados envían el ACK simultáneamente.

3.6.1 Control de acceso al medio

El dispositivo que desee enviar un telegrama puede iniciar la transmisión inmediatamente si encuentra el bus desocupado el bus.

Si dos dispositivos desean transmitir al mismo tiempo, se producirá una colisión de tramas, esta se resuelve con el protocolo de acceso al medio llamado por sus siglas en Inglés CSMA/CA (Acceso Múltiple por detección de portadora, evitando colisiones). Como ya vimos la codificación se realiza de modo que el estado lógico "0" es dominante (presencia de corriente) sobre el estado "1" (sin corriente).

El mecanismo de control de acceso al medio funciona de la siguiente forma:

- El dispositivo escucha el bus, si esta libre transmite. De lo contrario espera un tiempo aleatorio y reintenta.
- Durante el envío cada transmisor escucha los datos presentes en el bus comparándolos en todo momento con los que ha transmitido.
- Si no se producen colisiones el envío se completa sin contratiempos.
- Tan pronto como un transmisor con el estado lógico "1" detecta el estado lógico "0" (circulación de corriente en la línea), entonces hay colisión, y detiene la transmisión para dar paso al componente con mayor prioridad.
- El dispositivo con prioridad más baja (quien envía el "1") se mantiene a la escucha de la red para esperar el final de la transmisión y después buscara transmitir sus datos.

De esta forma, si hay varios dispositivos intentando transmitir a la vez, el procedimiento CSMA/CA asegura que sólo uno de esos componentes pueda ocupar el bus. Por tanto, no se reduce la capacidad de transmisión de datos.

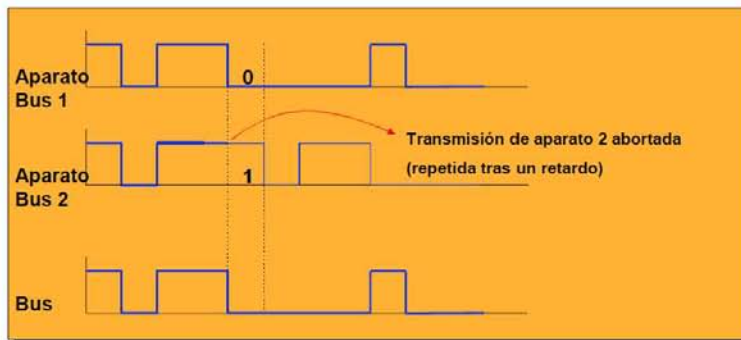


Figura 3.15. Resolución de colisión de tramas [6].

BIBLIOGRAFIA CAPITULO 1-3

- [1] Asociación Española de Domótica (CEDOM) “Cuaderno de divulgación Domótica”, Segunda Edición, AENOR, 2008.
- [2] I. R. Matías, C. Fernández, “Telecomunicaciones en la construcción”, Universidad Publica de Navarra, 2005.
- [3] <http://eibcontrol.sourceforge.net/>
- [4] “El hogar digital”, Segunda Edición, Colegio Oficial y Asociación Española de Ingenieros Técnicos de Telecomunicación, 2005.
- [5] <http://www.echelon.com/communities/developers/lonworks/>
- [6] Asociación KNX, “Curso básico KNX”, 2009.
- [7] <http://www.knx.org/>
- [8] Asociación Konnex, “System-architecture”, 2004, <http://www.knx.org/es/descargas-soporte/descargas/>

4. MODULO DE RECONOCIMIENTO DE VOZ.

En este capítulo analizaremos el procedimiento empleado para la realización del sistema de reconocimiento de voz.

El medio mas natural de comunicación entre los humanos es por medio de la voz, por esta razón es importante realizar interfaces entre los humanos y las máquinas que faciliten la interacción entre estos. Cada día la miniaturización de los dispositivos de mano como los teléfonos celulares es mayor, debido a esto, resulta más fácil y cómodo utilizarlos como interfaz para capturar, procesar la voz y reducir al mínimo el uso del teclado además de permitir mayor movilidad del usuario.

En la actualidad existe una gran tendencia a la automatización de procesos haciendo mas fácil la realización de tareas, en este caso nos concierne la automatización de casas inteligentes, en las cuales se pueden automatizar procesos como encendido y apagado de luces, abrir puertas, recursos multimedia, etc. mediante comandos verbales.

En esta tesis se ligó el trabajo realizado por el M. I. Crisóstomo Nieto Omar titulado **“Diseño de un reconocedor de comandos de voz para el DSP TMS320C6711”** y el trabajo realizado por los Ingenieros Rodríguez Román Lizbeth Andrea, Castañeda Trujillo Eduardo Daniel y Lovera Granados Fabián titulado **“Comunicación entre un dispositivo móvil y una PC usando comandos verbales”**. Del primero se utilizo la etapa de caracterización de la señal de voz mediante los métodos de análisis Mel Cepstral y Modelos ocultos de Markov y del segundo la interfaz cliente servidor entre el dispositivo móvil y la PC (servidor) además de la etapa de preprocesamiento del sistema de reconocimiento de voz.

Existen dos principales tipos de reconocimiento de voz: reconocedores de voz continúa y reconocedores de palabras aisladas a su vez estos pueden ser dependientes e independientes del locutor. Realizamos un motor de reconocimiento de palabras aisladas dependiente del locutor.

4.1 Reconocimiento de palabras aisladas.

El objetivo de esta tesis es poder ejecutar comandos de voz (prender luz, encender calefacción, etc.), por lo tanto es suficiente realizar el reconocimiento de palabras aisladas, el cual consta básicamente de una etapa de entrenamiento y otra de reconocimiento. En el entrenamiento se toma un número de repeticiones de la misma palabra (prender luz, encender calefacción, etc.), de las cuales por medio de algoritmos programados en el servidor se obtienen características de dicha palabra y se almacenan para posteriormente ser comparadas, con los comandos a ejecutar por medio del bloque de reconocimiento. En la etapa de reconocimiento se obtienen las características de una sola señal, se comparan con las almacenadas previamente en la etapa de entrenamiento para decidir que palabra se asemeja más de todas las almacenadas previamente.

4.1.1 Etapa de entrenamiento.

En el entrenamiento se obtienen características acústicas de un número de señales de audio en este caso 7, las cuales contienen la misma palabra a caracterizar (prender luz, encender calefacción, etc.), una vez obtenidas dichas características son almacenadas en una base de datos. En la figura 4.1 se muestra de manera general el proceso de entrenamiento.

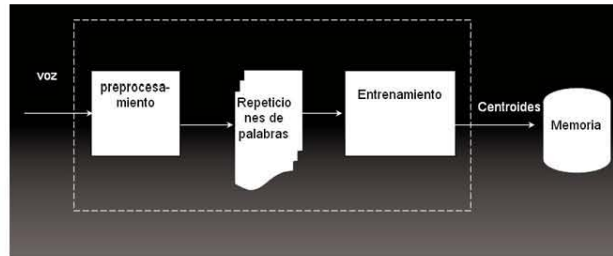


Figura 4.1. Bloque de entrenamiento.

El entrenamiento consta de una etapa de preprocesamiento y un bloque de entrenamiento. Una vez obtenida la señal de voz entra a un bloque de preprocesamiento en el cual se acondiciona la señal mediante algoritmos de filtrado, detección de inicio y fin de palabra y ventaneo en tramas, para después entrar en el bloque de entrenamiento donde se extraen las características acústicas que representan al comando, esto se realiza una vez que se completen 7 repeticiones. En el bloque de entrenamiento se calculan los coeficientes llamados Mel Cepstral y se utilizan modelos ocultos de Markov (HMM) para obtener el conjunto de características de la palabra a entrenar. A la salida se tiene una cantidad de vectores quienes contienen información acústica que representa a la palabra dicha.

4.1.2 Etapa de reconocimiento.

La etapa de reconocimiento utiliza el mismo bloque de preprocesamiento para acondicionar la señal antes de entrar al bloque de reconocimiento en el cual se calculan los coeficientes Mel Cepstral y mediante el algoritmo llamado Forward se comparan con las almacenadas previamente en la etapa de entrenamiento y se determina la palabra que mas se asemeja entre las almacenadas previamente a la palabra pronunciada. En la figura 4.2 podemos observar el proceso de reconocimiento.

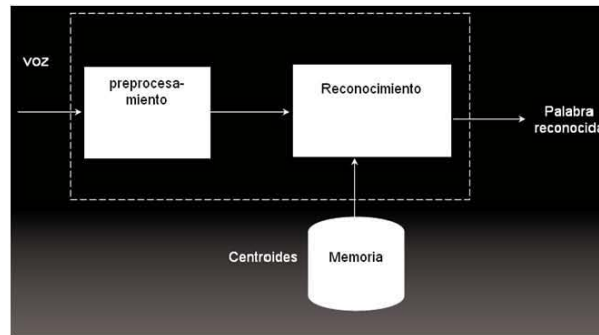


Figura 4.2. Bloque de reconocimiento.

4.2 Bloque de preprocesamiento.

El objetivo de este bloque es acondicionar la señal antes de entrar a la etapa de procesamiento ya sea para entrenar o para reconocer una palabra. En este bloque se realizan los siguientes procesos: una vez obtenida la conversión de analógico a digital se filtra paso bajas para limitar en frecuencia, se aplica un filtro preénfasis el cual realza las altas frecuencias, se calcula umbrales de ruido para después restarlos y así eliminar hasta cierto punto el ruido ambiental, se recorta la señal detectando el inicio y fin de la palabra y por último se aplica una ventana de Hamming por cada 128 muestras con la finalidad de hacer mas eficiente el procesamiento en frecuencia (no modificar el contenido espectral de la señal al tomar tramas). Enseguida veremos cada etapa en detalle. En la figura 4.3 podemos observar el esquema general del bloque de preprocesamiento.

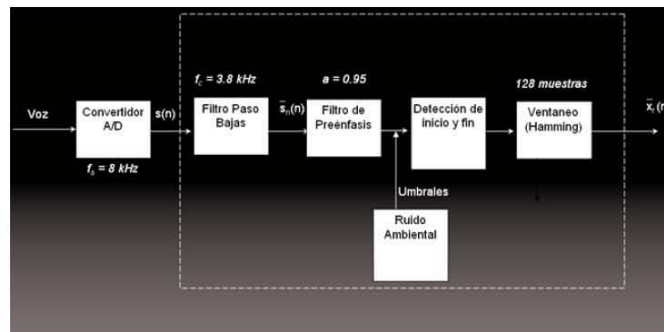


Figura 4.3. Bloque de preprocesamiento.

4.2.1 Filtro paso bajas.

Las señales de voz frecuentemente hablando se encuentran concentradas en la parte baja del espectro de frecuencias en el intervalo de 3 [Hz] hasta 4 [kHz] aproximadamente, por lo tanto podemos filtrar la señal paso bajas con una frecuencia de corte en este caso de 3.8 [kHz]. Un intervalo de 0 a 3.8 [kHz] es suficiente para dejar pasar la información contenida en las señales de voz, todo lo que este fuera de este intervalo es considerado ruido y por lo tanto es eliminado.

Otra razón por la cual es necesario filtrar paso bajas es para limitar en banda al señal y poder realizar correctamente el procesamiento de la señal, concretamente poder calcular la transformada discreta de Fourier [5].

4.2.2 Filtro preénfasis.

El espectro de sonidos llamados sonoros (los de mayor frecuencia en la señal de voz) presentan una caída de -6 [dB/octava], mientras se incrementa la frecuencia. Esta es una combinación de una caída de -12 [dB/octava] debido a la fuente de extracción de la voz y una ganancia de +6 [dB/octava] debida a la radiación producida por la boca [6]. Es decir que cada que se duplica la frecuencia, la amplitud de la señal es reducida en un factor de 16. Por esta razón es necesario compensar esa pérdida dándole una ganancia de +6 [dB/octava] en el intervalo de frecuencias indicado, para que el espectro medio tenga un intervalo dinámico similar en toda la banda de frecuencias. Esto se logra con el filtrado llamado preénfasis.

Debido a que los sonidos sordos se concentran en la parte alta del espectro de frecuencias de voz, el filtrado preénfasis se puede realizar por medio de un filtro digital paso altas. Este filtrado se puede lograr mediante la ecuación en diferencias:

$$y[n] = x[n] - ax[n-1] \dots \dots \dots (4.1)$$

Donde $y[n]$ es la muestra actual de la salida del filtro preénfasis, $x[n]$ es la muestra actual de la señal de voz, $x[n-1]$ es la muestra anterior de la señal de voz, a es una constante usualmente entre 0.9 y 1, el valor de esta no es crítico [6].

4.2.3 Ruido ambiental.

Para caracterizar el ruido en el ambiente se graba 10 milisegundos de sonido a los cuales se calcula el valor medio, este valor es restado a toda la señal para poder contrarrestar hasta cierto punto el ruido ambiental.

Existen otras técnicas de eliminación de ruido, como las de eliminación espectral que consiste en calcular el espectro del ruido y el de la señal de voz con ruido, para esto es necesario 2 micrófonos, una vez obtenidos los espectros son restados y de esta manera se cancela el ruido. Esta técnica esta fuera del alcance de esta tesis.

4.2.4 Detección de inicio y fin.

Al grabar la señal de voz se toman 3 segundos, lo ideal seria tener programado un detector de silencios y solo obtener muestras de voz, debido a que no se tiene es necesario realizar un procesamiento de la señal para recortarla y quedarnos con solo muestras de voz. Para esto utilizamos métodos en el dominio del tiempo.

Algunas de las ventajas que obtenemos al detectar el inicio y fin de la palabra son:

- Procesar menor número de información.
- Comparar únicamente los patrones de información.
- Evitar confusiones a causa de del ruido o señales de fondo.

Algunos de los problemas que se presentan en la detección son:

- Señales espurias que se pueden confundir con la señal.
- Silencios contenidos dentro de las palabras que tienen fonemas plosivos (ej. /t/, /p/, /k/) que pueden confundirse con un falso principio o fin.
- Los fonemas fricativos (ej. /f/, /th/, /h/, etc.), ya que tienen baja energía.
- Sonidos cortos (ej. /t/, /p/, /k/).
- Detección de fonemas nasales al final de la palabra (baja energía y cruces por cero)
- Respiraciones del locutor, que pueden confundirse por su duración.
- Los micrófonos tienen resonancia después de pronunciar una palabra (sobre todo en vocales).
- Los niveles de ruido pueden confundirse con la señal de voz.

Creo inconveniente explicar los algoritmos que se utilizan para la detección de inicio y fin de la palabra, simplemente diremos que se utilizan las funciones de cruces por cero y magnitud promedio (energía en tiempo corto). Para más información véase [6], [7].

4.2.5 Ventaneo de Hamming

Debido a que el procesamiento de la señal se hace por tramas, para aumentar la velocidad de procesamiento en el cálculo de la transformada de Fourier y otros procesos es necesario aplicar un ventaneo de Hamming por cada 128 muestras de la señal de voz. Se utiliza la ventana de Hamming en lugar de una rectangular debido a que aquella modifica en menor grado el contenido espectral de la señal, para más información véase [6], [7].

4.3 Bloque de entrenamiento.

Una vez preparadas las señales de voz mediante el bloque de preprocesamiento estas entran al bloque de entrenamiento, para caracterizarlas mediante la extracción de características acústicas por medio de los coeficientes Mel-Cepstral y algoritmos de clasificación de patrones llamados modelos ocultos de Markov. En esta investigación se utilizó el trabajo realizado por el M.I. Nieto Crisóstomo Omar, "Diseño de un reconocedor de comandos de voz para el DSP TMS320C6711".

El entrenamiento consiste básicamente de dos bloques, análisis, Mel Cepstral donde se calculan los coeficientes llamados Mel Cepstral que contienen las características acústicas de las señales, y un segundo bloque donde se modelan las características acústicas mediante los llamados Modelos ocultos de Markov (HMM). En la figura 4.4 podemos observar el diagrama general del proceso de entrenamiento.

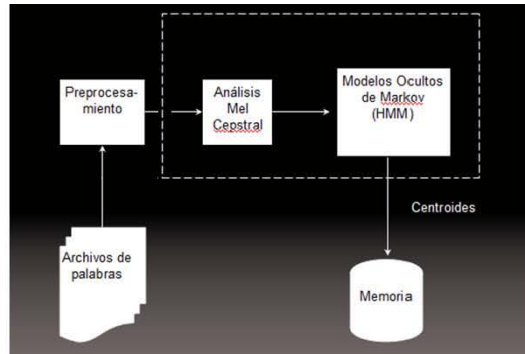


Figura 4.4. Método de entrenamiento.

4.3.1 Análisis Mel Cepstral (MFCC)

El objetivo de este bloque es calcular una secuencia de vectores que representen las características de la señal en una forma compacta. Para esto se utiliza el método conocido con el nombre de Mel-Frequency Cepstral Coefficients (MFCC) [9], el cual es un método para calcular los coeficientes Cepstral y referirlos a una escala en frecuencia no lineal llamada Mel.

Los coeficientes Mel-Cepstral se pueden calcular mediante la transformada de Fourier, análisis de banco de filtros y la transformada coseno [9]. En la figura 4.5 podemos observar de manera general el procedimiento para calcular los MFCC's.

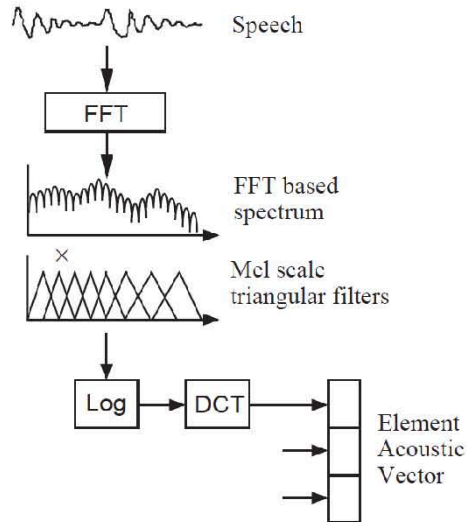


Figura 4.5. Cálculo de los vectores Mel-Cepstral [9].

La principal suposición del análisis Cepstral es que la señal de voz puede considerarse como un proceso estacionario, (características espectrales son relativamente constantes)

alrededor de un periodo de tiempo pequeño (algunos milisegundos). Entonces, es necesario dividir la señal de voz en pequeños bloques comúnmente de 10 a 25 [ms], en este caso se escogió bloques de 128 muestras que equivalen a 16 [ms] ya que la señal esta muestreada a 8 [kHz]. Este paso se realizó en el bloque de preprocesamiento, además de aplicarle un ventaneo de Hamming para no afectar el contenido espectral de la señal.

Los pasos necesarios para el cálculo de los coeficientes son los siguientes [2]:

- 1) Dividir la señal de voz en tramas de 10 a 25 [ms].
- 2) Para cada trama se calcula la transformada de Fourier a cada trama de la señal mediante la siguiente ecuación:

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \dots\dots\dots(4.2)$$

Para $k = 0, 1, 2, \dots, N - 1$

Donde:

$x[n]$ es la muestra en el instante presente de la señal de entrada.
 N es el número de muestras contenidas en una trama, en este caso 128.

- 3) Se calcula la potencia espectral de $X[k]$ mediante la siguiente expresión:

$$S_a[k] = [\text{real}(X[k])]^2 + [\text{imag}(X[k])]^2 \dots\dots\dots(4.3)$$

- 4) Filtrar por medio de banco de filtros. la respuestas en frecuencia del filtro es la siguiente:

$$H_m[k] = \begin{cases} 0 & k < f[m - 1] \\ \frac{(k - f[m - 1])}{(f[m] - f[m - 1])} & f[m - 1] \leq k < f[m] \\ \frac{(f[m + 1] - k)}{(f[m + 1] - f[m])} & f[m] \leq k \leq f[m + 1] \\ 0 & k > f[m + 1] \end{cases} \dots\dots\dots(4.4)$$

Donde:

$m = 0, 1, 2, \dots, M - 1$.
 $M =$ numero de filtros.

En la siguiente figura podemos observar la respuesta en frecuencia de dicho filtro, común mente el número de filtros es de 24 a 40:

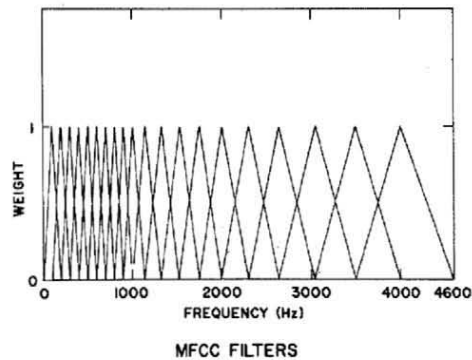


Figura 4.6. Respuesta en frecuencia del banco de filtros.

- 5) Ahora calculamos la energía logarítmica en la salida de cada filtro mediante la siguiente expresión:

$$S[m] = \ln \left[\sum_{k=0}^{N-1} S_a[k] H_m[k] \right] \dots\dots\dots (4.5)$$

Para $m = 0, 1, 2, \dots, M - 1$

- 6) Por último se calculan los coeficientes por medio de la transformada Coseno Discreta a la salida de los M filtros mediante la siguiente expresión:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\pi m \left(m + \frac{1}{2}\right) / M\right) \dots\dots\dots (4.6)$$

Para $m = 0, 1, 2, \dots, M - 1$

Para el reconocimiento de voz típicamente se utilizan los primeros 13 coeficientes. Entonces, al final de este método tenemos para cada trama 13 coeficientes que caracterizan a cada una de ellas.

4.3.2 Modelos ocultos de Markov (HMM)

Un modelo oculto de Markov es un modelo estadístico, en el que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos (u *ocultos*, de ahí el nombre) de dicha cadena a partir de los parámetros observables.

4.3.2.1 Procesos de Markov en tiempo discreto.

Los procesos de Markov son procesos estocásticos, es decir, estudian el comportamiento de variables aleatorias a lo largo del tiempo. En este caso estudiaremos

los procesos de Markov en tiempo discreto los cuales toman valores de tiempo discreto y los estados posibles de la variable aleatoria son igualmente discretos.

Formalmente, un proceso de Markov es un proceso estocástico tal que para $n = 1, 2, \dots$ y para cualquier sucesión posible de estados s_1, s_2, \dots, s_{n+1} , tenemos:

$$P(X_{n+1} = s_{n+1} | X_1 = s_1, X_2 = s_2, \dots, X_n = s_n) = P(X_{n+1} = s_{n+1} | X_n = s_n) \dots \dots (4.7)$$

Esto significa que la probabilidad de que la variable aleatoria X para el tiempo $n+1$ sea igual al estado s_{n+1} , conocidos los estados anteriores dependa solo del instante previo. En la figura 4.7 podemos observar un diagrama que ejemplifica un proceso de Markov con 5 estados de transición.

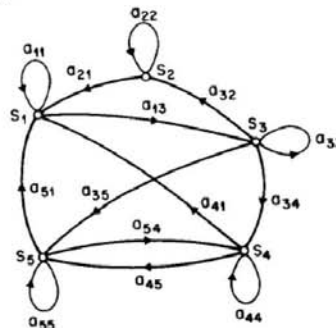


Figura 4.7. Proceso de Markov con 5 estados [2].

Los coeficientes de transición de estado están dados por:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad \text{para } 1 \leq i, j \leq N \dots \dots (4.8)$$

Y tienen las siguientes propiedades:

$$a_{ij} \geq 0 \quad \forall j, i$$

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i$$

4.3.2.2 Definición de los modelos ocultos de Markov.

En los procesos de Markov, cada estado corresponde a un evento observable determinístico, es decir, la salida de la fuente en cualquier estado dado, no es aleatorio. En el caso de los modelos ocultos de Markov la salida de la fuente en cualquier estado es una función probabilística del estado.

Un modelo oculto de Markov es básicamente un proceso de Markov, donde la salida es una variable aleatoria X , generada acorde a la salida de la función de probabilidad asociada a cada estado, ver figura 4.8.

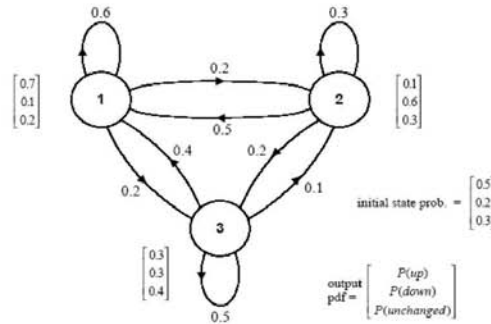


Figura 4.8. Modelo de Markov de tres estados [2].

Un modelo oculto esta caracterizado por lo siguiente [10]:

1. $\mathbf{O} = \{o_1, o_2, \dots, o_M\}$, alfabeto de salidas observadas. Donde M es el número de distintos símbolos de observación por estado.
2. $\mathbf{\Omega} = \{s_1, s_2, \dots, s_N\}$, conjunto de estados que representa el espacio de estados. Donde N es el número de estados en el modelo.
3. $\mathbf{A} = \{a_{ij}\}$, matriz de probabilidades de transición, donde a_{ij} es la probabilidad de realizar una transición del estado i al estado j .

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad \text{para } 1 \leq i, j \leq N \quad \dots\dots(4.9)$$

4. $\mathbf{B} = \{b_j(k)\}$, matriz de distribución de probabilidades de observación de los símbolos, donde $b_j(k)$ es la probabilidad de observar el símbolo o_k , estando en el estado j

$$b_j(k) = P(o_k \text{ en } t | q_t = s_j) \quad \text{para } 1 \leq j \leq N, \quad 1 \leq k \leq M \quad \dots\dots\dots(4.10)$$

5. $\pi = \{\pi_i\}$, distribución de probabilidad del estado inicial, donde:

$$\pi_i = P(q_i = s_i) \quad \text{para } 1 \leq i \leq N \quad \dots\dots\dots(4.11)$$

Por ultimo paso se modelan las palabras de entrenamiento tomando en cuenta su composición fonética. Cada uno de los fonemas es representado por un modelo oculto de Markov de tres estados. El modelo completo de cada palabra se obtiene concatenando los modelos HMM de cada uno de los fonemas que conforman la palabra. Y se aplica el método *forward-backward* con el algoritmo *Baum-Welch* descrito en [2] para finalizar el entrenamiento.

4.4 Bloque de reconocimiento.

El reconocimiento emplea la probabilidad $P(X|W)$, la cual se lee como la probabilidad que la palabra dicha sea X dado que tenemos W. Para el cálculo de dicha probabilidad se utiliza el algoritmo conocido como *forward* descrito en [2]. En la figura 4.9 vemos el proceso de reconocimiento, en el cual una vez obtenidos los coeficientes MFCC se

calculan las probabilidades para cada una de las palabras entrenadas previamente y la probabilidad mayor será en este caso la palabra reconocida.

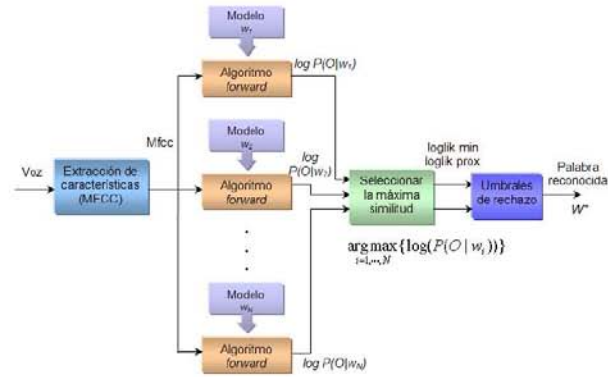


Figura 4.9. Proceso de reconocimiento [2].

BIBLIOGRAFIA CAPITULO 4

- [1] Lizbeth Rodríguez, Eduardo Castañeda, Fabián Lovera. “*Comunicación entre un dispositivo móvil y una PC usando comandos verbales*”. Facultad de Ingeniería, UNAM. 2008.
- [2] Omar Nieto Crisóstomo. “*Diseño de un reconocedor de comandos de voz para el DSP TMS320C6711*”. Facultad de Ingeniería, UNAM. 2008.
- [3] Huang Xuedong, Acero, Hon. “*Spoken Language Processing A Guide to Theory, Algorithm and System Development*”. Prentice Hall 2001.
- [4] Steve Young. “*Large Vocabulary Continuous Speech Recognition: a Review*”. Cambridge University Engineering Department 1996.
- [5] A. V. Oppenheim and R. W. Schaffer, “**Discrete-Time Signal Processing**”, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [6] Herrera Abel. “**Notas, tareas y prácticas del curso procesamiento de voz**”. Facultad de Ingeniería, UNAM.
- [7] Lawrence Rabiner, B. Hwang Juang, “*Fundamentals of Speech Recognition*”, Ed. Prentice Hall, 1993.
- [8] J. G. Proakis, J. R. Deller Jr, J. H. L. Hansen “*Discrete-Time Processing of Speech Signals*”, Ed. Prentice Hall, 1987.
- [9] Steve Young, “*Large Vocabulary Continuous Speech Recognition: a Review*”, Cambridge University Engineering Department, 1996.
- [10] Rabiner, L.R., “*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*”, Proc. of IEEE, 1989, 77(2), pp. 257-286.

5. INTEGRACIÓN DEL SISTEMA DOMÓTICO Y EL MOTOR DE RECONOCIMIENTO DE VOZ CON DISPOSITIVOS MÓVILES.

De manera general la aplicación desarrollada en esta investigación consta de tres partes: dispositivo móvil, motor de reconocimiento de voz y red domótica. Para integrar dichas partes utilizamos el modelo cliente-servidor que explicaremos en la siguiente sección.

El dispositivo móvil lleva una aplicación desarrollada en la plataforma Java Micro Edition (J2ME), capaz de obtener el comando de voz además de poder seleccionar comandos de una lista y mandarlos por Internet al servidor, donde se encuentra alojado el sistema de reconocimiento de voz y la pasarela a la red domótica (servidor domótico). Una vez que el cliente confirma el comando reconocido el servidor manda a ejecutar la acción deseada en la red domótica (encendido/apagado de luces, subir/bajar persianas, etc.).

5.1 Modelo cliente servidor.

El modelo cliente servidor de manera practica lo podemos definir como, “un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o mas procesos que cooperan entre si”. Usualmente la mayoría del trabajo pesado se hace en el llamado **servidor** y el **cliente** es encargado de la interacción entre el usuario y el servidor (aunque esto puede variar).

Los componentes del esquema cliente servidor son 3: el cliente, el (los) servidores y la infraestructura de comunicación, ver figura 5.1.

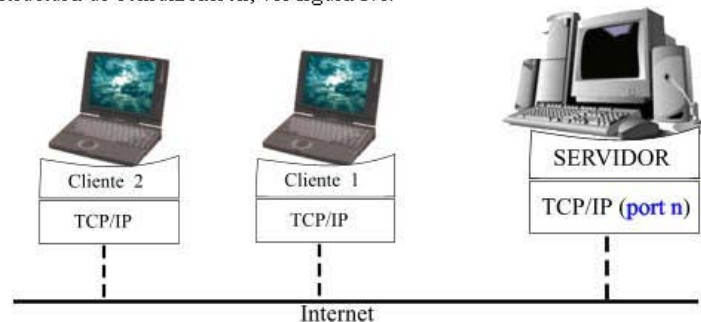


Figura 5.1. Modelo cliente servidor.

Los **Cientes** interactúan con el usuario. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar información, recibir datos (respuesta del servidor), manejar las fallas y realizar actividades de sincronización y de seguridad.

Los **Servidores** proporcionan un servicio al cliente y devuelven los resultados. En

algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para procesar la información dada por el cliente, recibir su respuesta y enviarla al cliente. Por las razones anteriores la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Además deben manejar servicios como administración de la red, mensajes de control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de bases de datos.

Para que los clientes y los servidores puedan comunicarse se requiere una **infraestructura de comunicaciones**, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor actuales se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

En nuestro caso el cliente es un celular con una aplicación desarrollada por nosotros en Java Micro Edition que sirve como interfaz entre el usuario y los servidores, como infraestructura de comunicación utilizamos Internet mediante el protocolo de comunicación TCP/IP. Los servidores tanto Web, domótico y la base de datos se encuentran en el PC.

Seleccionamos el modelo cliente servidor debido a la cantidad de procesamiento utilizada en el reconocimiento de voz además de que los clientes con solo cubrir características mínimas (J2ME, con perfil MIDP 2.0) pueden tener acceso a la aplicación. A continuación veremos en detalle tanto al cliente como el servidor.

5.2 El cliente.

El cliente se trata de un dispositivo móvil en el cual corre la versión Java Micro Edition (J2ME). En la figura 5.2 podemos ver donde esta ubicado J2ME con respecto a las demás versiones de la plataforma Java de Sun Microsystems.

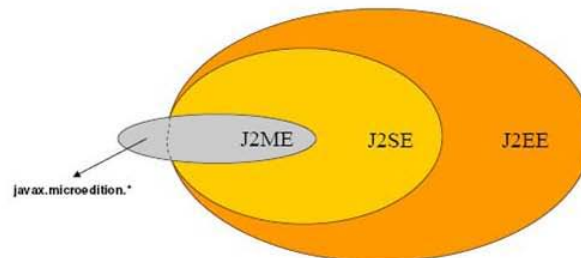


Figura 5.2. Estructura de Java 2. [1].

Seleccionamos J2ME ya que es un lenguaje multiplataforma independiente del sistema operativo que contenga el dispositivo, así es posible que la aplicación corra en múltiples dispositivos que contengan el llamado perfil MIDP 2.0, lo que quiere decir que podremos grabar la señal de voz y enviarla por Internet por medio de paquetes de

programación contenidos en dicho perfil. En la figura 5.3 podemos observar la arquitectura de la plataforma J2ME y un ejemplo de algún dispositivo.

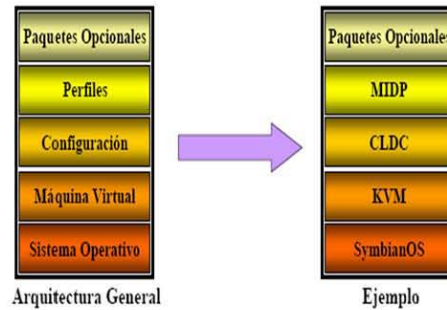


Figura 5.3. Arquitectura de la plataforma J2ME. [1]

En general la arquitectura consta de 5 capas las cuales están detalladas en [1], mencionaremos de manera general la función de cada una de estas:

- **Sistema operativo:** es propio de cada dispositivo y se encarga de ejecutar a nivel software las instrucciones dadas por el usuario. Ejemplos son Symbian, Android y propietarios de cada fabricante.
- **Máquina virtual:** es la encargada de interpretar código intermedio (bytecode) de los programas Java precompilados, a código máquina ejecutable por el sistema operativo. Destacan la Kilobyte Virtual Machine (KVM) y la compact Virtual Machine (CVM).
- **Configuración:** es un entorno de ejecución básico de J2ME, su objetivo es adecuarse a las necesidades de una familia de dispositivos con capacidades similares. Cualquier configuración está formada por tres elementos: una máquina virtual, un código nativo para realizar la interfaz entre Java y el sistema operativo y un conjunto de clases Java que constituyen el entorno de ejecución. Destacan la configuración para dispositivos conectados (CDC) y la configuración para dispositivos de conexión limitada (CLDC).
- **Perfil:** es un grupo más específico de paquetes de programación desde el punto de vista del dispositivo, el perfil añade funciones adicionales a las proporcionadas por la configuración. El más popular es el perfil MIDlet profile (MIDP) por ser el primero en llegar a una implementación.
- **Paquetes opcionales:** estos son como su nombre lo indica paquetes que se agregan aparte de los ya contenidos en el perfil y la configuración y es responsabilidad de cada fabricante.

Lo que nos importa de un dispositivo es que contenga J2ME y el perfil MIDP 2.0 el cual está especificado en Java community process como JSR 118 (MIDP 2.0) [2]. Esto quiere decir, que podemos utilizar los paquetes mostrados en la tabla 5.1:

API	Descripción
javax.microedition.rms	Manejo de base de datos.
javax.microedition.media javax.microedition.media.control	Capturar y reproducir audio.
javax.microedition.io.HttpConnection	Envío de información por Internet.

Tabla 5.1. Paquetes J2ME utilizados.

Utilizamos el editor Java llamado Eclipse [3] el cual se puede ligar al Wireless Toolkit de Sun Microsystems para realizar las simulaciones y crear los archivos que se instalaran en el móvil, ver figura 5.4 y figura 5.5.

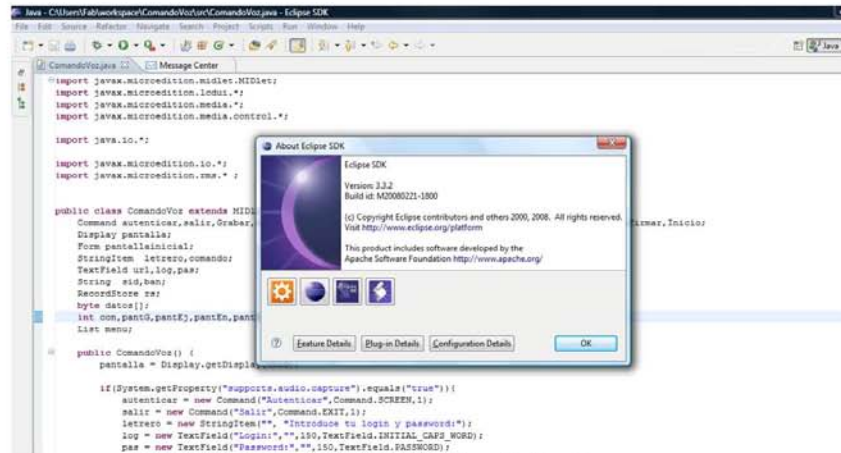


Figura 5.4. Editor Java (eclipse).

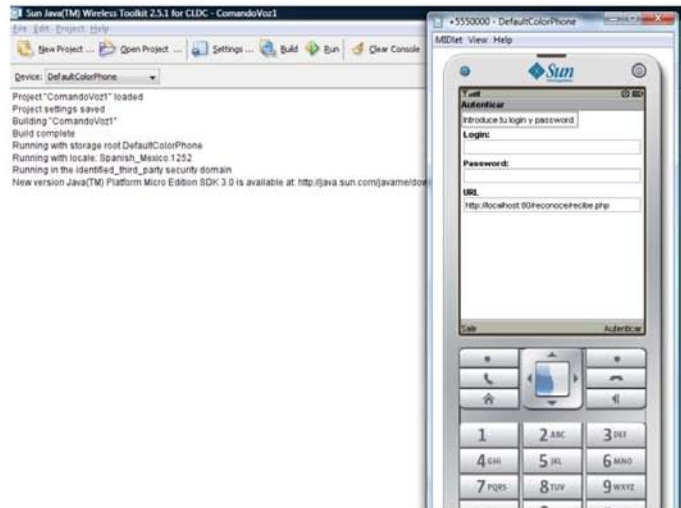
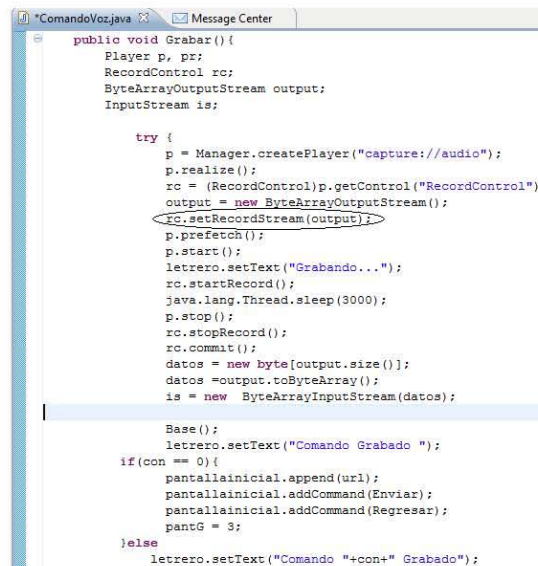


Figura 5.5. Simulador Wireless Toolkit.

Con J2ME programamos la interfaz del cliente los paquetes más importantes son `javax.microedition.media`, `javax.microedition.media.control` [4]. Mediante los cuales podemos grabar la señal de voz con el método **`setRecordStream`**, en la figura 5.6 podemos observar la función `Grabar()`. El formato en el que se graba la señal es una compresión llamada adaptive multi rate (amr).



```
public void Grabar(){
    Player p, pr;
    RecordControl rc;
    ByteArrayOutputStream output;
    InputStream is;

    try {
        p = Manager.createPlayer("capture://audio");
        p.realize();
        rc = (RecordControl)p.getControl("RecordControl");
        output = new ByteArrayOutputStream();
        rc.setRecordStream(output);
        p.prefetch();
        p.start();
        letrero.setText("Grabando...");
        rc.startRecord();
        java.lang.Thread.sleep(3000);
        p.stop();
        rc.stopRecord();
        rc.commit();
        datos = new byte[output.size()];
        datos = output.toByteArray();
        is = new ByteArrayInputStream(datos);

        Base();
        letrero.setText("Comando Grabado ");
        if(con == 0){
            pantallaInicial.append(url);
            pantallaInicial.addCommand(Enviar);
            pantallaInicial.addCommand(Regresar);
            pantG = 3;
        }else
            letrero.setText("Comando "+con+" Grabado");
    }
}
```

Figura 5.6. Función Grabar.

Una vez obtenida la señal de voz utilizamos el paquete `javax.microedition.io` y el método `write` para enviar al servidor la señal por Internet.

```

ComandoVoz.java Message Center
public String Conectar(String dato,String urla,byte[] dat) throws IOException {
    OutputStream os;
    InputStream is;
    StringBuffer sb = new StringBuffer();
    HttpURLConnection con;
    int i, longi;

    try {
        con = (HttpURLConnection)Connector.open( urla );
        con.setRequestMethod( HttpURLConnection.POST );
        con.setRequestProperty( "IF-Modified-Since", "05 Nov 2002 07:17:19 GMT" );
        con.setRequestProperty( "User-Agent", "Profile/MIDP-2.0 Configuration/CLDC-1.1" );
        con.setRequestProperty( "Content-Language", "es-ES" );
        con.setRequestProperty( "Content-Type", "application/x-www-form-urlencoded" );

        os = con.openOutputStream();
        os.write( dato.toString().getBytes() ); //mandamos el contenido de la variable dato

        if( dat != null ){
            longi=dat.length;
            os.write( ("dat=").getBytes() ); //mandamos solo un string
            for(i=0; i<longi; i++){
                os.write((" "+dat[i]).getBytes()); //mandamos el contenido de la variable dat
            }
        }

        is = con.openDataInputStream();
        while( (i=is.read()) != -1 ){
            sb.append( (char)i ); //recuperamos lo que nos envia el servidor como tipo char
        }
        return sb.toString(); //retornamos lo que contesta el servidor
    } catch (IOException e) {
        e.printStackTrace();
        return "Error";
    }
}

```

Figura 5.7. Función conectar.

Mediante la función conectar podemos enviar y recibir información entre el servidor y el cliente, los argumentos necesarios para esto son los datos a enviar y la URL del servidor al que queremos conectar. Los datos se envían por medio del método Post contenido en el HTTP protocol, ver figura 5.7.

Una vez terminado el programa se construye el proyecto y se crean los archivos que se instalaran en el móvil estos son con extensiones .jar y .jad esto se puede ver en la siguiente figura.

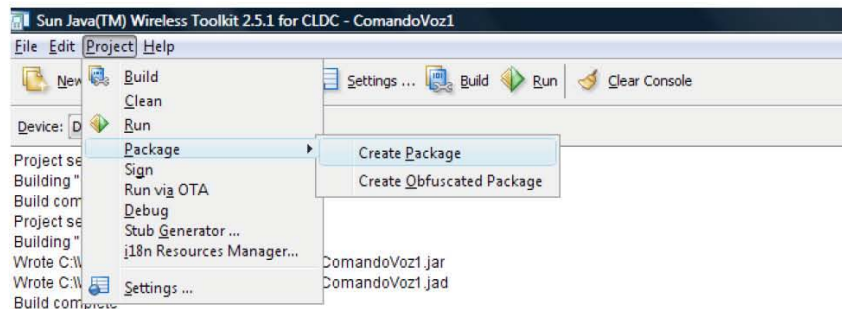


Figura 5.8. Creación de archivos .jar y .jad

Si quieren aprender a programar en J2ME en [5] se detalla completamente como realizar interfaces de usuario y muchas otras cosas mas, esto fuera del alcance de esta tesis.

5.3 El servidor.

En el servidor se ejecuta el procesamiento de la señal de voz y sirve como pasarela a la red doméstica. Los componentes de este son: un servidor Web (Xampp), motor de reconocimiento de voz y un servidor doméstico (Linux EIB Home Server) además de una base de datos utilizada para la gestión y administración del sistema.

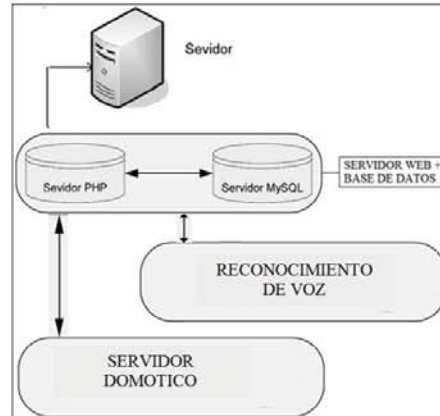


Figura 5.9. Arquitectura del servidor.

5.3.1 Servidor Web y base de datos.

El software aquí utilizado es un paquete de aplicaciones multiplataforma y de distribución gratuita llamado *xampp* [7]. Este paquete principalmente consta de una base de datos (MySQL), un servidor Web (Apache) e interpretes para lenguajes script (PHP y Perl). Una vez instalado e iniciado tecleamos <http://localhost> en el explorador y tendremos que ver la siguiente figura. Con esto ya tenemos la base de datos que utilizaremos para la autenticación del usuario y la gestión del sistema, además del lenguaje de programación PHP con el cual podremos manejar archivos y ligar el cliente con el motor de reconocimiento de voz y la red doméstica.

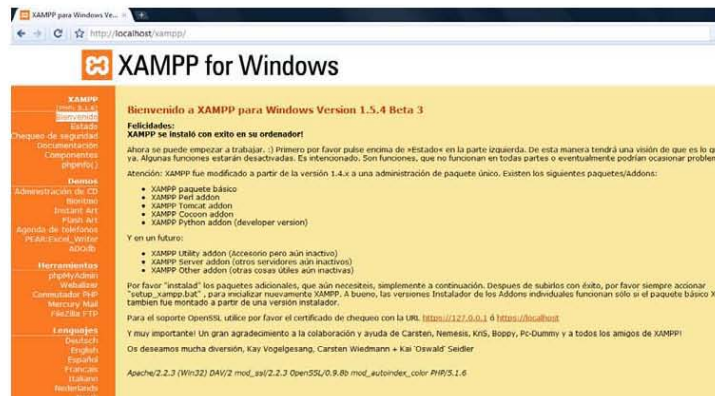


Figura 5.10. Paquete xampp.

Utilizamos un servidor Web para el intercambio de información entre el dispositivo móvil, el motor de reconocimiento de voz y la red domótica. Esto debido a la facilidad que presenta al tener funciones para esto, lo cual nos ahorra realizar una aplicación en lenguajes como Java o C++ que implemente los llamados sockets para establecer comunicación por medio de Internet.

Con el servidor Web utilizamos el lenguaje de programación llamado Hypertext Preprocessor (PHP) [6] este es un lenguaje de programación de “código abierto” altamente utilizado para realizar scripts dentro de paginas HTML y dotar de funciones extra a dichas paginas. Escribimos la pagina *recibe.php* de la cual podemos ver un fragmento en la figura 5.11.

```

recibe - WordPad
File Edit View Insert Format Help
Counter New 10 Western
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
<?php
session_start(); //inicia sesion
$time =microtime(true); //inicia cronometro
$domn=$HTTP_HOST_VARS["dom"]; //variable de configuracion del sistema
if(isset($_SESSION['user'])) { // verifica si tiene sesion abierta
    $dire = getcwd()."\\".$_SESSION['user'];
    chdir($dire); //cambia de directorio

    if($domf < 4){
        $dnl=$HTTP_HOST_VARS["dat"]; //recupera la señal de voz
        $archivo=fopen("audio.txt", "w");
        if ($archivo) {
            fputs($archivo, $dir); //escribe le señal de voz txt
        }
        fclose ($archivo);
    }

    if ($domf == 0) { //entonces reconoce
        $nomb="start audio.bat"; //reconocer palabra
        system($nomb); //ejecuta el reconocedor de voz
        $ar=fopen("nom_dom.txt", "r");
        $com = fread ($ar, 100); //recupera la palabra reconocida
        fclose ($ar);
    }
}

```

Figura 5.11. Pagina recibe.php

La función de la pagina recibe.php es obtener los datos enviados por el cliente pasarlos al sistema de reconocimiento de voz, a la base de datos o a la red domótica. En primera instancia verifica que el usuario tenga una sesión abierta si no es así le pedirá que se autentique para abrir esta, mediante la variable \$conf podemos saber que acción desea realizar el usuario ya sea entrenar, reconocer, ejecutar un comando o autenticar.

La base de datos la utilizamos para autenticar a los usuarios y guardar la lista de comandos a ejecutarse por medio de clicks.

5.3.1.1 Sistema de administración

El sistema de administración es utilizado para dar de alta nuevos usuarios, introducir la ruta a archivos a ejecutar dentro de la PC, direcciones URL o comandos domóticos a ejecutar cuando es confirmado alguno de estos.

Teclamos la dirección del servidor <http://127.0.0.1/reconoce/>, veremos la siguiente imagen.



Figura 5.12. Sistema de administración página 1.

En primera instancia debemos registrar nuestro login y password en el sistema, para esto damos click en “**crear registro**” llenamos los campos y damos enviar.



Figura 5.13. Sistema de administración página 2.

Si todo va bien veremos la siguiente imagen:



Figura 5.14. Sistema de administración página 3.

El siguiente paso es introducir la dirección IP del servidor domótico, damos click en “Cambiar IP Server Domotico” introducimos la IP y damos click en “Cambiar” y después en “Regresar”.



Figura 5.15. Sistema de administración página 4.

Si todo va bien veremos la nueva IP en la pantalla.



Figura 5.16. Sistema de administración página 5.

Con esto ya podemos agregar comandos de voz y configurar la lista de comandos ejecutados por click. En el caso de lista de comandos mediante el administrador de la base de datos que viene con el xampp, podemos dar de alta nuevos comandos y asignarles las rutas seleccionando en el menú Herramientas—phpMyAdmin.



Figura 5.17. Servidor Web Xampp.

Seleccionamos la base de datos “tesis”, la tabla “comandos”, si ya hay grabados damos click en “Examinar” de lo contrario en “Insertar”.

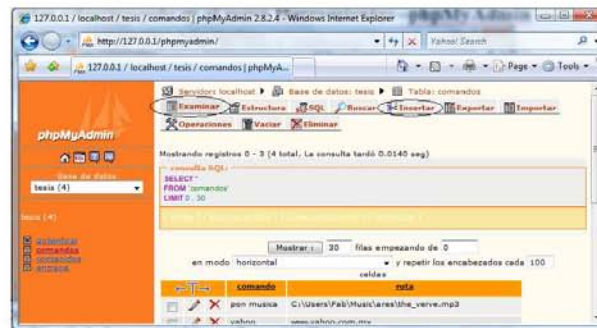


Figura 5.18. Base de datos.

En la siguiente imagen podemos ver el ejemplo de 4 comandos y sus respectivas rutas. Recordemos que en la aplicación desarrollada podemos ejecutar un archivo en la PC, direccionar a una dirección URL o introducir un comando Domótico.

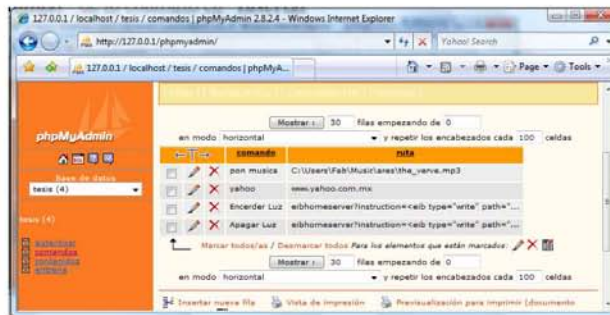


Figura 5.19. Comandos guardados.

1. El comando **pon música** esta ligado a un archivo .mp3 al seleccionar este comando se reproducirá el archivo especificado, de igual manera podemos disparar archivos .exe, .bat, etc. para ejecutar acciones dentro del PC.
2. El comando **yahoo** direcciona el navegador Web hacia la página www.yahoo.com.mx. Esta opción se puede ocupar para enviar datos a una URL por medio del método GET del protocolo TCP/IP de la forma **URL?variable1=valor1&variable2=valor2** o simplemente para visualizar paginas Web en nuestro ordenador, como consultar correo, clima, noticias, etc.
3. los comandos **Encender Luz** y **Apagar Luz** son ejemplos de comandos Domóticos. La sintaxis de estos es de la siguiente manera:

```
eibhomeserver?instruction=<eib type="write" path="/eib/groups/1/0/6/curvalue" data="1"/>
```

Para el comando **Encender Luz** mandamos a la red domótica un comando tipo "write" con el valor de "1" a la dirección de grupo 1/0/6, en el caso **Apagar Luz** será del mismo tipo y con el valor "data=0".

En el caso de comandos de voz contamos con una interfaz grafica realizada en Matlab donde podemos asignarle a cada comando ya sea una ruta a algún archivo, una URL o un comando Domótico.

Una vez entrenado algunos comandos de voz por medio del cliente podemos acceder a la página <http://127.0.0.1/reconoce/index.html>. Introducir nuestro login y password y Damos click en "**Ver Comandos Guardados**" y abrirá la siguiente interfaz:

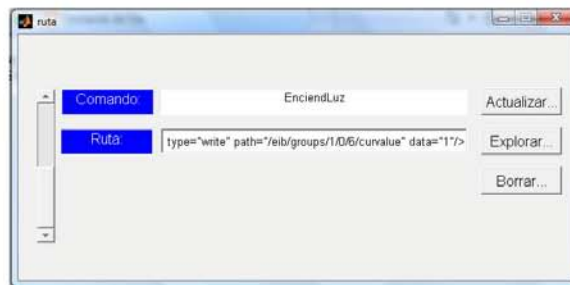


Figura 5.20. Comandos de voz entrenados.

Introducimos la acción deseada que se ejecute cuando se reconozca dicho comando y damos click en "**Actualizar**", el botón "**Explorar**" sirve para buscar el archivo al cual se ligara el comando de ser la opción deseada, siempre que queramos modificar la ruta tenemos que dar click en "**Actualizar**". Podemos borrar algún comando por medio del botón "**Borrar**".

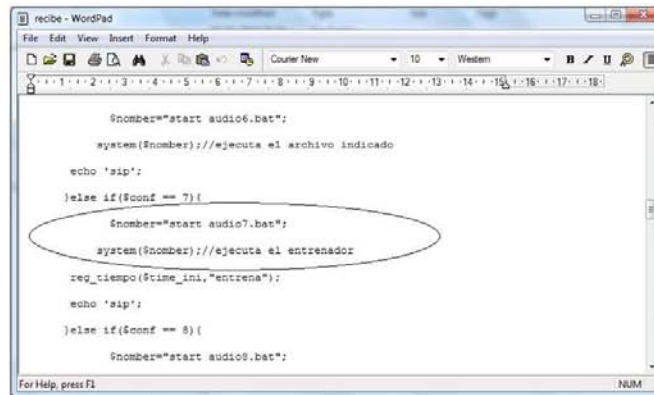
5.3.2 Motor de reconocimiento de voz.

El motor de reconocimiento como ya hemos dicho consta de dos partes: etapa de entrenamiento y la de reconocimiento. El motor está programado en Matlab en dos archivos, `entrena_web` y `reconoce_web` estos a su vez utilizan diversas funciones para el procesado de la señal.

Matlab nos permite crear archivos ejecutables que disparamos por medio de código PHP, una vez guardados los archivos de audio en el servidor.

5.3.2.1 Entrenamiento.

Una vez que el cliente envía las 7 señales de voz necesarias para el entrenamiento, por medio del comando *system()* de código PHP se manda a ejecutar el archivo *audio7.bat*, en la figura 5.21 podemos observar dicha parte de código.



```
recibe - WordPad
File Edit View Insert Format Help
Counter New 10 Western
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

$number="start audio6.bat";
system($number);//ejecuta el archivo indicado

echo 'sip';
}else if($conf == 7){
    $number="start audio7.bat";
    system($number);//ejecuta el entrenador
    reg_tiempo($time_ini,"entrena");
    echo 'sip';
}else if($conf == 8){
    $number="start audio8.bat";

```

Figura 5.21. Pagina recibe.php.

El archivo *audio7.bat* contiene una serie de comandos MS-DOS, los cuales podemos observar en la figura 5.22.



```
audio7 - Notepad
File Edit Format View Help
gen_arch
converter amr2wav audio.amr audio7.wav
entrena_web
exit
Ln 1, Col 1
```

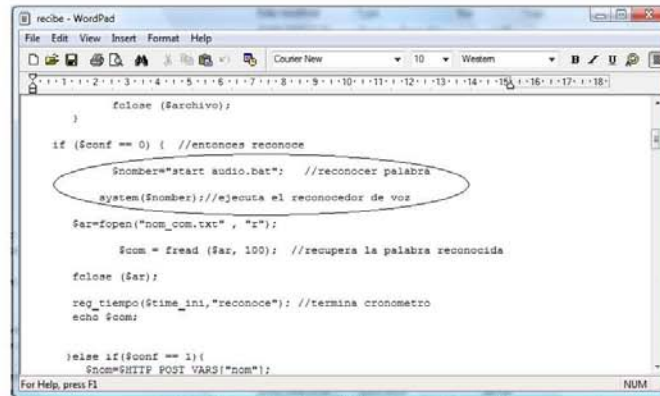
Figura 5.22. Archivo audio7.bat.

- **gen_arch:** Ejecuta un archivo del mismo nombre, el cual toma la señal de voz en código ascii extendido y genera un archivo con extensión *.amr*
- **converter amr2wav audio.amr audio7.wav:** Ejecuta el convertidor de la señal de voz de formato *.amr* a formato *wav*. Esto, debido a que *.amr* es una compresión de la señal de voz, y no puede ser procesada en dicho formato.
- **entrena_web:** por ultimo se ejecuta el archivo *entrene_web.exe* que toma las 7 señales, las procesa y entrena el comando deseado.

Por ultimo se notifica al cliente que el entrenamiento a terminado.

5.3.2.2 Reconocimiento.

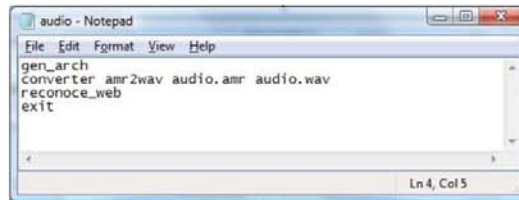
Para reconocer una palabra del mismo modo que en el entrenamiento, se manda a ejecutar un archivo con extensión `.bat` por medio de la función `system()`.



```
folose ($archivo):
}
if ($conf == 0) { //entonces reconoce
    $nombre="start audio.bat"; //reconocer palabra
    system($nombre); //ejecuta el reconocedor de voz
    $ar=fopen("nom_com.txt", "r");
    $com = fread ($ar, 100); //recupera la palabra reconocida
    folose ($ar);
    reg_tiempo($time_ini,"reconoce"); //termina cronometro
    echo $com;
} else if ($conf == 1){
    $com=SKIP POST VARS!"nom";
}
```

Figura 5.23. Ejemplo de código PHP (función system).

El archivo `audio.bat` dispara varios ejecutables entre estos el reconocedor de voz.



```
gen_arch
Converter amr2wav audio.amr audio.wav
reconoce_web
exit
```

Figura 5.24. Archivo audio.bat.

- **gen_arch:** Ejecuta un archivo que lleva el mismo nombre el cual toma la señal de voz en código ascii extendido y genera un archivo con extensión `.amr`
- **converter amr2wav audio.amr audio7.wav:** Ejecuta el convertidor de la señal de voz de formato `.amr` a formato `wav`. Esto debido a que `.amr` es una compresión de la señal de voz, y no puede ser procesada en dicho formato.
- **reconoce_web:** por ultimo se ejecuta el archivo `reconoce_web.exe` que toma la señal, la procesa y devuelve el nombre del comando reconocido escrito en el archivo `nom_com.txt`.

Por medio de código PHP se recupera el nombre del comando reconocido y se envía al cliente para que este lo confirme.


```

recibe - WordPad
File Edit View Insert Format Help
Courier New 10 Westde
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

$nombre="start audio.bat"; //reconocer palabra
system($nombre)//ejecuta el reconocedor de voz

$var=fopen("nom_com.txt", "r");
$com = fread ($var, 100); //recupera la palabra reconocida
fclose ($var);
reg_tiempo($time_ini,"reconocer"); //termina cronometro
echo $com;

}else if($conf == 1){
$nom=$HTTP_POST_VARS["nom"];
$nombre="start audiol.bat";
$archivo=fopen("nombre.txt", "w");

```

Figura 5.25. Recuperación del nombre del comando reconocido.

5.3.3 Servidor domótico.

El servidor domótico utilizado es el llamado: **Linux EIB Home Server**, de distribución libre y aunque es referido al protocolo EIB, este es compatible con el estándar KNX.

El **Linux EIB Home Server** ofrece una interfaz para controlar y monitorear la instalación EIB/KNX. Esencialmente consiste de dos componentes:

- **Homedriver:** interfaz entre **homeserver** y el Bus KNX (capa física). Es el encargado de poner los telegramas al bus mediante el puerto serie de la PC, para esto se utiliza el protocolo FT 1.2.
- **Homeserver:** es una interfaz de comunicación entre la capa de aplicación y el **Homedriver**. Se encarga de crear sockets de comunicación entre los clientes y el **Homedriver** para el paso de información entre el cliente y la red domótica.

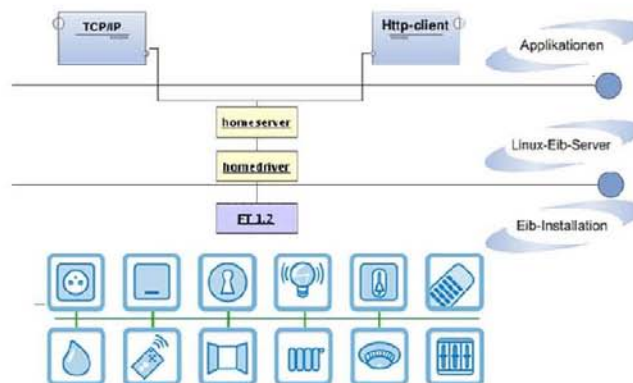


Figura 5.26. Arquitectura del servidor eibcontrol [9].

El flujo de información puede ser de dos maneras:

- Del bus al cliente: EIBbus -> FT 1.2 serial device -> **homedriver** -> **homeserver** -> Clients
- Del cliente al bus: Client -> **homeserver** -> **homedriver** -> -> FT 1.2 serial device -> EIB-Bus

Los clientes pueden ser aplicaciones que corren en la PC programadas por ejemplo en JAVA o clientes HTTP mandando información por medio de navegadores Web.

Los clientes son cualquier programa que utilice el protocolo TCP/IP para conectar con el servidor y enviar a la red alguno de los 4 posibles mensajes:

- **Read**: para saber el estado de algún dispositivo por medio de su dirección de grupo.
- **Write**: para cambiar el estado de un dispositivo por medio de su dirección de grupo.
- **Subscribe**: para monitorear los cambios de algún dispositivo.
- **Unsubscribe**: para cancelar el comando subscribe.

Para la aplicación desarrollada solo nos interesa la instrucción **Write**, con la cual vamos a poder manipular el estado de los dispositivos (encender apagar luces, abrir cerrar persianas, etc.)

La sintaxis para modificar el valor de un dispositivo una vez determinada su dirección de grupo es la siguiente:

```
<eib type="write" path="/eib/groups/2/0/2/curvalue" data="11">
```

En este caso le enviamos el valor "11" al dispositivo con dirección de grupo 2/0/2.

Para el caso concreto de un cliente Web es necesario tener funcionando correctamente el Homedriver, Homeserver y la conexión física mediante puerto serie en el PC (servidor). Una vez verificado esto el cliente puede mandar a escribir un valor al bus por medio de la siguiente URL:

```
http://192.168.169.33:8080/eibhomeserver?instruction=<eib type="write" path="/eib/groups/2/0/2/curvalue" data="11"/>
```

En este caso el PC que funge como servidor tiene la dirección IP 192.168.169.33 y el puerto por el cual escucha por default es el 8080. Con esto mandamos el valor de "11" al dispositivo cuya dirección de grupo es la 2/0/2.

Con lo anterior ya podemos cumplir nuestro objetivo de modificar el estado de los dispositivos de la red domótica.

5.3.3.1 Instalación del servidor domótico en Windows.

1. Lo primero que se necesita es una instalación domótica funcionando, en la figura 5.27 podemos observar la maqueta realizada con la cual controlamos dos luces y un ventilador.


```

C:\WINDOWS\system32\cmd.exe
C:\eibcontrol>set BOU2DEICE=COM1
C:\eibcontrol>echo "homedriver start"
homedriver start
C:\eibcontrol>homedriver\Release\homedriver -d COM1
EIB bus connection initialized
HomeDriver waits for connection from HomeServer!

```

Figura 5.31. Comunicación lista para utilizarse.

Con esto tenemos el driver listo para trabajar.

6. Iniciar el Homeserver: una vez iniciado el homedriver damos click al archivo homeserver_windows.
- 7.

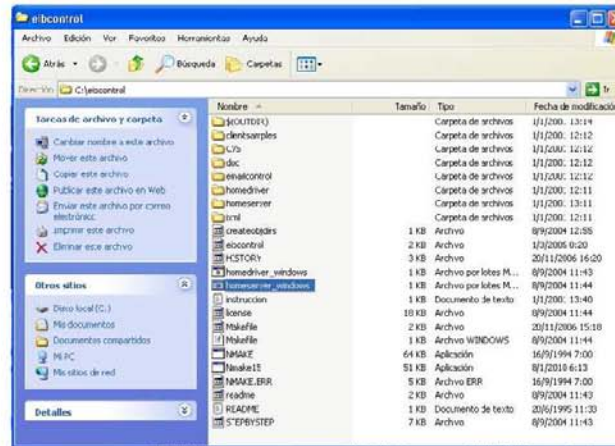


Figura 5.32. Archivo .bat para inicializar el servidor.

Si todo va bien veremos la siguiente pantalla en MS-DOS

```

C:\WINDOWS\system32\cmd.exe
C:\eibcontrol>set DATABASE=homeserver\eibdb.xml
C:\eibcontrol>echo "homeserver start"
homeserver start
C:\eibcontrol>homeserver\Release\homeserver -db homeserver\eibdb.xml
Database homeserver\eibdb.xml has been loaded.
Wait on the QueueSemaphore.
Connect to HomeDriver is OK.

```

Figura 5.33. Servidor listo para utilizarse.

Ya estamos listos para manipular los dispositivos conectados a la red domótica.

5.4 Funcionamiento del sistema.

El sistema desarrollado consta de una serie de peticiones del cliente al servidor, estas pueden ser: autenticar, entrenar comando de voz, reconocer comando de voz o ejecutar comando de voz o por medio de lista de comandos.

5.4.1 Autenticar.

Es necesario autenticar debido a que el reconocimiento de voz es dependiente del locutor, por lo tanto, debemos saber que usuario esta accediendo al sistema y cargar su configuración personal.

El usuario introduce su login y password previamente introducidos al sistema mediante las paginas de administración, además de la dirección URL del servidor Web, dicha información es enviada por medio de Internet al servidor Web utilizando el método POST del protocolo HTTP. En la siguiente imagen podemos ver dicho proceso:

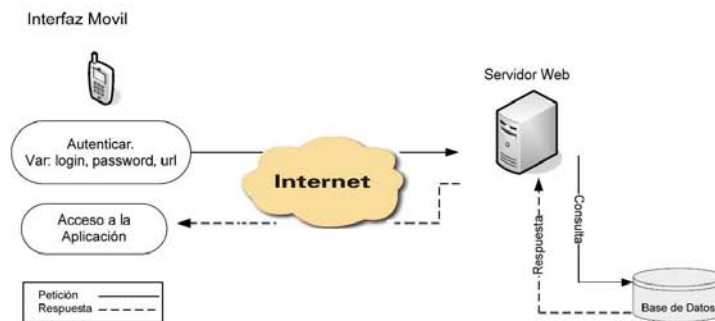


Figura 5.34. Proceso de autenticación.

Una vez autenticados podemos entrenar comandos, reconocer y ejecutar acciones por medio de voz y por lista de comandos.

5.4.2 Entrenar.

Antes de poder ejecutar comandos por voz es necesario realizar un entrenamiento de estos. Para esto es necesario enviar el nombre asociado al comando y la dirección URL del servidor, el sistema muestra por default la URL que introducimos en el proceso de autenticación, pero es posible cambiar esta en cualquier momento. Una vez introducidos estos parámetros iniciamos el proceso de entrenamiento el cual consta de enviar 7 repeticiones del mismo comando al servidor, en la siguiente figura podemos observar dicho proceso.

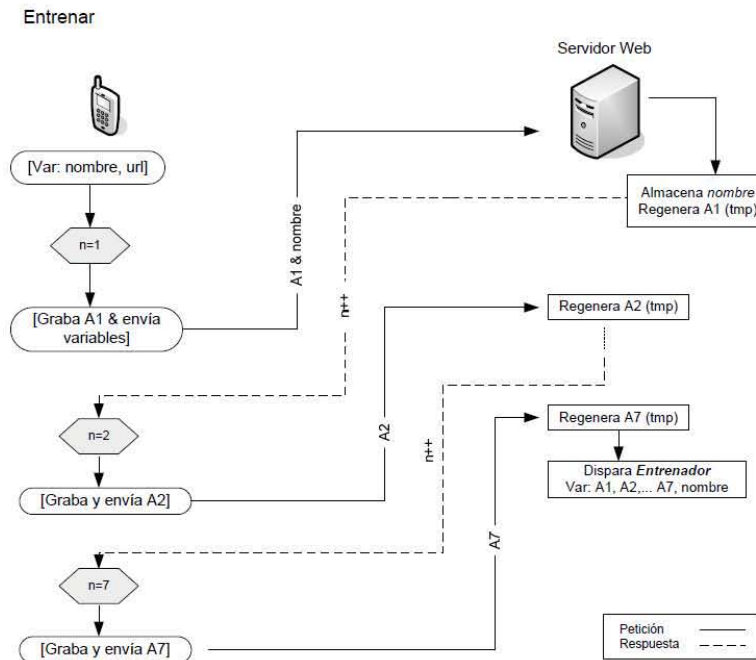


Figura 5.35. Proceso de Entrenamiento.

El cliente graba la señal de voz (A_n) por medio de un formato de compresión llamado Adaptive Multi Rate (AMR), estas son enviadas al servidor donde son convertidas a formato Waveform Audio Format (WAV), cuando se tiene las 7 señales se dispara el entrenador.

5.4.3 Reconocimiento.

En el reconocimiento se graba y reproduce la señal de audio en el cliente antes de ser enviada al servidor, el servidor dispara el reconocedor de voz devolviendo al cliente el nombre del comando reconocido, para que esté confirme si el reconocimiento fue exitoso, de ser así se ejecuta la acción deseada de lo contrario puedes intentarlo de nuevo. Este proceso lo podemos observar en la figura 5.36.

Recordemos que la acción a ejecutar puede ser dispara un archivo, direccionar a alguna pagina Web o enviar un comando a la red domótica.

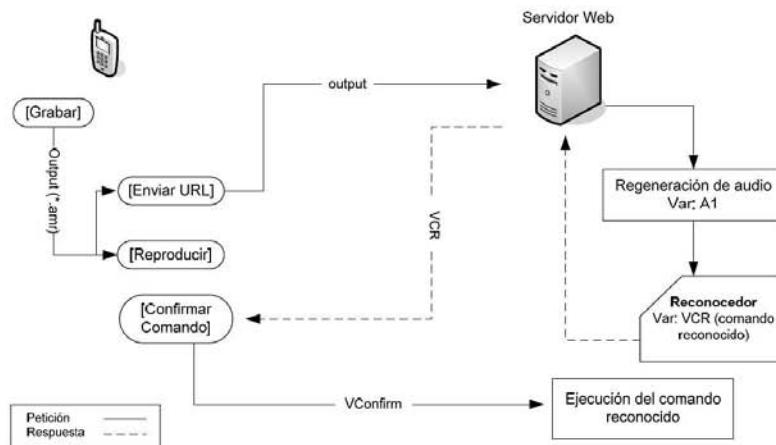


Figura 5.36. Proceso de reconocimiento.

5.4.4 Ejecutar comando.

Para ejecutar un comando en el servidor se puede hacer de dos maneras, por medio de voz o por lista de comandos, por medio de la voz se explicó en el diagrama anterior cuando el usuario confirma el reconocimiento exitoso envía la señal al servidor y este ejecutara el proceso. El proceso de ejecución por medio de click se visualiza en el siguiente diagrama.

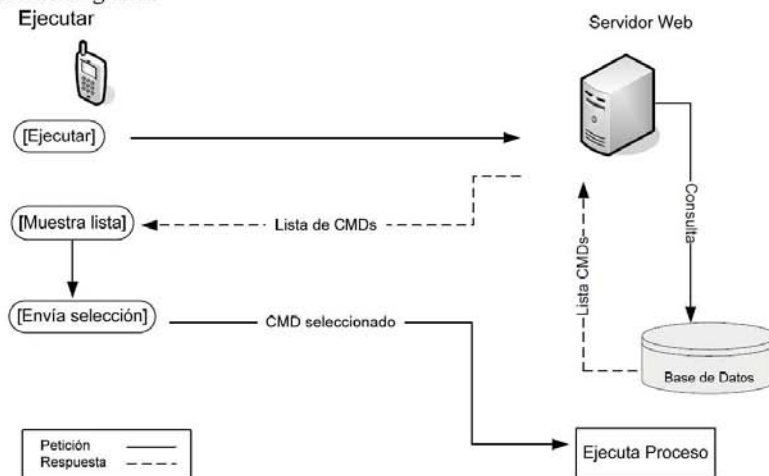


Figura 5.37. Ejecución de comandos por medio de "Click"

Al seleccionar la opción de "ejecutar" en el cliente el servidor envía la lista de comandos previamente guardada en el sistema, al seleccionar un comando de la lista el cliente envía la señal para que el servidor ejecute el proceso.

En la siguiente figura podemos observar la pantalla del servidor doméstico cuando enviamos datos al bus, en este caso el valor “1” al dispositivo con dirección de grupo 1/0/6. Podemos observar que crea un socket, detecta el tipo de comando “write”, lee la dirección de grupo y el valor a enviar, en este caso “1”.

```

C:\WINDOWS\system32\cmd.exe
WebSock has been created. Socket: 1732.
Message parse. type: write
Message parse. path: \\cell/groups/1/0/6/curvalue
Message parse. data: 1
Message parse. gad: 1544 gadStr=1/0/6
Io Database now: path-database.setValue/eib/groups/1/0/6/curvalue data=1
EibType 1 got from the collection.
EIBTYPE=1
callback(). path: database.value/eib/groups/1/0/6/curvalue data: 1
DatenBank-callback: Path: database.value/eib/groups/1/0/6/curvalue
DatenBank-callback: Data: 1
socket Nr: 1732 closed.
The QueueSemaphore has been deblocked.
The output message will be sent to HoneDriver!
Wait on the QueueSemaphore.
callback(). path: database.value/eib/groups/1/0/6/timestamp data: $error
DatenBank-callback: Path: database.value/eib/groups/1/0/6/timestamp
DatenBank-callback: Data: $error
WebSock has been created. Socket: 1732.
socket Nr: 1732 closed.

```

Figura 5.38. Ejecución de comando en la red doméstica.

Pantalla de la conexión con la interface FT1.2 al mandar a escribir el valor de “1” al dispositivo con dirección de grupo 1/0/6.

```

C:\WINDOWS\system32\cmd.exe
ERROR: Repetition!
ERROR: Repetition!
ERROR: Repetition!
EibConnection::changeState: Old state: EibStateReady ==> New state: EibStateSent
Sent state read
EibConnection::changeState: Old state: EibStateSent ==> New state: EibStateReady
ERROR: Repetition!
ERROR: Wrong Checksum of Stop Bit
ERROR: PEI-message
ERROR: Unknown message
ERROR: Unknown message
ERROR: Unknown message
ERROR: Unknown message
ERROR: Unknown message

```

Figura 5.39. Comunicación con el driver al ejecutar algún comando.

BIBLIOGRAFIA CAPITULO 5

- [1] Lizbeth Rodríguez, Eduardo Castañeda, Fabián Lovera. “*Comunicación entre un dispositivo móvil y una PC usando comandos verbales*”. Facultad de Ingeniería, UNAM. 2008.
- [2] JSR 118: MIDP 2.0 <http://jcp.org/aboutJava/communityprocess/final/jsr118/>
- [3] Editor Java: <http://www.eclipse.org/platform>
- [4] JSR 135: Mobile Media API: <http://jcp.org/en/jsr/detail?135>
- [5] Sergio Gálvez y Lucas Ortega: “Java a tope: J2ME (Java 2 Micro Edition)”. Edición electrónica.
- [6] Lenguaje PHP: <http://php.net/index.php>
- [7] xampp: <http://www.apachefriends.org/es/xampp.html>
- [8] eibcontrol: <http://eibcontrol.sourceforge.net/>
- [9] documentación Eibcontrol: <http://eibcontrol.sourceforge.net/Linux-EIB-Home-Server-eng.html>
- [10] EIBA Handbook: EIB Association: www.eiba.org

6. PRUEBAS Y CONCLUSIONES.

Se utilizaron 2 PC's en PC1 se encuentra el servidor Web y el motor de reconocimiento de voz, en PC2 el servidor domótico, fue necesario instalar el servidor domótico en la PC2 debido a que esta cuenta con puerto serie el cual utilizamos para acceder a la red domótica.

La instalación domótica consta de los siguientes dispositivos:

Dispositivo	Descripción
Siemens RS232 Interface N 148-04	Interface entre puerto serie RS232 y bus EIB/KNX por medio del protocolo FT1.2.
Siemens 5WG1 191-5AB11	Accesorio para administración de energía.
ZENNiO ZPS 160M ZN1PS-160M	Fuente de alimentación KNX 160 [mA]
ZENNiO LUZEN ONE ZN1DI-4001	Dimmer Universal, utilizado para regular cargas.
ZENNiO InZennio Z38 ZN1VI-TP38	Pantalla táctil para controlar dispositivos conectados al bus, contiene control IR
Weinzierl KNX USB Interface 310	Interfaze USB a bus EIB/KNX, utilizado para programar los dispositivos mediante el ETS.

Tabla 6.1. Dispositivos KNX utilizados.

Mediante el ETS programamos los dispositivos. Asignándole la dirección física, direcciones de grupo y parámetros a los objetos de cada dispositivo.

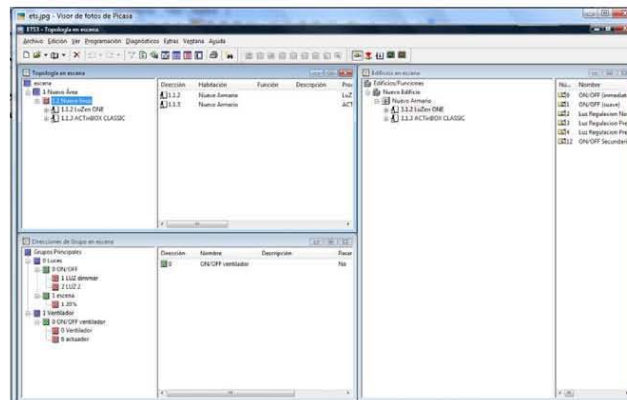


Figura 6.1. Pantalla del ETS al programar los dispositivos.

La maqueta final es la siguiente:

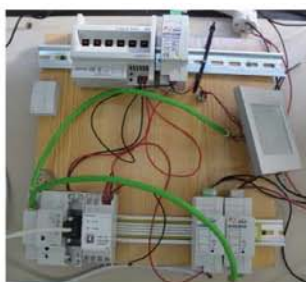


Figura 6.2. Red domótica.

Con la cual controlamos tres cargas (Luz1, Luz2 y Ventilador) ya sea por comandos de voz y por lista de comandos.

Contamos con 2 sistemas de reconocimiento de voz el que utiliza coeficientes Mel Cepstral con modelos ocultos de Markov (SR1) y otro que utiliza coeficientes LPC con cuantización vectorial (SR2) para los cuales realizamos pruebas con los siguientes comandos:

- | | |
|----------------------|--------------------|
| 1. Luz | 6. Luz 30% |
| 2. Apagar Luz | 7. Poner Música |
| 3. Ventilador | 8. Abrir Persianas |
| 4. Apagar Ventilador | 9. Cerrar Puertas |
| 5. Ver Noticias | 10. Televisor |

Resultados para el SR1 (Mel Cepstral y Markov):

	Comando Reconocido										% promedio	
	1	2	3	4	5	6	7	8	9	10		
1	10											
2		10										
3	1		9									
4				10								
5					10							
6	5					5						
7	2				3		4		1			
8	1							9				
9									10			
10							1			9		
% de acierto	100	100	90	100	100	50	40	90	100	90		86
% de error	0	0	10	0	0	50	60	10	0	10		14

Tabla 6.2. Resultados del sistema de reconocimiento 1.

Resultados para el SR2 (Coeficientes LPC y Cuantización Vectorial):

	Comando Reconocido										% promedio	
	1	2	3	4	5	6	7	8	9	10		
1	8					2						
2		7		1		2						
3			8	1		1						
4				9						1		
5					9	1						
6						10						
7							10					
8				1				9				
9									10			
10				1	1	1				7		
% de acierto	80	70	80	90	90	100	100	90	100	70	87	
% de error	20	30	20	10	10	0	0	10	0	30	13	

Tabla 6.3. Resultados del sistema de reconocimiento 2.

Podemos observar que el porcentaje total de acierto para el SR1 es de 86%, aunque los porcentajes de acierto para cada una de las palabras es mayor entre 90 y 100% excepto para 2 palabras (Luz 30% y Poner Música). Esto se puede arreglar cambiando dichas palabras por algunas que funcionen mejor, por ejemplo Luz 30% por Luces al 30% y Poner Música por Pon Música, aunque se deben realizar pruebas para encontrar comandos que no se confundan entre si.

En el caso de SR2 el porcentaje de acierto es de 87% mayor que SR1 pero los porcentajes de acierto individuales no son uniformes y oscilan entre 70, 80, 90 y 100 %.

En las siguientes imágenes podemos observar algunos tiempos de entrenamiento y de reconocimiento para SR1 y SR2.

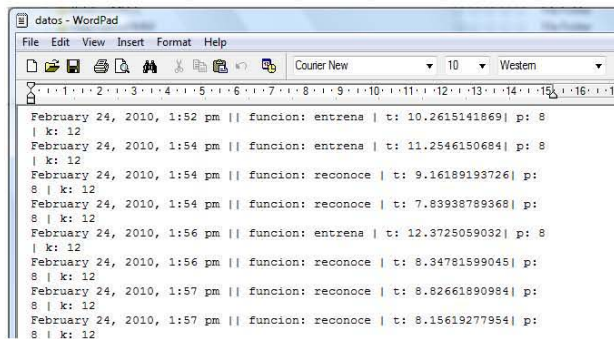


Figura 6.3. Tiempos de entrenamiento y reconocimiento para SR1.

```

February 3, 2010, 3:07 pm || funcion: reconoce | t: 12.4797580242| p: 8
| k: 12
February 3, 2010, 3:08 pm || funcion: reconoce | t: 12.0710909367| p: 8
| k: 12
February 3, 2010, 3:11 pm || funcion: entrena | t: 33.5255231857| p: 8
| k: 12
February 3, 2010, 3:11 pm || funcion: reconoce | t: 9.57054805756| p: 8
| k: 12
February 3, 2010, 3:12 pm || funcion: reconoce | t: 14.9803800583| p: 8
| k: 12
February 3, 2010, 3:12 pm || funcion: reconoce | t: 12.3590168953| p: 8
| k: 12
February 3, 2010, 3:13 pm || funcion: reconoce | t: 12.1387181282| p: 8
| k: 12

```

Figura 6.4. Tiempo de entrenamiento y reconocimiento para SR2.

En general los tiempos de entrenamiento se mantienen en un intervalo relativamente constante ya que siempre se procesan solo 7 señales, en el caso de reconocimiento los tiempos aumentan a medida que hay más palabras en el sistema.

El SR1 es más rápido en el entrenamiento que el SR2. Para la etapa de reconocimiento depende del número de palabras almacenadas, para este caso tenemos las mismas 10 palabras en cada uno de los dos sistemas y el SR1 es más rápido que el SR2 por unos segundos. Aunque este tiempo varía dependiendo de la longitud de la palabra dicha.

Conclusiones

En esta investigación se logró obtener un prototipo funcional del sistema domótico el cual sirve de interfaz entre los usuarios y los dispositivos conectados a la red KNX, logrando ejecutar comandos de voz dentro de esta. La aplicación desarrollada en J2ME fue probada exitosamente en el teléfono celular modelo W810i de la marca Sony Ericsson y puede ser ejecutada en dispositivos que cuenten con la capa de configuración CLDC 1.1 y perfil MIDP 2.0.

El reconocimiento de voz es un campo un tanto difícil ya que depende de factores como la cantidad de ruido en el ambiente, entonación del locutor, etc.

Decidimos realizar un reconocimiento de palabras aisladas dependiente del locutor ya que en este se puede alcanzar porcentajes de acierto elevados, alrededor del 99%, además que en un entorno automatizado creemos suficiente tener alrededor de unos 20 o 30 comandos de voz. Así que no es necesaria mayor interpretación verbal como lo puede dar el reconocimiento de voz continua. Alcanzamos un porcentaje de acierto del 86% para 10 comandos donde 8 de estos estuvieron en un porcentaje de acierto de entre 90 y 100 %.

Dejamos el reconocimiento de voz del lado del servidor para que en el dispositivo móvil solo sea necesario características mínimas, es decir, tener conexión a red y que sea capaz de grabar la voz.

En general los objetivos fueron alcanzados, proponemos mejorar el tiempo de reconocimiento, esto se puede lograr migrando el código a un lenguaje de alto nivel como Java o C/C++. Mejorar el porcentaje de reconocimiento y agregar algoritmos más robustos para cancelación de ruido ambiental.

APENDICE A: PROGRAMACION DE DISPOSITIVOS.

Para programar los dispositivos que componen la red domótica utilizamos la herramienta de software llamado ETS 3, son las siglas de Engineering Tool Software. ETS ® es una marca registrada por la Konnex Association. <http://www.knx.org/knx-tools/ets/description/>. En la siguiente figura podemos observar la apariencia del software.

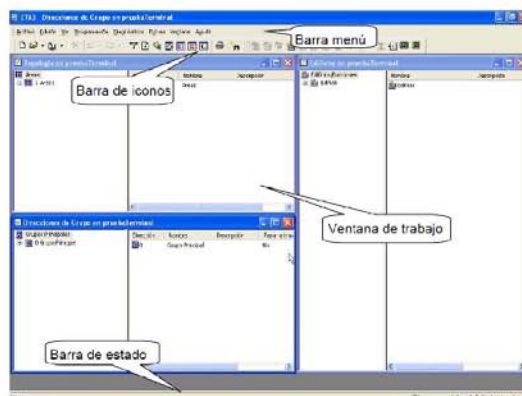


Figura A.1.

Se puede modificar la apariencia mediante el menú *Extras/Opciones*.

Base de datos

Cuando se abre por primera vez el ETS 3 Professional crea una base de datos con el nombre EIB.DB, en la carpeta *Ets\Database*. Esta carpeta estará ubicada en la ruta donde está instalado (normalmente, C:\Archivos de programas. Así, la ruta completa sería C:\Archivos de programas\Ets\Database). Esta base de datos es la base central de todos los proyectos. Contiene tanto los datos de los productos del fabricante, como los datos del proyecto introducidos por el usuario.

En un primer momento, esta base de datos está completamente vacía. Por lo tanto, lo habitual es comenzar por importar datos de productos.

Importar productos

Para poder trabajar con el ETS 3 Professional, deben importarse los datos de los dispositivos en dicha base de datos. Para ello, tenemos la función *Importar* del ETS, a la que podemos acceder a través de la barra de menú *Archivo/Importar...* Las bases de datos de productos de los fabricantes son gratuitas y podemos encontrarlas en CD-ROM ó, aún más cómodo, en Internet.

La función *Importar* permite importar no sólo datos de productos, sino también datos de proyectos. En este punto, hablaremos sólo de datos de productos. Las bases de datos con datos de productos tienen la terminación *.vd?* El signo de interrogación equivale a los siguientes valores:


- Los archivos .vd5 caracterizan los datos de producto soportados por el ETS 3.0f y versiones posteriores.
- Los archivos .vd4 caracterizan los datos de producto soportados por el ETS 3.0d y versiones posteriores.
- Los archivos .vd3 caracterizan los datos de producto que poseen el formato de datos del ETS 3 Professional.
- Los archivos .vd2 son aquéllos que han sido desarrollados para el ETS 2 V1.3 ó V1.2. Mediante la importación se convierten en el formato ETS 3 Professional
- Los archivos .vd1 y .vdx contienen bases de datos de productos del ETS 2 V1.1 ó V1.0. También estos archivos pueden importarse correctamente en el ETS 3 Professional.

Si no se desea importar la base de datos de producto completa de un fabricante, deberá activarse la opción *Deseo seleccionar los elementos a importar*, del diálogo de Importar. A continuación, el ETS 3 Professional mostrará en la ventana *Importación selectiva* una lista de los productos del fichero de datos de productos seleccionados. Entonces pueden importarse, según se desee, todos o sólo los productos marcados.



Figura A.2.

Catálogo de productos

Mediante el catálogo de productos obtenemos una panorámica de estos y los fabricantes disponibles en la base de datos del ETS 3 Professional actual. Para llegar al catálogo de productos, basta hacer clic sobre el icono  de la barra de iconos.

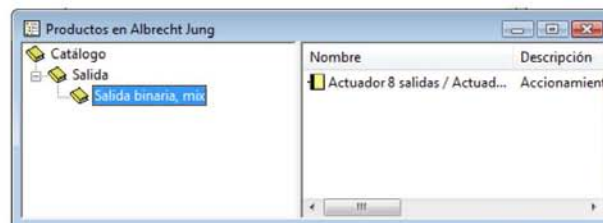



Figura A.3.

Creación de un nuevo proyecto

Para crear un proyecto nuevo podemos proceder de dos formas: haciendo clic en el botón  o seleccionando el comando *Archivo/Nuevo Proyecto*. En la pantalla aparecerá el diálogo “Crear proyecto nuevo”, en el cual debemos introducir un nombre para el mismo. Además, en esta pantalla debe seleccionarse el medio utilizado (TP para Twisted Pair ó PL para Powerline). Si dejamos marcada la función “Generar línea 1.1”, se crearán automáticamente una línea y un área. De lo contrario, por el momento el proyecto no tendrá una estructura de topología.

Vistas del diseño de proyecto

En el ETS 3 Professional podemos visualizar los proyectos a través de diferentes ventanas de trabajo, que pueden ser utilizadas simultáneamente. La disposición de estas ventanas de trabajo puede modificarse según las necesidades de cada momento. Para ello, el ETS 3 Professional ofrece:

- Vista de edificio y funciones
- Vista de topología de Bus
- Vista de direcciones de grupo
- Vista de aparatos

Vista de edificio/funciones

La vista de edificio es la fundamental del ETS 3 Professional y se utiliza para estructurar los proyectos KNX de acuerdo con la estructura real del edificio, así como para introducir los componentes KNX en las diferentes estancias. Los aparatos pueden introducirse en habitaciones o armarios domésticos.

Una vista jerarquizada puede ser muy útil para proporcionar una visión general en el caso de proyectos de grandes dimensiones.

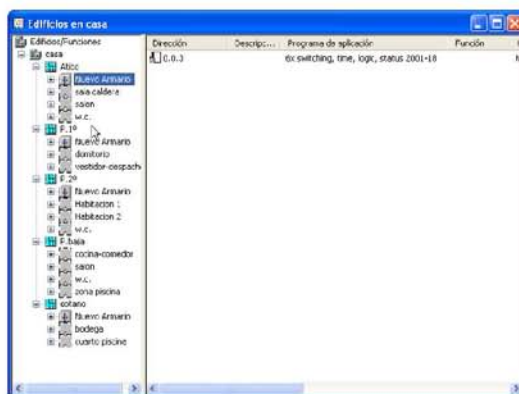


Figura A.4.

Vista de direcciones de grupo

La vista de direcciones de grupo sirve para la generación y definición de direcciones de grupo. Esta vista se utiliza, en combinación con la vista de edificio, para unir los objetos de comunicación con las direcciones de grupo correspondientes. Las direcciones de grupo aparecen estructuradas en 2 ó 3 niveles, dependiendo de la opción preseleccionada.

La representación de las direcciones de grupo en distintos niveles no tiene consecuencias funcionales; sólo sirve para esquematizar. En la presente documentación nos basaremos en una estructura de 3 niveles. Si se selecciona un subgrupo, en vista de lista de la derecha podremos ver a qué objeto de comunicación se ha asignado la dirección de grupo.

La vista de estructura (parte izquierda) muestra las direcciones de grupo existentes (aquí, direcciones de grupo en tres niveles).



Figura A.5.

Vista de topología de bus

La vista de topología de Bus se utiliza para definir la estructura real del Bus, así como para asignar las direcciones físicas de los aparatos. Esta vista, que puede usarse simultáneamente a otras, muestra el proyecto KNX en relación con la estructura del Bus. En ella podemos ver los aparatos asignados a las distintas líneas. Las líneas/áreas Twisted Pair y Power Line se representan con distintos símbolos.

La vista de estructura (parte izquierda) nos muestra la topología de Bus del proyecto KNX actual y en la parte derecha encontramos una vista de lista de los objetos marcados en la ventana de la izquierda.

La topología se genera, por regla general, automáticamente si se asignan direcciones físicas en otro lugar. Todos los elementos jerárquicamente superiores, como áreas y líneas, se introducen de acuerdo con las direcciones adjudicadas.

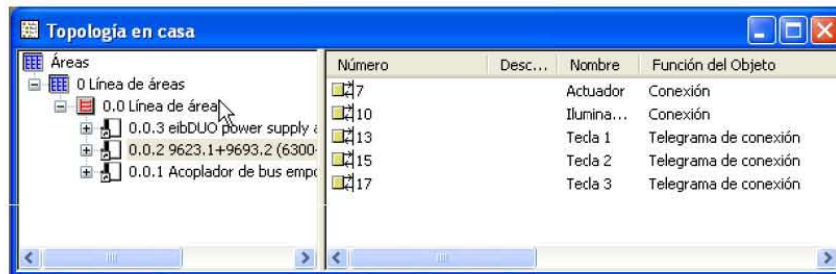


Figura A.6.

Vista de aparatos

Mediante el icono de la barra de iconos o la función *Ver/Vistas del proyecto/Todos los aparatos* podemos abrir la ventana de trabajo *Todos los aparatos*.

En la ventana *Todos los aparatos* se muestran todos los componentes del proyecto, incluidos los que aún no han sido asignados a ninguna habitación, función o línea. Es una buena manera de ver una panorámica general del proyecto (para comprobar, por ejemplo, si hay aparatos sin dirección física asignada)

Este listado puede ser muy largo, por lo que se recomienda introducir un filtro. Este tema se trata en detalle en el capítulo de diseño de proyecto avanzado.

Con los aparatos que aparecen en esta ventana pueden llevarse a cabo todas las acciones disponibles en la ventana de edificio o en la ventana de topología: modificar aparatos, modificar objetos, etc.



Figura A.7.

NUEVO PROYECTO

Vamos a basarnos en un ejemplo para explicar el diseño rápido de un proyecto KNX pequeño. Este se divide en dos partes, Diseño de proyecto y puesta en marcha.

Diseño del proyecto

Prácticamente un proyecto consta de una estructura física (edificios, habitaciones y aparatos) y una lógica, direcciones físicas y de grupo de los aparatos. Además de una topología de la red.

Con el botón NUEVO se abre la ventana Crear Proyecto Nuevo. Aquí debe asignarse el nombre del proyecto en el campo Nombre. Y seleccionar el tipo de medio, en este caso TP.

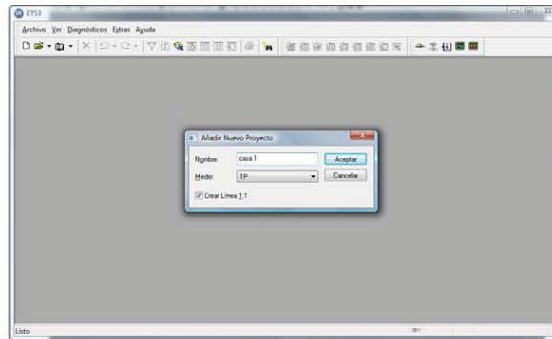


Figura A.8.

Escogemos las opciones *Ver/Entornos de Trabajo/SmallProject* y se abrirá la ventana de de Edificios y la de Direcciones de grupo.

En la barra de iconos seleccionamos Introducir edificio, introducimos el nombre de este y aparecerá en la ventana de edificios.

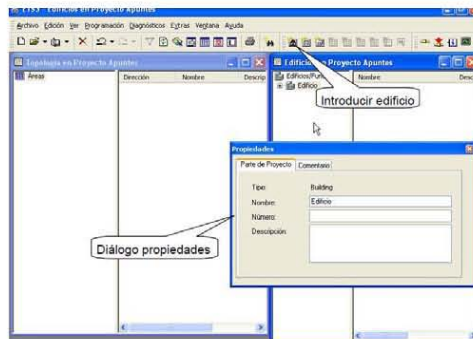


Figura A.9.

Ahora tenemos que agregar ya sea un armario o una habitación, para esto seleccionamos el edificio que introducimos previamente y damos clic en el icono de añadir habitación. Y colocamos el nombre de esta.

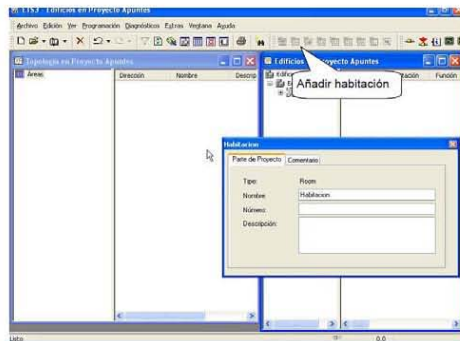



Figura A.10.

Para introducir aparatos abrimos el buscador de productos mediante el icono . Los aparatos deben importarse previamente. Seleccionamos las opciones de búsqueda y damos clic en el botón “Encontrar”. Una vez encontrado damos doble clic sobre este y se introducirá en la habitación seleccionada.

El ETS **asigna automáticamente las direcciones físicas** en orden ascendente, a menos que esta opción haya sido desactivada en *Extras/Opciones.../Presentación*.


Como resultado, los aparatos introducidos aparecen en la estructura de árbol (debajo de la habitación correspondiente) de la vista de edificio. Y al expandirlos podemos ver los objetos con los que cuenta este.

Haciendo doble clic sobre el aparato introducido se abre el diálogo de propiedades, que está compuesto de muchas fichas. Dado que no se ven todas las pestañas de una vez, hay dos flechas para poder acceder a todas ellas. Algunas fichas tienen sólo carácter informativo, es decir, no pueden modificarse. La más importante es la que lleva el título *General*.

En la ficha *General* podemos modificar la dirección física del aparato. Un cambio en el cuadro de texto *Dirección física* asigna automáticamente al aparato otra línea y/u otra área en la topología del Bus.

Nota: ¡Sólo se pueden introducir direcciones libres!

Además, se puede introducir un comentario en el cuadro de texto *Descripción*. La práctica recomienda que dicha descripción contenga los siguientes elementos: qué es, dónde está instalado y para qué sirve.

Mediante el botón  situado en la ficha *General* del diálogo de propiedades de los aparatos se llega al diálogo de parámetros, donde podemos ajustar los parámetros de los aparatos. Los parámetros definen la función concreta del programa de aplicaciones (Ej.: puede seleccionarse, como en el ejemplo de más arriba, la función de los LEDs y la de las teclas)

La ventana “Parámetros” depende de cada producto en concreto. Para obtener información detallada sobre los parámetros individuales de un aparato, habrá que acudir a las descripciones de los programas de aplicaciones que proporciona cada fabricante.

Los parámetros se dividen en grupos, que aparecen en la columna de la izquierda de la ventana de parámetros. Seleccionando cada uno de los grupos de la lista se accede a los parámetros correspondientes en la mitad derecha de la ventana. Para devolver a los parámetros los valores originales definidos por el fabricante basta hacer clic en el botón *Por defecto*.

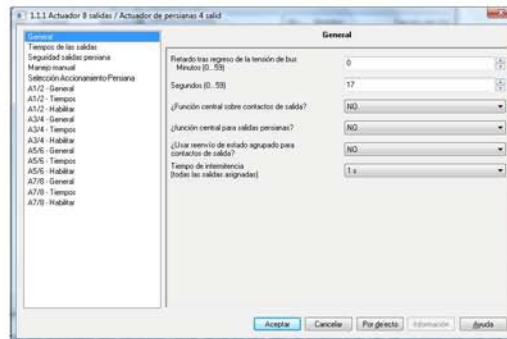


Figura A.11.

Haciendo clic sobre el signo \oplus que encontramos delante de cada aparato en la vista de árbol (mitad izquierda de la ventana) llegamos a los objetos de los aparatos. Los objetos de comunicación del aparato en cuestión se desplegarán hacia abajo.

Para abrir el diálogo de propiedades de los objetos haremos doble clic sobre cada uno de los ellos. A continuación, podemos modificar esas propiedades de los objetos.

En el caso de los objetos de comunicación, puede modificarse la prioridad de envío del telegrama definida por defecto por el fabricante. Las prioridades que pueden seleccionarse son las siguientes:

- **Baja** prioridad baja para funciones en las que el tiempo no es crítico
- **Alta** prioridad normal para funciones de accionamiento manual
- **Alarma** prioridad alta para funciones en las que el tiempo es crítico

Consejo: la prioridad debería modificarse únicamente en casos excepcionales.

En la segunda ficha, Direcciones de Grupo, aparece un listado de todas las direcciones de grupo asignadas a este objeto y también se pueden editar (Ej.: se puede borrar la asignación a un objeto).

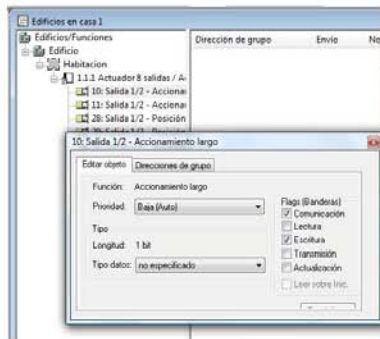


Figura A.12.

Hasta ahora tenemos la estructura física y ahora vamos a introducir las direcciones de grupo, las cuales nos dan la conexión lógica entre los dispositivos.

Los objetos destinados a cumplir en combinación con otros una función concreta deben tener una “Unión lógica” mediante direcciones de grupo, las cuales pueden estar organizadas en dos o tres niveles.

La presentación en dos niveles se compone de grupo principal (0 a 15) y subgrupo (0 a 2047). La presentación en tres niveles se compone de grupo principal (0 a 15), grupo intermedio (0 a 7) y subgrupo (0 a 255). Los distintos niveles de una dirección de grupo se representan siempre separados por una barra „/“ (Ej.: 1/0/2).

En *Extras/Opciones/Presentación* podemos modificar la configuración de los niveles de las direcciones de grupos. Se recomienda utilizar la estructura de las direcciones de grupo como punto de referencia organizativo.

La estructura de direcciones de grupo que vemos en el dibujo (mitad izquierda de la ventana de direcciones de grupo) se crea de manera similar a la estructura del edificio de la ventana de edificio (mediante el icono correspondiente de la barra de iconos).



Figura A.13.

Una vez creadas las direcciones de grupo, desde grupo principal, grupo intermedio y dirección, estas tienen que ser asignadas a los objetos correspondientes.

Para asignar las direcciones de grupo a los objetos se recomienda tener abiertas dos ventanas de trabajo simultáneamente. Hay varias posibilidades para llevar a cabo la asignación, siendo la más rápida (mediante “arrastrar & soltar”) la que veremos a continuación:

Seleccionar y arrastrar con el ratón (manteniendo apretado el botón izquierdo) la dirección de grupo deseada hasta el objeto correspondiente (o el objeto hasta la dirección de grupo correspondiente); por último, soltar. Con este proceso habremos asignado las direcciones de grupo a los objetos.

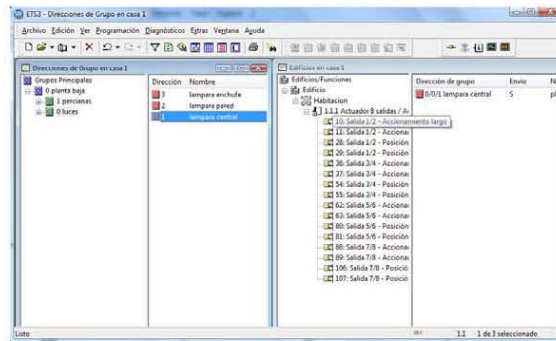


Figura A.14.

Es posible la unión múltiple de un objeto de comunicación mediante la asignación de varias direcciones de grupo. Si el objeto de comunicación tiene la función de un sensor, entonces se utilizará la dirección de grupo asignada en primer lugar como dirección de destino del telegrama (dirección de grupo emisora). Si se desea modificar la designación de la dirección de grupo emisora, debe hacerse clic con el botón derecho del ratón sobre el objeto de comunicación de la vista de lista de la ventana de edificio (mitad derecha de la ventana).

Nota: Hay también sensores que sólo permiten una dirección por objeto de comunicación respectivamente. El número de direcciones de grupo que pueden asignarse a un objeto de comunicación, así como el número máximo de asociaciones (número de asignaciones por producto) dependen de cada producto en concreto. Sólo pueden crearse uniones con direcciones de grupo entre objetos de comunicación del mismo tipo (1 Bit, 4 Bit, etc.). La dirección de grupo adopta, con la primera asignación, el tipo del objeto de comunicación.

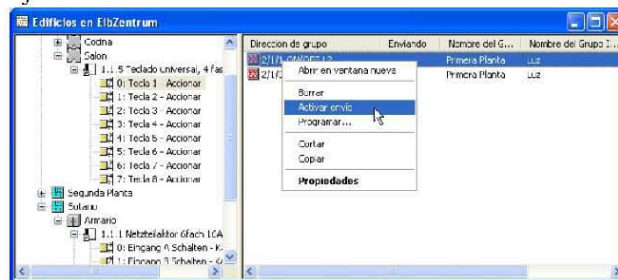


Figura A.15.

PUESTA EN MARCHA

La programación de los dispositivos se puede realizar por medio de una interfaz serie, interfaz USB o por una interfaz con conexión a red.

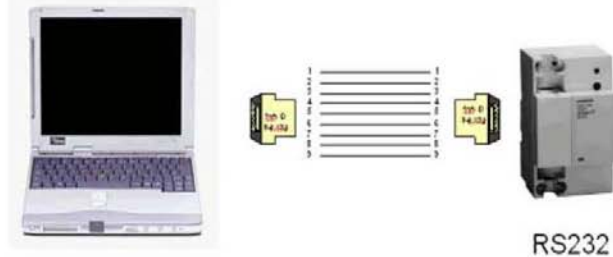


Figura A.16.

El proyecto de la instalación KNX debe estar contenido en la base de datos. Si el proyecto se realizó en otra PC, tenemos dos posibilidades: La primera posibilidad consiste en exportar el proyecto con el ETS 3 e importarlo en el PC que se va a utilizar para la puesta en marcha.

La segunda posibilidad consiste en copiar toda la base de datos (Ej.: „eib.db“) desde un PC a otro. Este proceso tiene la ventaja de que, en caso de una ampliación posterior, todos los productos están incluidos en la base de datos. El inconveniente, por el contrario, es que la base de datos puede llegar a tener un gran tamaño (varios 100 MB).

Comenzar la puesta en marcha

Los iconos de menú específicos para la puesta en marcha y el diagnóstico son los siguientes:

- **Online:** repara el PC para acceder al Bus.
- **Programar:** Abre el diálogo de **Programación** para direcciones físicas y programas.
- **Información de aparato:** Abre el diálogo **Información de aparatos**.
- **Monitor de Bus:** Abre el **monitor de Bus**.
- **Monitor de grupos:** Abre el **monitor grupos**.

Además, en el menú Puesta en marcha o Diagnóstico encontramos las funciones:

- **Desprogramar:** Repone un componente que ya se ha puesto en marcha en su estado original.
- **Reinicializar aparato:** Envía una orden de reiniciar a un componente Bus.
- **Direcciones físicas:** Comprueba los componentes Bus existentes en la instalación con el proyecto.

Acceso al bus

Antes de poder proceder con la puesta en marcha, es imprescindible, por una parte, conectar la interfaz de comunicación al Bus con el PC de puesta en marcha mediante el cable adecuado y, por otro, configurar correctamente los ajustes en dicho PC. Podemos utilizar un interfaz serie (RS232), un interfaz USB ó una conexión de red. Independientemente del interfaz utilizado, antes de la primera puesta en marcha, es necesario verificar las configuraciones de comunicación del ETS.

En el menú *Extras* encontramos el apartado *Opciones*. Las posibilidades de configuración correspondientes están en la ventana *Comunicación*.



Figura A.17.

Al hacer clic sobre el botón *Configurar interfaces* de la ventana *Comunicación* se abre la ventana de diálogo *ETS Connection Manager*. En el cuadro *Propiedades* puede seleccionarse una de las siguientes posibilidades de conexión:

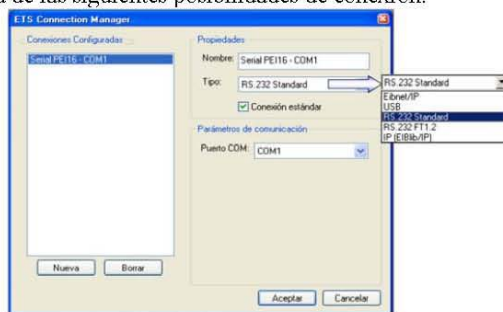


Figura A.18.

- USB, comunicación vía USB.
- RS.232, conexión serie estándar.
- RS.232 FT1.2, conexión mediante BCU.
- IP (EIBlib/IP), comunicación mediante iETS.
- KNXnet/IP, comunicación IP (tunnelling).
- KNXnet/IP Routing comunicación IP (routing).

Configurar la interfaz local

Al hacer clic sobre el botón “Configuración” de la ventana “Opciones” - “Comunicación” se abre la ventana de diálogo “Configuración del Interfaz local”. Aquí

se introduce la dirección física del interfaz. La dirección debe ser tal que coincida en número de área y línea con la localización actual del interfaz. El número de componente, por su parte, debe ser una dirección no existente en el proyecto. Lo más aconsejable es utilizar el “255” como número de usuario ya que, al ser el más alto, es más probable que no se utilice en la instalación.



Figura A.19.

Es recomendable tener solo un proyecto abierto para evitar confusiones a la hora de programar.

Vistas

En principio, podemos utilizar todas las vistas, cada una de las cuales tiene sus puntos fuertes. La vista de edificio tiene la ventaja de que, durante la puesta en marcha, aunque las direcciones físicas de los aparatos no sean correlativas, no habrá que seguir largos caminos innecesarios. La vista de topología, por su parte, tiene la ventaja de poder poner en marcha varias habitaciones simultáneamente. Además, puede seleccionarse una línea y, a continuación, programarla completamente de una vez.

Con la vista de *Aparatos modificados* vemos sólo los aparatos KNX cuyas propiedades han sido modificadas por el usuario desde la última programación (descarga). Esta función está disponible en el submenú *Vistas del Proyecto* del menú *Ver*.

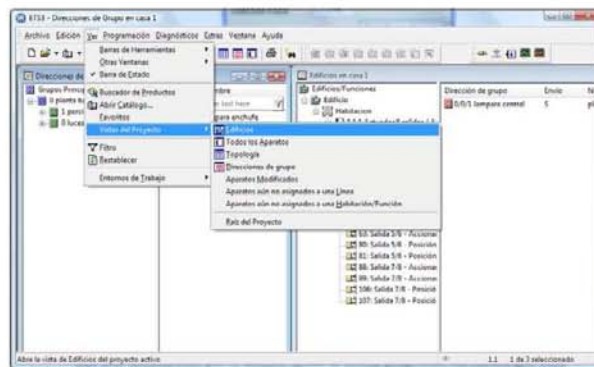


Figura A.20.

GLOSARIO DE TERMINOS.

AMR: Del ingles, Adaptive Multi-Rate. Estándar de compresión de Audio optimizado para codificación de voz.

EIB: Del ingles, European Installation Bus. Estándar de comunicación entre dispositivos inteligentes dentro de hogares y edificios automatizados.

ETS 3: Del ingles, Engineering Tool Software. Herramienta de software independiente **del fabricante** para diseñar y configurar instalaciones inteligentes para el control de casas y edificios hechas con el sistema KNX/EIB.

FT1.2: Protocolo que define el formato de trama a enviarse por capa física, para la comunicación entre dispositivos conectados a la red domótica.

HMM: Del ingles, Hidden Markov Model. Modelo estadístico utilizado para el modelado y reconocimiento de patrones.

HTTP: Del ingles, Hyper Text Transfer Protocol. Protocolo de comunicación usado por la World Wide Web para comunicar los servidores con los navegadores (clientes).

IP: Del ingles, Internet Protocol. Protocolo que permite el envío de paquetes de datos. IP es un elemento fundamental en la estructura técnica de Internet. Ver también Dirección IP.

J2ME: Del ingles, Java Micro Edition. Versión para dispositivos con características limitadas de memoria y procesamiento de la plataforma de programación Java.

KNX: Estándar abierto a nivel mundial para el control tanto de casas como de edificios.

LONWORKS: Estándar a nivel mundial para el control tanto de casas como de edificios.

MÉTODO POST: Método utilizado en el protocolo HTTP para envío de datos desde el cliente al servidor.

MFCC: Del ingles, Mel-Frequency Cepstral Coefficients, coeficientes que representan la potencia espectral de una señal en el tiempo corto.

MIDP: Del ingles, Mobile Information Device Profile. Es una capa dentro de la plataforma J2ME (Java 2 Micro Edition), integrada en el hardware de dispositivos móviles, que permite el uso de programas java denominados MIDlets, tales como juegos, aplicaciones o todo tipo de software.

MySQL: Sistema de gestión de base de datos relacional, multihilo y multiusuario.

PASARELA RESIDENCIAL: Dispositivo que interconecta diferentes redes de una vivienda o edificio (control domótico, telefonía, televisión y tecnologías de la información), a una red pública de datos, como por ejemplo Internet, efectuando en su caso, la adaptación y traducción entre diferentes protocolos.

PDA: Del inglés, Personal Digital Assistant. Siglas con las que se denominan a las computadoras de mano o handhelds. También conocidas como agendas electrónicas, pese a tener muchas más funciones que la de una simple agenda.

PHP: Del inglés, Hyper Text Pre-Processor. Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

PLC: Del inglés Power Line Control. Sistema de transmisión de información por medio de corrientes portadoras confinadas en líneas de alimentación eléctrica convencionales.

POCKET PC: Dispositivo de mano que tiene muchas de las características de las computadoras de escritorio.

PROTOCOLO: Conjunto de reglas y señalizaciones (¿Quién empieza la comunicación? ¿Cómo se comunican? ¿Con quien? ¿Con que prioridad?), que permiten el intercambio de información entre dos o más dispositivos conectados entre sí. Indica el formato de las tramas para la transmisión de datos entre dos dispositivos, además de las técnicas de acceso al medio para los dispositivos.

RED: Interconexión de nodos (agentes, dispositivos...) que intercambian información o recursos. O también, conjunto de elementos independientes interconectados.

REDES DOMÓTICAS: Sistema de dispositivos conectados por medio de protocolos de comunicación, quienes tienen la finalidad de realizar acciones automatizadas dentro de una vivienda o edificio.

SERVIDOR WEB: Aplicación que sirve archivos de un sitio web a petición de los usuarios. Se llama así también a la máquina conectada a la red en la que están almacenadas físicamente las páginas que componen un sitio.

SR1: Sistema de reconocimiento 1.

SR2: Sistema de reconocimiento 2.

TCP/IP: Familia de protocolos de Internet, es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

URL: Del inglés, Uniform Resource Locator. Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

XAMPP: Servidor independiente de plataforma, software libre, que consiste principalmente en una base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

X-10: Estándar a nivel mundial para el control tanto de casas como de edificios.

WAV: Del inglés, Waveform Audio Format. Formato de audio digital normalmente sin compresión de datos, desarrollado y propiedad de las compañías Microsoft e IBM que se utiliza para almacenar y reproducir sonidos en el PC.