



TESINA DE LICENCIATURA

TITULO: Kindergarten: Mecanismo de sensing y comunicación para sistemas context-aware.

AUTORES: Nahuel Lofeudo

DIRECTOR: Dr. Gustavo Rossi

CODIRECTOR: Dra. Silvia Gordillo

CARRERA: Licenciatura en informática

Resumen

En la presente tesina se abordan los mecanismos que permiten a los sistemas context-aware obtener su posición física dentro de un ambiente y comunicarse unos con otros. El enfoque utiliza un mismo hardware de muy bajo consumo de energía para localización y comunicaciones, reduciendo el consumo de energía del sistema móvil en su conjunto, y por ende aumentando su autonomía y practicidad.

Se presenta una combinación de hardware y software a la que se bautizó Kindergarten, un tipo de red de datos y localización diseñada y desarrollada íntegramente con esas dos características en mente, para dispositivos móviles de bajo consumo y alta autonomía.

Se describen los aspectos del hardware de comunicaciones seleccionado especialmente para el proyecto, los circuitos que fue necesario diseñar y construir para permitir la comunicación entre dicho hardware y los sistemas host donde se ejecuta el software context-aware, y los módulos de software de los distintos sistemas integrantes de la red Kindergarten.

Líneas de Investigación

Sistemas context-aware
Arquitecturas de redes
Sistemas de tiempo real
Redes inalámbricas

Conclusiones

Los mecanismos actuales de estimación de posición distan de ser óptimos para sistemas context-aware móviles debido a su alto consumo y limitada precisión de estimación de posición.

El desarrollo aquí expuesto provee una solución para estos problemas a través de la creación de una red inalámbrica de bajo costo de implementación y alta precisión de posicionamiento.

Trabajos Realizados

Relevamiento de mecanismos de comunicación.
Selección de hardware de comunicaciones wireless.
Diseño e implementación de un stack de red inalámbrica para comunicaciones y localización.
Creación de dispositivos de hardware.

Trabajos Futuros

Creación de dispositivos de hardware autónomos.
Mejoras de performance en la transferencia de datos.
Implementación de filtros de localización para mejorar la calidad de la estimación de posición.
Mejora de los algoritmos para disminuir aún más el consumo de energía.

Fecha de la presentación: Diciembre de 2008

Índice general

Agradecimientos	6
Capítulo 1: Introducción.....	7
Descripción general del presente trabajo.....	8
Organización del trabajo.....	9
Capítulo 2	
Estado del arte.....	10
Trabajos previos.....	10
Puerto infrarrojo.....	10
Bluetooth.....	11
Wi-Fi (standard IEEE 802.11b/g).....	12
GPS.....	12
Capítulo 3	
Descripción resumida de La red Kindergarten.....	13
Capítulo 4	
El hardware de la red Kindergarten.....	15
Microcontroladores.....	15
La selección del hardware.....	17
Acerca de la Entrada/Salida en microcontroladores.....	18
La norma IEEE 802.15.4	18
La modulación de IEEE 802.15.4.....	19
El Spread-Spectrum de Secuencia Directa (DSSS).....	20
La trama PHY de 802.15.4.....	21
La trama MAC de IEEE 802.15.4.....	21
Capítulo 5	
La organización lógica de la red Kindergarten.....	24
Motivo del desarrollo.....	24
Modo de operación.....	24
El ciclo de trabajo de la red.....	26
Sub-división de la red.....	28
Formato de las tramas de Kindergarten.....	29
Los mensajes de Kindergarten.....	29
Los mensajes de datos.....	30
El mensaje OOT.....	30
El mensaje POE.....	30
El mensaje de joining Request.....	31
Ruteo de mensajes entre estaciones.....	31
Organización del stack de Kindergarten.....	33
Código común.....	33
Código específico de las estaciones móviles.....	35
Código específico de las estaciones base.....	35
Capítulo 6	
La estimación de posición en Kindergarten.....	38

Descripción del módulo de location awareness	38
Los parámetros A y N.....	39
El parámetro A.....	39
El parámetro N.....	39
Integración con el resto de la estación móvil.....	41
Estructura del software de localización.....	42
Soporte para otros modelos de localización.....	43
Capítulo 7	
La comunicación PDA-CC2431 en las estaciones móviles.....	44
Descripción de Secure Digital.....	44
SDIO.....	46
Mapa de memoria.....	47
Common I/O Area (CIA)	47
Card Common Control Registers (CCCR)	47
Function Basic Registers (FBR)	47
Card Information Structure (CIS)	48
Embedded I/O Code Storage Area (CSA)	48
Los comandos de SDIO.....	48
Secuencia de inicialización.....	52
Implementación del bus SDIO sobre el CC2431.....	52
El hardware.....	53
SPI y su adaptación a las conexiones SDIO.....	53
Recepción de un comando transmitido desde el host.....	54
Stack de SDIO en el CC2431.....	55
La HAL.....	55
El Core	56
Capítulo 8	
La comunicación cableada de las estaciones base.....	57
El controlador enc28J60.....	57
μ IP.....	58
Integración de μ IP con el stack de Kindergarten.....	59
Capítulo 9	
Creación y resultados de una red Kindergarten real.....	60
Prueba de campo.....	62
Capítulo 10	
Conclusiones.....	64
Trabajo futuro.....	64

Índice de ilustraciones

Ilustración 1: Esquema general de los integrantes de una red Kindergarten....	14
Ilustración 2: Microcontrolador.....	15
Ilustración 3: Diagrama de bloques de un microcontrolador típico.....	16
Ilustración 4: Las portadoras I y Q, y la señal resultante.....	21
Ilustración 5: Esquema de la trama PHY de IEEE 802.15.4.....	21
Ilustración 6: El formato de la trama MAC de IEEE 802.15.4.....	21
Ilustración 7: Ejemplo de red Kindergarten.....	26
Ilustración 8: Tama IEEE 802.15.4 conteniendo un mensaje Kindergarten....	26
Ilustración 9: Esquema temporal de funcionamiento de Kindergarten.....	28
Ilustración 10: División de una red en dos subredes con cobertura total.....	28
Ilustración 11: Línea de Heartbeat sincronizando las estaciones base y sus superframes.....	29
Ilustración 12: El formato del mensaje básico de Kindergarten.....	30
Ilustración 13: Formato del mensaje OOT de Kindergarten.....	30
Ilustración 14: Formato del mensaje POE de Kindergarten.....	30
Ilustración 15: Formato del mensaje JR de Kindergarten.....	31
Ilustración 16: Formato del mensaje JA de Kindergarten.....	31
Ilustración 17: Camino recorrido por un mensaje Kindergarten entre dos estaciones móviles.....	32
Ilustración 18: Operaciones utilizando la tabla de routing del Master Router.	33
Ilustración 19: Máquina de estados de una estación móvil Kindergarten.....	35
Ilustración 20: Máquina de estados de una estación base Kindergarten.....	36
Ilustración 21: Relación distancia-nivel de señal.....	38
Ilustración 22: Frente y dorso de una tarjeta SD.....	44
Ilustración 23: Distintos anchos de DAT en tarjetas SD.....	45
Ilustración 24: Líneas de tiempo de comandos/respuesta de SD.....	46
Ilustración 25: Formato de los comandos y respuestas del bus SD.....	46
Ilustración 26: Formato de los bloques de datos de SD.....	46
Ilustración 27: Mapa de memoria de una tarjeta SDIO.....	48
Ilustración 28: Formato de un comando SDIO.....	48
Ilustración 29: Estructura del comando CMD5.....	49
Ilustración 30: El formato de respuesta R4.....	49
Ilustración 31: El formato de respuesta R6.....	49
Ilustración 32: Estructura del comando CMD7.....	50
Ilustración 33: El formato de la respuesta R1.....	51
Ilustración 34: Parámetros del CMD52.....	51
Ilustración 35: El formato de la respuesta R5.....	51
Ilustración 36: Diagrama de bus SPI con dos dispositivos esclavos.....	54
Ilustración 37: Conexión de los puertos SPI para simular una interfaz SDIO.	54
Ilustración 38: Conexión entre el CC2431 y el ENC28J60.....	58
Ilustración 39: Evaluation board del chip CC2431.....	60
Ilustración 40: Battery board del chip CC2431.....	60
Ilustración 41: Programador de los evaluation boards del chip CC2431.....	60
Ilustración 42: Adaptador CC2431 - SDIO, junto a una placa Wi-Fi SDIO para comparación.....	61
Ilustración 43: Prototipo de estación móvil conectada a una PocketPC iPaq. .	61
Ilustración 44: Módulo ENC28J60-H.....	61
Ilustración 45: Diagrama esquemático del adaptador CC2431 - ENC28J60... .	62

Ilustración 46: Diagrama del circuito impreso del adaptador CC2431 - ENC28J60.....	62
Ilustración 47: Prototipo de estación base Kindergarten.....	62
Ilustración 48: Diagrama de la prueba de campo realizada con Kindergarten.	63

Agradecimientos

*“¡¡Gracias totales!!”
Gustavo Ceratti, Octubre de 2007*

Se cierra una etapa de mi vida, una etapa importante e irreplicable, por momentos difícil, por momentos feliz, por momentos triste, por momentos profunda y emotiva.

Por momentos, endemoniadamente frustrante.

Quiero aprovechar este pequeño espacio para agradecer a las personas que hicieron que este momento fuera posible, a las que compartieron conmigo el camino, y a las que me llevaron de la mano en los momentos en que mis fuerzas flaquearon.

En primer lugar, a mis padres, confiaron siempre en mí y me apoyaron incondicionalmente, en las buenas y en las malas, y fueron mi ejemplo a seguir y mis consejeros en todo momento.

A mi hermano Ismael, uno de esos tipos que siempre están.

A Gustavo y Silvia, a quienes admiro y aprecio sobre todo por ser las excelentes personas que son. Porque me permitieron entrar a un mundo nuevo, me hicieron crecer y me enseñaron de muchas más formas de las que ellos mismos se imaginan.

A Viviana, la personita que cambió mi vida para siempre.

A todos los compañeros de laboratorio de quienes me llevo los mejores recuerdos: Andrés, Cecilia, Alejandra, Julián, Lucas, Emiliano, Sebastián, Gabriela, Lautaro y Santiago.

A todas las personas con las que compartí estos años: Esteban, Rocío, Silvana, Marcos, Carina, Bea, Esteban, Esteban, Gastón, Juan, Andrés, Juan, Diego, Damián, Martín, Damián, Andrea, Marcelo, Matías, Mariano, Leonor, Lía, Lucho, Juan, Nicolás, Cali, Nicolás y Einar (sólo para que no se queje).

A todos los que me olvidé de incluir en la lista anterior.

A los profesores que cedieron las horas.

A Diego Torres, porque es un groso.

A Terry Pratchett.

A Douglas Adams. ¡Adiós, y gracias por el pescado!

Y muy especialmente... a todos los que dijeron que este día nunca iba a llegar.

Capítulo 1: Introducción

*“In the beginning the universe was created.
This made a lot of people very angry, and has been widely regarded as a bad move.”
Douglas Adams, “The hitch-hiker’s guide to the galaxy”*

La tecnología avanza constantemente a pasos agigantados a nuestro alrededor. Hace sólo diez años habría sido imposible hasta para los más poderosos gobiernos y las más grandes corporaciones contar con una infraestructura informática como la que hoy en día nos rodea y nos sigue hasta los lugares más recónditos del mundo.

En cualquier comercio de artículos electrónicos se pueden comprar teléfonos celulares capaces de procesar en tiempo real audio, video, aplicaciones y transferencias de datos a velocidades impensadas sólo cinco años atrás. Hoy mismo están poniéndose en marcha nuevos estándares de transferencia de datos inalámbricos que prometen aumentar en 10 veces las velocidades de los equipos de escritorio y portátiles con la norma IEEE 802.11n, y unas 100 veces la velocidad de los equipos móviles gracias a los nuevos protocolos celulares W-CDMA, HSDPA y HSUPA más conocidos como las tecnologías de tercera generación (“3G”)[1].

El poder de cómputo avanza, se abarata y se populariza. Los microprocesadores están en todos lados y prácticamente no existe ningún dispositivo electrónico que no utilice lógica programable, sea en forma de computadoras propiamente dichas como es el caso de las PDAs y computadoras portátiles, o microcontroladores: sistemas programables contenidos en un único chip que pueden dar comportamiento dinámico a cualquier equipo electrónico al que se conecten.

Estamos sumergidos en una red de tecnología e información, rodeados de dispositivos que sólo por el hecho de estar presentes cambian la forma en que pensamos y vivimos. Conectados unos a otros por los hilos invisibles de una telaraña de ondas electromagnéticas y los haces de luz multicolor que surcan los canales de fibras ópticas a través de ciudades, países y continentes.

Tal es la disponibilidad y el abaratamiento de la infraestructura de cómputo, que en los últimos años los grandes fabricantes de equipo informático (INTEL siendo su exponente más notorio) han comenzado a volcar sus principales líneas de investigación a la creación de tecnologías eficientes, de bajo consumo y bajo costo[2]. Ya quedaron atrás los años de la carrera por el dominio de los gigahertz de reloj y no es tan importante qué cantidad de gigabytes de memoria tenga una computadora. Hoy lo importante es la movilidad.

Es por esta razón que los sistemas móviles actuales están preparados para servir como plataforma de desarrollo para el siguiente paso en su evolución: los sistemas que no sólo responden a las órdenes del usuario sino que además son capaces de realizar acciones basados en características de su entorno. Son los llamados sistemas sensibles al contexto, o “context-aware”

Quien primero tuvo la idea de que los sistemas informáticos se beneficiarían de obtener información del entorno fue Bill N. Schilit, quien junto a Norman Adams y Roy Want publicó en 1994 el trabajo “Context-aware computer applications”[3]. Allí los autores describen software que examina el contexto de un usuario y reacciona a los cambios en él, definiendo “contexto” como la combinación de tres aspectos: quién es el usuario, dónde está y qué recursos existen cerca de él.

Dicho paper se centra en el desarrollo que Schilit y su equipo realizaron en los laboratorios de investigación de Xerox en Palo Alto, California, utilizando un dispositivo móvil muy rudimentario llamado ParcTab. La ParcTab era esencialmente una versión muy primitiva de lo que hoy conocemos como PDAs: tenía una pantalla táctil de 128x64 pixels, un parlante y un transceiver infrarrojo, que se comunicaba con transceivers similares ubicados en cada habitación. Las ParcTabs actuaban como terminales bobas de un software centralizado, que al saber desde qué habitación estaba cada terminal, podía inferir el contexto en el que se encontraba.

En 2001, Anind K. Dey publicó en el Instituto de Tecnología de Georgia su tesis de doctorado “Providing architectural support for building context aware applications”[4] donde amplía el concepto de contexto de una aplicación. Dey lo define como “Cualquier información que

pueda ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, un lugar o un objeto que es considerado relevante para la interacción entre el usuario y una aplicación, eventualmente incluyendo al usuario y la aplicación misma”. Como complemento luego Dey indica que algunos tipos de contexto son más importantes que otros, como por ejemplo la ubicación, la identidad del usuario, la hora y la actividad que se está llevando a cabo.

En 2001 Kirsi Varrassaus y otros[5] publicaron el primer trabajo donde se define el concepto de servicios basados en localización. En él, los autores dicen que “Los Servicios Basados en Localización (LBSs) son servicios de información accesibles con dispositivos móviles a través de la red inalámbrica y que hacen uso de la posición del dispositivo móvil”.

Una posición en el espacio puede representarse de múltiples maneras desde las simplistas posiciones X-Y en algún plano predeterminado, pasando por coordenadas geográficas sobre un datum específico, hasta sistemas jerárquicos que representen relaciones de inclusión de espacios y sistemas de conexiones y adyacencias entre ellos. Otros trabajos se han encargado de establecer un amplio campo de estudio acerca de estos temas, y no es el propósito de esta investigación el definir uno ad-hoc sino crear un soporte para la estimación de una posición que pueda ser luego interpretada en cualquier sistema que se necesite.

A través de los años se han utilizado para la determinación de la posición de un sistema informático móvil diversos enfoques. El primer intento perteneció a creador del concepto de context-awareness, Bill Schilit. Él utilizaba en una primitiva computadora de mano un conjunto de emisor/receptor de luz infrarroja, que se comunicaba con emisores/receptores similares ubicados en las distintas habitaciones. Sabiendo desde qué receptor se captaba señal de una computadora de mano en particular, el sistema era capaz de establecer la habitación donde se encontraba.

Con el advenimiento de las diferentes tecnologías de comunicación por ondas de radio, primero a través de métodos ad-hoc y luego con estándares como Bluetooth o Wi-Fi, otros proyectos[6] han intentado obtener resultados similares eliminando la necesidad de contar con una línea de vista ininterrumpida en todo momento entre emisor y receptor.

Los resultados de estos esfuerzos han sido mixtos. Ninguna de las tecnologías mencionadas anteriormente ha sido diseñada con la medición de posición como una de sus características, por lo que los métodos que las utilizan deben emplear algoritmos complejos y no siempre precisos. Tampoco tecnologías como Wi-Fi han sido creadas con miras a utilizarse por períodos prolongados (8 horas o más) en sistemas de mano, por lo que su consumo de energía es elevado y su autonomía en los casos de uso que competen al presente trabajo es sumamente reducida.

Descripción general del presente trabajo

En la presente tesina se abordan los mecanismos que permiten a los sistemas context-aware obtener su posición física dentro de un ambiente y comunicarse unos con otros. Si bien este último aspecto ha sido abordado por múltiples tecnologías existentes, como se mencionó anteriormente, el presente enfoque utiliza el mismo hardware de muy bajo consumo de energía para localización y comunicaciones, reduciendo el consumo de energía del sistema móvil y por ende aumentando su autonomía y practicidad.

Se presentará una combinación de hardware y software a la que se bautizó Kindergarten, un tipo de red de datos y localización diseñada y desarrollada íntegramente con esas dos características en mente, para dispositivos móviles de bajo consumo y alta autonomía.

En una red Kindergarten existen tres tipos de sistemas de procesamiento, cada uno de ellos con una función específica:

- **Estaciones móviles:** se les da este nombre a el conjunto de la PDA/Smartphone y el microcontrolador que opera como mecanismo de comunicación y estimación de posición que el usuario lleva consigo.
- **Estaciones base:** son los componentes estáticos de la red; se instalan en los ambientes cubiertos por la red Kindergarten en grupos de tres o más. Están conectados a una red Ethernet y proveen servicios de comunicación y localización a

las estaciones móviles.

- **Coordinador central:** es un único equipo, típicamente un PC estándar, conectado a la misma red Ethernet que las estaciones base y se encarga de recibir todos los mensajes Kindergarten emitidos por las estaciones (base y móviles) y redirigirlos a su destino. Actúa como un concentrador central de tráfico.

Organización del trabajo

El capítulo 1 realiza una introducción a los sistemas context-aware, los servicios basados en localización y la red Kindergarten.

El capítulo 2 presenta el estado del arte de los mecanismos de comunicaciones y sensado de posición para dispositivos inalámbricos.

El capítulo 3 describe resumidamente el motivo del presente trabajo, la red Kindergarten.

El capítulo 4 describe el hardware sobre el que se basó la red Kindergarten. Sus características físicas y operativas, y conceptos que se utilizarán en los capítulos subsiguientes.

El capítulo 5 aborda la organización lógica de la red Kindergarten. Se describe su principio de funcionamiento, las decisiones que se tomaron en su diseño y los mecanismos de comunicación y routing de mensajes entre nodos que cambian constantemente de posición.

El capítulo 6 habla del esquema de estimación de posición de Kindergarten, de su principio básico de funcionamiento y del trabajo futuro que puede hacerse en este aspecto.

El capítulo 7 explora la comunicación entre el hardware del nodo Kindergarten y la PDA o Smartphone al que están conectados. Se introduce el bus SD, del protocolo SDIO y de cómo se implementó este mecanismo sobre el hardware descrito en el capítulo 3.

El capítulo 8 está dedicado a la comunicación entre estaciones base y la red cableada mediante la cual están conectadas entre sí. Se habla del hardware y software que permiten que las estaciones base envíen y reciban mensajes unas de otras.

El capítulo 9 describe una red Kindergarten real creada con el fin de poner a prueba el sistema descrito en los capítulos anteriores. Se describen los circuitos reales creados, las pruebas realizadas y los resultados obtenidos.

El capítulo 10 presenta las conclusiones y se discuten posibles trabajos futuros al respecto.

Capítulo 2

Estado del arte

En este capítulo se habla del estado actual del campo de los sistemas context-aware en el mundo y de los distintos mecanismos de sensado de posición que se evaluaron al momento de comenzar con el presente trabajo.

Durante la última década, las aplicaciones móviles sensibles al contexto (context-aware) han ido ganando importancia y filtrándose lentamente en nuestra vida cotidiana. De forma casi inconsciente estamos empezando a incorporar tecnologías, dispositivos y aplicaciones que eran impensables tan solo diez años atrás. La penetración de este tipo de aplicaciones en la sociedad se da principalmente gracias a mejoras en las áreas de hardware y comunicaciones.

Sin embargo, aún estamos lejos de haber resuelto todos los problemas en estas áreas. Aún cuando hay diferentes dispositivos de hardware y protocolos para comunicación inalámbrica y estimación de posición, todavía necesitamos uno que sea lo suficientemente pequeño para ser integrado en una gama amplia de dispositivos, no sólo PDAs y Smartphones, sino también en llaveros y credenciales. Este dispositivo debe tener muy bajo requerimiento de energía de manera de no acortar significativamente la carga de la batería. La mayoría de los estándares de comunicación inalámbrica han logrado alcanzar el objetivo del tamaño, y son efectivamente utilizados en dispositivos móviles. Sin embargo el bajo consumo es aún hoy un problema que no ha podido resolverse, lo que nos lleva a la necesidad de concebir un nuevo tipo de hardware.

Durante la presente investigación se han probado tecnologías existentes susceptibles de ser utilizadas en el proyecto, incluyendo Wi-Fi [7] y Bluetooth[8] las cuales, a pesar de su presencia casi ubicua en todas las PDAs y Smartphones actuales, fueron descartadas por su alto consumo de energía. También se descartó el sistema GPS[9], cuya capacidad de determinación de posición sólo funciona en exteriores haciéndolo inutilizable dentro de edificios.

Desafortunadamente se arribó a la conclusión de que no existe un dispositivo de hardware estándar que pueda ser añadido a cualquier sistema móvil y pueda ser usado tanto para comunicación como para estimación de posición de manera práctica. Por esta razón se decidió construir una plataforma propia de hardware y software para comunicar y posicionar dispositivos móviles de forma eficiente.

Trabajos previos

Con el fin de entender las necesidades para nuevo hardware y arquitecturas de sensores, deben considerarse los enfoques anteriores y analizar sus puntos débiles y fuertes. En los próximos párrafos se hablará de algunos de los mecanismos más generalizados de comunicación y localización disponibles en la actualidad. En particular se describirán los sistemas infrarrojos, Bluetooth, Wi-Fi y la tecnología GPS y se examinarán sus características desde el punto de vista de una aplicación context-aware.

Puerto infrarrojo

Tal vez el dispositivo más básico de hardware es el puerto de infrarrojos (IR), que es principalmente considerada como una interfaz de comunicaciones para pequeños dispositivos como PDAs o teléfonos celulares[10]. En su forma más pura, tal como se define por la asociación IrDA, un puerto de infrarrojos envía y recibe datos entre dos dispositivos codificándolos como una serie de pulsos de infrarrojos. Cada dispositivo tiene un pequeño emisor de infrarrojos (un LED infrarrojo) para enviar los impulsos de luz a los demás y un dispositivo detector de infrarrojos para recibir a los enviados a él.

Este mecanismo deja al stack de software la responsabilidad de ensamblar y desensamblar las estructuras de datos de más alto nivel para permitir a las aplicaciones en ambos lados de la conexión de infrarrojos intercambiar datos de manera fiable y estructurada.

La falta de sofisticación en los niveles más bajos del mecanismo de comunicaciones puede ser una ventaja para adaptar este puerto para servir a otras funciones. En particular, podemos hacer que actúe no como un puerto de comunicaciones, sino como un dispositivo de detección. Para ello, un emisor de infrarrojos está configurado para enviar continuamente un flujo de datos infrarrojos, por ejemplo un identificador numérico o un identificador de posición. Estos dispositivos se pueden fijar a determinados objetos o lugares en el medio ambiente para transmitir sus datos a cualquier receptor de infrarrojos de un dispositivo de mano en la cercanía, actuando efectivamente como una baliza de proximidad. Cuando un usuario llevando uno de estos dispositivos receptores de infrarrojos (que podría ser un Smartphone, una PDA o un dispositivo diseñado específicamente) se acerca a la baliza, el software se ejecuta en el dispositivo será capaz de decodificar la información presente en la baliza y deducir su posición.

El enfoque de utilizar luz infrarroja para etiquetar posiciones en el espacio tiene todas las ventajas y desventajas intrínsecas del medio de transmisión. Una ventaja importante es su alta precisión espacial: ya que el haz infrarrojo puede ser enfocado a cubrir un área específica, sólo los dispositivos dentro de este área podrán recibir la señal IR y por tanto interpretar los datos que lleva.

Como un aspecto negativo puede citarse que el único evento que este tipo de mecanismo de sensado de posición es capaz de detectar es la aproximación de la computadora de mano (receptor) al dispositivo emisor de la baliza. Un segundo evento, el cual puede estar relacionado con el usuario quitando el dispositivo de mano o alejándose del emisor de la baliza, puede inferirse cuando el receptor deja de detectar la señal.

En resumen, un sistema de localización como el descrito es una buena opción cuando hay una necesidad de crear una etiqueta de localización precisa, ya que proporciona un tiempo de respuesta aceptable y relativamente buena resolución espacial. No puede, sin embargo, utilizarse para detectar la posición del dispositivo de mano (y por lo tanto la posición de su usuario) en zonas amplias, debido a su campo de visión estrecho, o en lugares en los que la baliza no se puede recibir.

Por último, otra desventaja de una solución empleando balizas de infrarrojos es su ancho de banda limitado cuando se utiliza para transferir datos. La especificación estándar IrDA llega sólo hasta 115,2 Kbps, y aunque existe un estándar que soporta velocidades de hasta 4 Mbps, no es soportado por todos los dispositivos.

Bluetooth

Bluetooth es un mecanismo de comunicaciones que sólo en años recientes ha encontrado su lugar en el mercado masivo. Desarrollado en 1999 como un enlace inalámbrico entre dispositivos móviles de bajo consumo, su objetivo principal es proveer servicios de transmisión de datos confiables sin reducir significativamente la vida de las baterías. Aún cuando el ancho de banda alcanzado por Bluetooth no es tan alto como para competir con otros métodos de comunicación como Wi-Fi, esto no es un problema para aplicaciones donde no se necesita transferir cantidades importantes de datos como es el caso en aplicaciones context-aware.

De la misma forma que con la tecnología infrarroja, Bluetooth también puede usarse para estimar la posición de un dispositivo basándose en otros dispositivos similares actuando como balizas para marcar una posición específica. El uso de señales de radio es mejor que el de señales de luz por ser un mecanismo inmune a interferencias por la sola presencia de objetos en el camino entre transmisor y receptor.

Para aplicar este método se debe programar una computadora de escritorio o un transmisor autónomo para transmitir continuamente su identificación Bluetooth que será recibida por el dispositivo móvil. El dispositivo móvil traduce entonces el identificador de la baliza a su posición física de manera similar a la forma con balizas infrarrojas.

Una segunda solución para inferir la posición del usuario es usar un mecanismo de triangulación. En este procedimiento, la intensidad relativa de la señal de una trama recibida tiene una correlación directa con la distancia del dispositivo móvil al dispositivo que envía la trama.

Cuando funciona como mecanismo de comunicaciones, Bluetooth funciona como una red master-slave. Un dispositivo Bluetooth actuando como master puede conectarse con hasta 7 dispositivos esclavos para transferir datos y un dispositivo esclavo puede conectarse con sólo un master. Esto hace imposible que un dispositivo móvil mantenga una transferencia de datos mientras se desplaza, pues para que un dispositivo esclavo (móvil) pueda conectarse a un nuevo dispositivo Bluetooth master primero debe desconectarse del anterior, haciendo que todas las conexiones se interrumpan. Este problema no puede ser salvado porque la naturaleza cerrada del stack de Bluetooth hace imposible acceder directamente al hardware de manera portable entre distintas implementaciones de hardware.

En las pruebas que se realizaron se encontró que el proceso de detectar una baliza Bluetooth es lento, requiriendo varios segundos para completarse. De la misma manera, la API expuesta más comúnmente a los programadores de Bluetooth sólo permite detectar si una cierta baliza está presente o no. Algunos vendedores de stacks de Bluetooth (en particular la empresa IVT, creadores del stack Blue Soleil) proveen llamadas ad-hoc para obtener la intensidad relativa de señal de una trama recibida, que como se ha explicado antes puede relacionarse con la distancia entre emisor y receptor. Por desgracia esta característica no es estándar en todas las implementaciones de stacks Bluetooth, y mucho menos en dispositivos móviles como PDAs.

Como efecto secundario, el tener un adaptador Bluetooth constantemente realizando una tarea de scanning de la red hace que no esté disponible para comunicaciones de datos, y además reduce significativamente la vida de la batería del dispositivo móvil.

Wi-Fi (standard IEEE 802.11b/g)

Wi-Fi es estándar desarrollado inicialmente para ser utilizado en computadoras portátiles. Con velocidades de transferencia de hasta 54 Mbps, ninguno de los estándares de hardware de comunicaciones móviles se puede comparar con la velocidad de IEEE 802.11g, haciéndolo una herramienta de comunicación inalámbrica excelente.

De la misma manera que con Bluetooth, se puede crear un mecanismo basado en Wi-Fi para inferir la presencia de un dispositivo en el área de cobertura de un access point detectando las balizas que este emite a intervalos regulares. Además, la mayoría de los adaptadores wireless tienen la capacidad de detectar y reportar la intensidad relativa de señal de las balizas de los diferentes access points que detectan. Como esto puede realizarse de forma portable y relativamente standard, permite estimar la distancia entre un dispositivo móvil y diversos access points, y por ende estimar su posición.

Sin embargo, aún habiendo sido diseñados con características de ahorro de energía en mente, el consumo de corriente de los adaptadores Wi-Fi es demasiado alto para poder ser utilizado durante más de unos pocos minutos en dispositivos altamente portables, como smartphones y PDAs. Con un adaptador Wi-Fi activo, la duración de batería de una PDA típica se reduce a menos de una hora y media, algo inaceptable para aplicaciones context-aware del mundo real donde se desean autonomías de al menos 8 horas.

GPS

Finalmente, aún cuando no puede ser utilizado como medio de comunicación, debemos considerar el Sistema de Posicionamiento Global (GPS), que es el estándar de facto para sistemas de medición de localización. Incluso siendo la única opción confiable cuando se necesita obtener la localización espacial de un sistema, tiene la desventaja de sólo funcionar en exteriores, haciéndolo inapropiado para las necesidades antes descritas.

Por todos los motivos expuestos se decidió descartar las tecnologías existentes en favor de una más adecuada para su aplicación en sistemas context-aware.

Capítulo 3

Descripción resumida de La red Kindergarten

En este capítulo se introduce en forma resumida el tema a tratar en el resto de la presente tesis de grado: la red Kindergarten en todos sus aspectos; desde el hardware en el que corre hasta el formato de los mensajes cursados entre estaciones.

En el capítulo 1 se presentó el problema de los sistemas sensibles al contexto, se describió sus posibilidades de aplicación y sus necesidades de información proveniente del mundo real.

En el capítulo anterior se ha hecho una reseña de los mecanismos para comunicación y determinación de la posición de un dispositivo móvil existentes actualmente: se describieron los mecanismos de transmisores/receptores infrarrojos, descartados por su baja precisión, los métodos de Wi-Fi y Bluetooth, que se dejaron de lado por su alto consumo y baja velocidad de actualización, y el GPS abandonado por su incapacidad de funcionar en interiores.

En base a lo expuesto anteriormente se elaboró una serie de requerimientos para un nuevo tipo de red que se bautizó Kindergarten:

- Permitir pasaje de mensajes.
- Soportar estimación de posición.
- Todas las operaciones deberían completarse en tiempo determinístico.
- Eficiente para aprovechar mejor el ancho de banda del canal de datos.
- Simple para implementarse sobre procesadores sencillos y de poco consumo.
- Poseer dos tipos de estaciones: estaciones base, similares en función a los puntos de acceso Wi-Fi, y estaciones móviles, que el usuario lleva consigo y donde se ejecuta el software context-aware.

Se eligió el microcontrolador CC2431 de la empresa Texas Instruments para servir como hardware sobre el que realizar la implementación de Kindergarten, basando la elección principalmente en su bajo consumo e integración de un modem de radio y un módulo de estimación de posición en el mismo chip. El CC2431 se utiliza tanto en las estaciones móviles como en las estaciones base, ejecutando distintas bases de código que dan los comportamientos de los dos tipos de estaciones.

Para la creación de Kindergarten se tomaron características deseables de múltiples redes existentes, tales como Ethernet, Token Ring, Wi-Fi y Aloha ranurado, para obtener un nuevo tipo de capa de enlace que garantizara los requerimientos enumerados arriba.

El protocolo divide el tiempo en slots discretos (frames) que se agrupan en bloques llamados superframes; cada superframe se asigna a una estación base, que sólo puede enviar y recibir información durante su superframe. Cada transferencia de información se hace en un slot de tiempo, y una estación móvil sólo puede transmitir cuando una estación base se lo ordena. De esta forma, se obtiene una red completamente libre de colisiones que maximiza la transferencia de información.

Las estaciones base emiten en cada uno de sus superframes un mensaje especial donde señalan su posición física; las estaciones móviles reciben este mensaje y a partir de él triangulan su posición utilizando la intensidad de la señal recibida como indicador de la distancia entre estación base (emisor) y estación móvil (receptor). Estos datos alimentan al módulo de estimación de posición del chip CC2431.

Las estaciones móviles Kindergarten basan su capacidad de localización en la recepción de los mensajes de todas las estaciones base que utilizan como referencia. Si el área cubierta por una red Kindergarten es lo suficientemente amplia, las estaciones móviles podrían quedar fuera del alcance del modem de radio de una o más estaciones base. En este caso, la red Kindergarten se subdivide en dos o más sub-redes, todas sincronizadas entre sí de manera de no interferirse.

Además de las estaciones base y móviles se creó un tercer tipo de sistema conectado a la red Kindergarten: un coordinador central que se encarga de monitorear qué estaciones móviles se

comunican a través de cuál estación base, y de realizar el routing de mensajes de una a otra. Este sistema puede ser simplemente una PC de escritorio.

La comunicación entre el microcontrolador que corre el stack de Kindergarten y la PDA o Smartphone que aloja la aplicación context-aware se realiza a través del bus SDIO. Este es un derivado del bus SD utilizado en muchos dispositivos portátiles como medio de comunicación para tarjetas de memoria extraíbles. SDIO amplía las capacidades de SD y le permite controlar y comunicar dispositivos de entrada/salida y lógica programada.

A través de SDIO el sistema context-aware puede enviar al stack de Kindergarten comandos o información, y ser notificado de nuevos mensajes que arriben de otras estaciones. Para ello se creó un stack de software que aprovecha los dos puertos seriales síncronos (SPI) presentes en el chip CC2431 para actuar como un emulador de tarjeta SDIO, permitiendo que el sistema host (PDA o Smartphone) lo detecte y cargue el controlador correspondiente dentro de su sistema operativo.

La comunicación entre estaciones base, por su parte, se realiza a través de la red Ethernet utilizando un controlador ENC28J60 de la empresa Microchip. Este controlador integra en un solo chip de silicio un controlador MAC y PHY, más toda la lógica necesaria para enviar y recibir tramas Ethernet sin hardware adicional. Se controla a través de una interfaz SPI cableada al CC2431 de la estación base.

Se utilizó un stack TCP/IP open source llamado μ IP para comunicar la estación base a través del protocolo TCP/IP con el coordinador de la red. μ IP es un stack totalmente compatible con los estándares de IP diseñado para ejecutarse en procesadores de gama baja y microcontroladores. El pasaje de mensajes se realiza a través del protocolo UDP. Como parte del proyecto se diseñaron y fabricaron placas de circuito impreso especiales para alojar el conexionado entre los distintos módulos de las estaciones base.

Para poner a prueba el concepto de Kindergarten, se implementó y construyó una pequeña red de prueba de cinco estaciones (cuatro estaciones base y una móvil) y se realizaron mediciones de precisión del mecanismo de localización. Resta aún por verse al momento de la escritura de este documento cuál es el rendimiento de transferencia de datos entre estaciones.

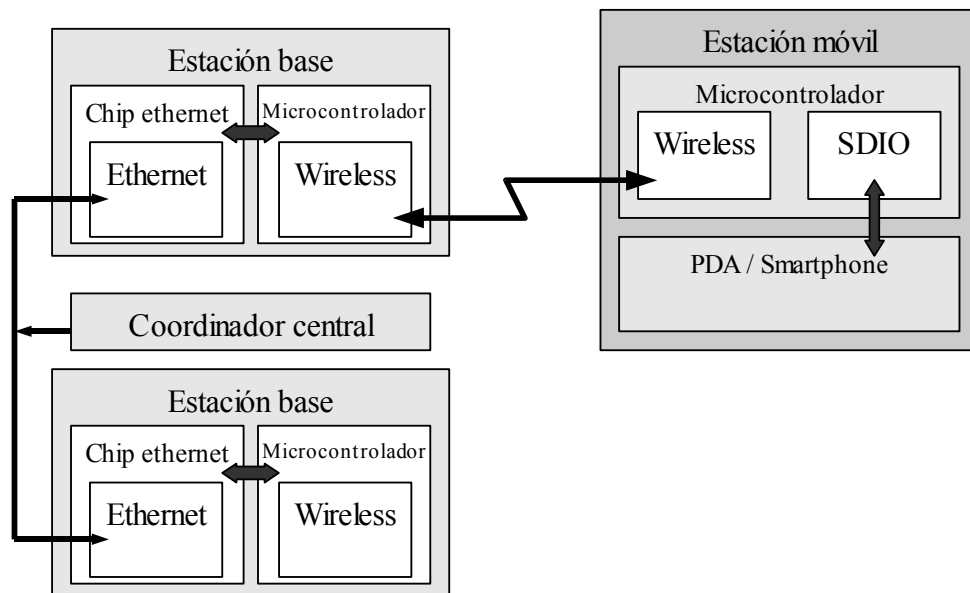


Ilustración 1: Esquema general de los integrantes de una red Kindergarten

Capítulo 4

El hardware de la red Kindergarten

En este capítulo se describirá el hardware sobre el que corre el protocolo Kindergarten, las decisiones que se tomaron para su elección y los diseños de los circuitos de soporte.

Una vez que se evaluaron las alternativas propuestas en el capítulo anterior quedó claro que ninguna de ellas podía satisfacer plenamente los requerimientos de consumo, comunicación, estimación de posición y flexibilidad. Por ello se comenzó a buscar hardware sobre el que crear una solución específica.

Tomada la decisión de ir con una solución de hardware distinta a los dispositivos tradicionales surgió la necesidad de decidir qué tan independiente sería este circuito de la PDA o Smartphone al que estaría conectado. Mientras que una PDA es una plataforma excelente para correr programas de aplicación, está diseñada para optimizar la experiencia del usuario; no para realizar operaciones de transmisión y recepción de datos en tiempo real donde los plazos de reacción se miden en microsegundos.

En este momento se tomó la decisión de buscar un componente que además de ser transmisor/receptor de radio pudiera realizar al menos parte del procesamiento del protocolo que se desarrollaría mediante la técnica conocida como offloading de procesamiento. El offloading es sencillamente la separación de parte del algoritmo de un programa, en este caso el stack de red, y adaptarlo para correr en un elemento de cómputo separado de la CPU principal.

Esta técnica se desarrolló en principios de la década de 1990 para descargar de la CPU de los grandes servidores de red[11] el procesamiento del armado, mantenimiento y desarmado de conexiones TCP/IP de alta velocidad y delegarlos en controladores de red especializados.

Microcontroladores

En el resto esta sección se describirá la electrónica elegida para el procesamiento del protocolo Kindergarten, de manera que resulta útil hacer hincapié en la familia de circuitos integrados que forma el corazón de dicho hardware: los microcontroladores.

Tradicionalmente las computadoras, incluso las de mano, son fabricadas con la combinación de múltiples partes o componentes, por lo general provenientes de diferentes fabricantes. Así, existen empresas dedicadas principalmente a la fabricación de microprocesadores como Intel o AMD, otras empresas cuya producción es memorias de semiconductores (Hynix) o almacenamiento magnético (Seagate). Para crear la computadora funcional, un fabricante debe conectar todas las diferentes piezas entre sí, a través de circuitos impresos o cables según sea necesario.

Pero existen casos en los que se debe incorporar lógica programable que por su baja complejidad, o grandes restricciones de espacio, costo o consumo hacen impráctica la utilización de componentes discretos como en los sistemas tradicionales. Para ellos es que existe el microcontrolador.



Ilustración 2: Microcontrolador

En contraste con un microprocesador tradicional, que sólo contiene una CPU, el microcontrolador integra además de las funciones aritmético-lógicas tradicionales elementos de almacenamiento de datos como memoria RAM para datos, memoria de sólo lectura para almacenar el programa, periféricos e interfaces de entrada/salida.

En comparación con los microprocesadores tradicionales, los microcontroladores operan a velocidades relativamente bajas (típicamente menores a 20 MHz), pero que resultan más que suficientes para las tareas que están diseñados para realizar. Su baja velocidad y alta integración les permite requerir poca energía, miliwatts o incluso microwatts[12] y suelen incluir para ello modos muy avanzados de bajo consumo que incluso permiten detener el reloj de CPU cuando el programa está a la espera de un evento externo, reanudando el procesamiento luego como si nada hubiera ocurrido.

A diferencia de CPUs de propósito general, los microcontroladores pueden no poseer un bus de direcciones o datos externos, al integrar todas las memorias en el mismo chip de silicio que la CPU. Esto permite crear componentes con menos pines (conexiones externas) y contribuye a abaratar el costo de producción de los sistemas que los utilizan.

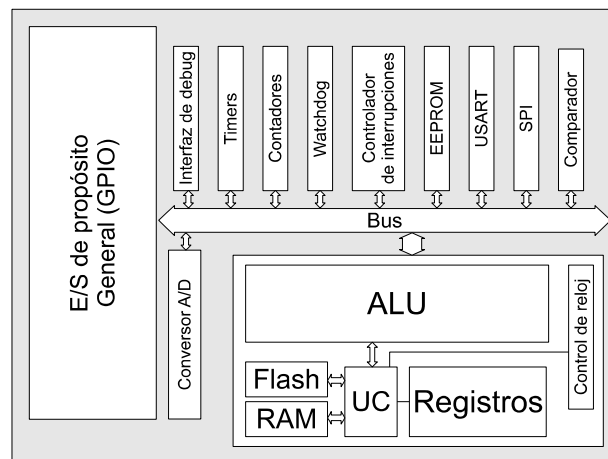


Ilustración 3: Diagrama de bloques de un microcontrolador típico

Un microcontrolador típico posee las siguientes características:

- Unidad central de proceso, que puede variar desde unidades aritmético-lógicas de 4 bits hasta procesadores de 32 bits con capacidad de manejo de memoria virtual (dependiendo del modelo de microcontrolador).
- Memoria volátil (RAM) para almacenamiento de datos.
- Memoria no volátil (Flash, EPROM, EEPROM, PROM) para almacenamiento del programa.
- Periféricos de entrada/salida digital, que permiten controlar y leer desde software el estado lógico individual de un conjunto de conexiones externas al chip.
- Puertos de entrada/salida serie, que pueden ir desde simples UARTs o USARTs hasta buses especializados como I²C, SPI o CAN para interconexión con otros sistemas similares.
- Periféricos tales como relojes, contadores, generadores de PWM y watchdogs.
- Circuito generador de reloj, generalmente basado en un cristal de cuarzo o un resonador del tipo resistor-capacitor (RC)
- Puede incluir conversores analógico/digitales, según el modelo.
- Soporte para programación in-circuit (chip ya instalado en el dispositivo) y debugging in-circuit.

Muchos microcontroladores aprovechan la diferencia de soporte entre memoria de datos y memoria de programas implementando una arquitectura Harvard: buses separados para ambos tipos de memoria, permitiendo el acceso simultáneo a ambas y aumentando el rendimiento. Esto también permite tener longitudes de palabra distintas entre ellas y de hecho es común entre los microcontroladores de 8 y 4 bits el tener longitudes de instrucción de 12 y 14 bits.

Las arquitecturas de los microcontroladores varían enormemente. Algunos diseños incluyen núcleos de procesador de propósito general con más o menos memoria interna y funciones de

entrada/salida integradas. Otros diseños se construyen específicamente para aplicaciones de procesamiento de señales, e incluyen coprocesadores matemáticos, o módulos que realizan operaciones numéricas específicas.

La mayoría de los microcontroladores posee instrucciones de manipulación de bits para hacer el control de programa más sencillo. Por ejemplo, un microprocesador de propósito general podría requerir de varias instrucciones para hacer un salto condicional basado en el valor de un bit determinado en cierto registro interno de la CPU, mientras que es común que los microcontroladores tengan instrucciones que realizan específicamente esa tarea.

Como consecuencia de su diseño todo-en-uno, no existe una forma de personalizar o adaptar un modelo de microcontrolador una vez que ya ha sido fabricado, de manera que las empresas que los desarrollan por lo general fabrican familias enteras de chips donde cada uno de los miembros de la familia se diferencia por la inclusión o ausencia de algún periférico, por características de la CPU, o por el tamaño de alguna de las memorias.

Como ejemplo, la tabla siguiente describe sólo una pequeña fracción de una de las familias de microcontroladores más populares[13], la línea AtMega de la empresa Atmel:

Modelo	Características
Atmega8	8KB de memoria Flash 1KB de memoria RAM 23 pines para E/S 8 conversores A/D 3 timers
Atmega64	64KB de memoria Flash 4 KB de memoria RAM 54 pines para E/S 8 conversores A/D 4 timers
AtxMega128A1	128 KB de Flash 8 KB de RAM 4 canales de DMA 78 pines para E/S 4 conversores D/A

Tabla 1: Algunos de los integrantes de la familia AtMega de Atmel

La selección del hardware

El hardware sobre el que corre Kindergarten es el microcontrolador CC2431 fabricado por la empresa Texas Instruments. Este microcontrolador integra un núcleo de procesamiento derivado del microcontrolador 8051 (un estándar de la industria desde hace casi tres décadas[14]), 128 kilobytes de memoria flash para el almacenamiento de programas y 8 kilobytes de memoria RAM estática. El CC2431 integra, entre otros dispositivos, dos interfaces USARTs y un módem de radio para la norma IEEE 802.15.4 en un chip de sólo 7 mm² de superficie.

La tabla completa de sus características es:[15]

- Núcleo de procesador 8051 de alto rendimiento y bajo consumo.
- Transceiver de radio compatible con la norma IEEE 802.15.4.

- 128 KB de memoria Flash programable in-circuit.
- 8 KB RAM.
- Controlador de DMA.
- Módulo de estimación de posición basado en señales de radio.
- Bajo consumo de energía (RX: 27mA, TX: 27mA, corriendo a 32 MHz).
- Soporte de hardware para CSMA/CA.
- Monitor de batería y sensor de temperatura.
- Conversor A/D de hasta 8 entradas de resolución configurable.
- Coprocesador de seguridad para AES de 128-bits.
- Dos USARTs con soporte para múltiples protocolos seriales.
- Soporte de debugging en el hardware.
- Watchdog por hardware.
- Cuatro timers, dos de 8 bits y dos de 16 bits.
- 21 pines de E/S de propósito general.
- Empaquetado QLP de 48 pines de 7mm²

La elección de este chip en particular se debió a la integración de los módulos de modem de radio y de estimación de posición mencionados en la lista de características. En las páginas que siguen se hará una descripción detallada de los mismos.

Acerca de la Entrada/Salida en microcontroladores

La alta tasa de integración del CC2431 y su gran cantidad de periféricos entran en conflicto con el tamaño pequeño del encapsulado físico y la necesidad de mantener bajos los costos de fabricación: si bien fabricar el “chip” de silicio que forma el microcontrolador es relativamente barato, cada conexión externa que se hace a una pata del encapsulado añade un costo adicional a la fabricación del componente en su conjunto y complica el diseño del circuito impreso sobre el que el microcontrolador estará soldado.

Estas razones llevan a los fabricantes de microcontroladores a combinar múltiples funciones lógicas en cada pin físico del encapsulado del chip, y proporcionar al programador medios para seleccionar qué unidad lógica o periférico será el que use cada una de las conexiones compartidas.

En el caso del CC2431, existen tres puertos de entrada/salida (GPIO, General-Purpose Input/Output) de 8 bits que se denominan Port0, Port1 y Port2, y comparten sus pins con los módulos conversores A/D, puertos seriales, contadores y controlador de interrupciones. Registros especiales dentro del espacio de memoria de programa permiten asignar cada uno de los pines compartidos a la función que se desee. En el caso de los puertos de entrada/salida, también es posible programar cada pin independientemente como línea de entrada, salida o simplemente desconectarlo eléctricamente de la conexión externa. Esta característica será abordada nuevamente cuando se describa la implementación del bus SDIO.

Otra característica importante del módulo de GPIO del CC2431 es su controlador de interrupciones, que permite programar independientemente cada uno de los bits de los 3 ports de GPIO como una fuente de interrupción externa. Este hecho también será aprovechado en el módulo de SDIO.

La norma IEEE 802.15.4

El estándar IEEE 802.15.4 describe las dos capas inferiores de un tipo de red inalámbrica de área personal (WPAN) enfocada hacia la comunicación ubicua de bajo costo entre dispositivos[16]. Esto la diferencia de redes como Wi-Fi, que pone el énfasis en la experiencia

de usuario. El objetivo de IEEE 802.15.4 es la comunicación de dispositivos cercanos con un bajo costo y poca o ninguna infraestructura adicional.

La especificación original describe un mecanismo capaz de cubrir un área de 10 metros de radio con una velocidad de transferencia de 250 kilobits por segundo. También provee la posibilidad, para dispositivos con requerimientos de consumo aún más bajos, de varias capas físicas alternativas: la especificación original preveía velocidades posibles de 20 y 40 kilobits por segundo, y la revisión actual de la norma agrega una opción para 100 kilobits por segundo. Esta posibilidad busca cumplir el objetivo de 802.15.4 dentro del conjunto de WPANs: costos de fabricación y operación extremadamente bajos sin sacrificar generalidad ni flexibilidad. Otras características importantes incluyen la posibilidad de uso en aplicaciones de tiempo real, a través de el uso de reservación de time slots, prevención de colisiones a través de CSMA/CA y soporte para comunicaciones seguras. Los dispositivos que implementan IEEE 802.15.4 suelen incorporar funciones de manejo de energía como medición de la calidad del enlace y del nivel de energía recibida. La definición de las capas de la red está basada en el modelo OSI, aunque sólo las capas inferiores están definidas en el estándar. La interacción con capas superiores puede implementarse como derivados de la subcapa de manejo de enlace de IEEE 802.2.

La capa física (PHY) provee el servicio de transmisión de datos además de la interfaz con la entidad de administración de la capa, encargada de realizar la selección de canales y las funciones de administración de energía y señal. Dependiendo de la implementación de la PHY, el dispositivo puede operar en una de las siguientes tres bandas:

Frecuencia	Velocidad	Canales
868 - 868,8 MHz	20, 40, 100 Kb/s	1
902 - 928 MHz	20, 40, 100 Kb/s	30
2400 - 2483,5 MHz	250 Kb/s	16

Tabla 2: Bandas de frecuencias para IEEE 802.15.4

IEEE 802.15.4 utiliza el método de modulación spread-spectrum de secuencia directa (DSSS¹[17]), donde las bandas de 868 y 902 MHz pueden utilizar desplazamiento binario o modulación en cuadratura de fase desplazada (OQPSK²[18]). La banda de 2400 MHz utiliza exclusivamente QPSK³[18]. Este método de modulación será abordado más adelante por ser el utilizado en el chip CC2431. En Agosto de 2007, la IEEE publicó la norma 802.15.4a que amplía los métodos de modulación agregando spread-spectrum de banda ultra-ancha de secuencia directa (DSSSUWB⁴[19]) y otro usando Chip Spread Spectrum (CSS[20]). La PHY UWB tiene asignadas frecuencias en tres bandas ISM⁵: por debajo de 1GHz, entre 3 y 5 GHz y entre 6 y 10 GHz. La PHY CSS tiene espectro asignado en la banda de 2450 MHz.

El estándar no define ninguna otra capa de nivel superior ni de interoperabilidad. Existen especificaciones, como ZigBee[21], que están construidas utilizando IEEE 802.15.4 como capas física y de enlace. Sin embargo, estos estándares de capas superiores no están relacionados con Kindergarten, y no se abordarán en el presente trabajo.

La modulación de IEEE 802.15.4

El entendimiento del método de modulación utilizado por IEEE 802.15.4 es importante a los efectos del presente trabajo porque permite entender mejor el mecanismo por el cual se determina el nivel de señal recibida en cada una de las tramas que llegan a cada uno de los

-
- 1 Del inglés Direct-Sequence Spread Spectrum
 - 2 Del inglés Offset Quadrature Phase-Shift Keying
 - 3 Del inglés Quadrature Phase-Shift Keying
 - 4 De DSSS Ultra WideBand
 - 5 Las bandas de frecuencia ISM (Industrial, Scientific and Medical) son grupos de frecuencias de radio que pueden utilizarse sin necesidad de solicitar una licencia del Estado.

nodos de la red, y es precisamente este nivel de señal lo que se utiliza como base para la estimación de la posición física del nodo, en caso de ser el nodo una estación móvil. En esta sección se abordará el método de OQPSK utilizado en la banda de 2400 MHz por ser el que utiliza el chip CC2431. El resto de los métodos de modulación están descriptos en la misma documentación del estándar, obtenible en [16].

El Spread-Spectrum de Secuencia Directa (DSSS)

Las técnicas de spread-spectrum son métodos mediante los que la energía generada en una banda de frecuencias se esparce deliberadamente sobre todo el dominio de la frecuencia, resultando en una señal con un ancho de banda mucho mayor al de la información original. Estas técnicas se usan por razones como el establecimiento de comunicaciones seguras, aumentar la resistencia de la señal contra interferencias, o evitar su detección. Las transmisiones de spread-spectrum por secuencia directa multiplican los datos a ser transmitidos con una señal de “ruido” que es una secuencia pseudoaleatoria de valores 1 y -1, a una frecuencia mucho más alta que la de la señal original.

Los valores binarios obtenidos del producto de la información original con la señal de ruido se conocen con el nombre de “chips”, y son los que modulan la portadora que finalmente se transmite. La señal resultante se parece mucho al ruido blanco, pero esta señal puede ser usada para reconstruir exactamente la señal original en el receptor, al multiplicarla por la misma secuencia pseudoaleatoria. Este proceso se conoce como de-spreading.

En el caso de IEEE 802.15.4, la información a transmitir se divide en símbolos de 4 bits, y cada una de los 16 símbolos posibles se transforma a chips usando una secuencia definida en el mismo estándar.

Símbolo	Chips
0	01011001110000110101001000101110
1	11101101100111000011010100100010
2	00101110110110011100001101010010
3	00100010111011011001110000110101
4	01010010001011101101100111000011
5	00110101001000101110110110011100
6	11000011010100100010111011011001
7	10011100001101010010001011101101
8	10001100100101100000011101111011
9	10111000110010010110000001110111
10	01111011100011001001011000000111
11	01110111101110001100100101100000
12	00000111011110111000110010010110
13	01100000011101111011100011001001
14	10010110000001110111101110001100

15	11001001011000000111011110111000
----	----------------------------------

Tabla 3: Secuencia de chips para cada uno de los 16 símbolos de IEEE 802.15.4

Los 32 chips que forman un símbolo se dividen en dos secuencias de 16 chips cada uno; cada una de estas secuencias modula sendas portadoras, llamadas I y Q, desfasadas 90° entre sí:

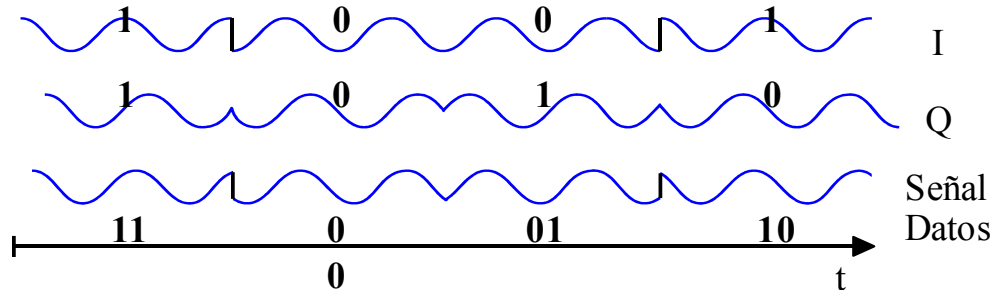


Ilustración 4: Las portadoras I y Q, y la señal resultante

La trama PHY de 802.15.4

La trama generada por la capa física de 802.15.4 se inicia con un grupo de 5 bytes que forman en su conjunto una secuencia única de bits pensada para que el receptor de la trama se sincronice con el transmisor. El estándar los define como un preámbulo de 4 bytes de valor 0, y un byte llamado Delimitador de Inicio de Trama (SFD) que lleva el valor 167 (A7 hexadecimal).

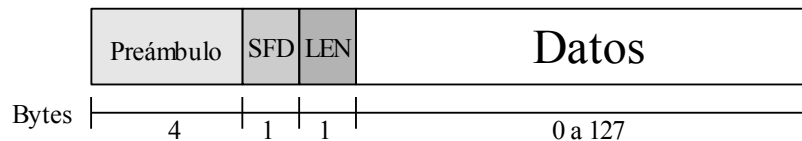


Ilustración 5: Esquema de la trama PHY de IEEE 802.15.4

Seguido a esta cabecera de sincronización se transmite un byte que representa la longitud del área de datos que sigue. A pesar de que la longitud se transmite como un campo de 8 bits sin signo, la longitud máxima de una trama es de 127 bytes. La trama PHY 802.15.4 no posee ningún mecanismo de corrección de errores, trabajo que se deja a la subcapa MAC.

La trama MAC de IEEE 802.15.4

La subcapa MAC es la encargada de definir los esquemas de direccionamiento, control de flujo y control de errores para las capas superiores.

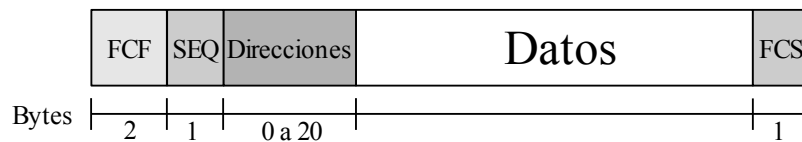


Ilustración 6: El formato de la trama MAC de IEEE 802.15.4

El Campo de Control de Trama (FCF) es un descriptor de 16 bits que contiene información sobre el resto de la trama:

Bits	Significado
0 a 2	Tipo de trama
3	Seguridad habilitada
4	Trama pendiente
5	Pedido de acknowledge
6	Intra-PAN
7 a 9	Reservados
10 y 11	Modo de direccionamiento de origen
12 y 13	Reservados
14 y 15	Modo de direccionamiento de destino

Tabla 4: Estructura del campo FCF de la trama MAC de IEEE 802.15.4

El tipo de trama define uno de cuatro posibles tipos de frames:

Tipo de trama	Descripción
000	Baliza
001	Datos
010	Acknowledge
011	Comando MAC
100 - 111	Reservados

Tabla 5: Tipos de trama MAC de IEEE 802.15.4

Los bits de modo de direccionamiento permiten especificar en las direcciones de origen y destino de cada trama dos posibles. IEEE 802.15.4 soporta direcciones cortas de 16 bits y direcciones estándar IEEE de 48 bits, similares a las empleadas por Ethernet (IEEE 802.3) y Wi-Fi (IEEE 802.11).

Modo de direccionamiento	Tipo de dirección
00	La dirección está ausente
01	Reservado
10	Dirección de 16 bits
11	Dirección de 48 bits

Tabla 6: Modos de direccionamiento de IEEE 802.15.4

El número de secuencia de la subcapa MAC se incrementa con cada trama transmitida y permite diferenciar entre retransmisiones de la misma trama.

Las direcciones de origen y destino de la trama se colocan a continuación del número de secuencia dentro de la trama MAC, y su formato depende del valor de los tipos de dirección definidos en los campos correspondientes del FCF.

Capítulo 5

La organización lógica de la red Kindergarten

En este capítulo se describirán las necesidades que llevaron a la creación una red inalámbrica desarrollada especialmente y su principio de operación.

Motivo del desarrollo

De lo abordado en el capítulo 2 estaba claro que las redes existentes en el mercado no estaban diseñadas para satisfacer los requerimientos que las aplicaciones context-aware. Se procedió entonces, con base en el hardware elegido y descrito en el capítulo 3, a diseñar un esquema de comunicaciones y localización capaz de soportar aplicaciones context-aware.

Los requerimientos para la red Kindergarten han sido definidos por las necesidades de la aplicación:

- Soportar un mecanismo de estimación de posición física del dispositivo con una exactitud tal que permita el seguimiento del movimiento de una persona en una habitación.
- Permitir construir un sistema de comunicación confiable a través del pasaje de mensajes entre estaciones.
- La posición estimada debe actualizarse mientras el usuario se desplaza, a intervalos regulares e independientemente de la existencia o ausencia de tráfico de datos y de la carga de CPU del sistema móvil.
- Todo intercambio de datos debe completarse en un tiempo determinístico.
- Ya que los dispositivos móviles que estarán comunicados a través de la nueva red tendrán un poder de procesamiento limitado, es imperativo que un dispositivo rápido no pudiera inundar con datos a otro más lento ni apropiarse de todo el ancho de banda de la red.
- La capacidad de transmisión del canal de datos debe aprovecharse de forma eficiente.
- El protocolo debe ser sencillo, de forma de permitir su implementación sobre dispositivos pequeños y baratos.

Modo de operación

Como primer paso para abordar el diseño de la red se creó una lista de capas de enlace de redes existentes en el mercado que presentan características deseadas para Kindergarten. También se analizaron las desventajas de cada una de ellas a fin de evitar, dentro de lo posible, caer en las mismas.

Capa de enlace	Ventajas	Desventajas
CSMA/CD (Ethernet) [22]	No requiere ningún tipo de coordinación entre estaciones	Tiempo no determinístico
Token Ring / Token bus [23]	Tiempo determinístico. No tiene colisiones	Requiere una topología física especial.
Aloha	Extremadamente simple	Muy bajo aprovechamiento del canal
Aloha ranurado	Simple	Bajo aprovechamiento del canal
CSMA/CA (Wi-Fi)	Hardware muy difundido en PDAs y Smartphones	El mecanismo de CA es complejo
Bluetooth	Hardware muy difundido en PDAs y Smartphones	Cantidad limitada de estaciones por red

Tabla 7: Pros y contras de los tipos de redes utilizados como punto de partida para Kindergarten

En vista de este análisis la idea fue combinar las características más deseables de cada una de los mecanismos actuales. De esta forma se intentó conservar el determinismo de Token Ring, pero sacrificando eficiencia para mantener la complejidad cerca de la del esquema ALOHA.

Se tomó de la red Token Ring la idea de división del tiempo en bloques o “slots” donde cada estación posee el control exclusivo del canal de transmisión. La división del tiempo hace que sea posible evitar las colisiones, donde más de una estación intenta transmitir a la vez y las señales se interfieren entre sí, eliminando la necesidad de proveer un manejo de colisiones en el protocolo mismo. Se eligió hacer que cada slot tuviera una longitud de tiempo fija, estándar y prefijada en todos los casos. Estas dos decisiones de diseño permitieron simplificar enormemente el protocolo, siendo una gran ventaja al momento de la implementación.

Al diseñar una red inalámbrica se tiene la ventaja de que en ella participarán al menos dos tipos de estaciones: las estaciones móviles que los usuarios llevarán consigo, y las estaciones fijas, llamadas en adelante estaciones base, en que estarán presentes en las distintas áreas cubiertas por la red.

Las estaciones móviles están alimentadas por baterías y por ende muy limitadas en prestaciones y autonomía. El software que corre en ellas debe ser sencillo, eficiente y muy optimizado para reducir el consumo hasta el límite.

Las estaciones base, por otro lado, no tienen limitaciones en cuanto a consumo porque pueden cablearse entre sí y a una fuente de alimentación conectada a la red eléctrica, y tampoco necesitan ser tan pequeñas como las estaciones móviles. Las estaciones base pueden ser complejas y tener un comportamiento más activo en la administración del acceso al medio.

Esto es similar a lo que ocurre en las redes inalámbricas existentes en la actualidad, donde routers Wi-Fi tienen un papel más complejo en la administración del acceso a la red en comparación con otras estaciones como computadoras, PDAs y teléfonos celulares.

De manera que en lugar de hacer que todas las estaciones de la red tuvieran el mismo nivel de libertad, se prefirió convertir a las estaciones base en coordinadoras del acceso al medio de las estaciones móviles. Cada estación móvil sólo puede transmitir información en intervalos predefinidos y exclusivamente cuando una estación base así se lo ordena. Y al igual que en Wi-Fi y Bluetooth, cada estación móvil está asociada a una única estación base a través de la que envía y recibe los mensajes intercambiados con otras estaciones móviles.

El funcionamiento general de la red, entonces, queda definido con los siguientes puntos principales:

- Funcionamiento master-slave: una estación móvil sólo puede utilizar el canal de transmisión cuando la estación base a la que está asociada así se lo indique.

- División del tiempo en intervalos predefinidos: cada estación base tiene un intervalo de tiempo en el que hace uso del canal de transmisión. En ese intervalo se definen slots de tiempo, también de tamaño fijo, en los que se transmiten cada una de los paquetes de datos entre estaciones.
- No puede darse el caso de que dos estaciones transmitan a la vez en una misma zona de cobertura.
- Todas las estaciones móviles deben poder transmitir información a través del canal de comunicaciones. No debe ser necesario que las estaciones móviles compitan entre sí.
- El algoritmo de scheduling de las estaciones es simple.
- El estado interno que deben mantener las estaciones móviles es el mínimo posible.

El ciclo de trabajo de la red

Para la explicación del ciclo de trabajo y el funcionamiento detallado se pondrá como ejemplo una red con cuatro estaciones base y dos estaciones móviles.

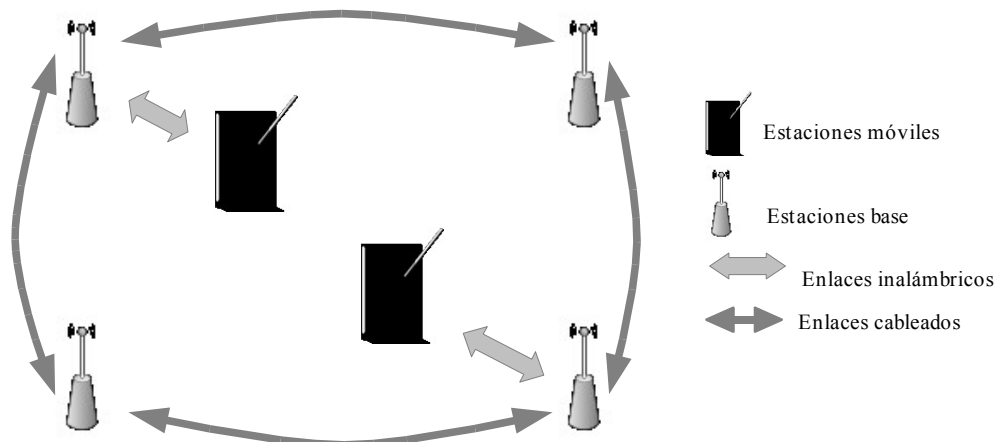


Ilustración 7: Ejemplo de red Kindergarten

La primer tarea para diseñar el modo de operación del protocolo Kindergarten es decidir la longitud de tiempo que tendría el slot dentro del que cada estación transmitiría sus datos. Para ello es necesario tomar en cuenta la longitud de todas las componentes de una trama de datos 802.15.4:

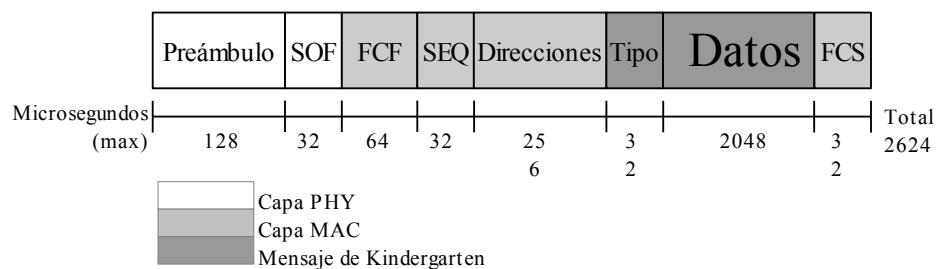


Ilustración 8: Tama IEEE 802.15.4 conteniendo un mensaje Kindergarten

Es decir que la transmisión de una trama IEEE 802.15.4 conteniendo un mensaje Kindergarten de 64 bytes (el máximo definido) toma 2624 microsegundos. A este tiempo debe sumarse 192 microsegundos para permitir que el módem de radio de la estación transmisora cambie a modo recepción luego de transmitir. Este tiempo está fijado por el fabricante del microcontrolador CC2431 y es inherente a su diseño. De manera que sumando los 2624 microsegundos del tiempo de transmisión de paquete a los 192 microsegundos del tiempo de calibración se obtiene un total de 2816 microsegundos.

El microcontrolador CC2431 posee un timer-counter interno dedicado especialmente al

módem de radio, con un período de 320 microsegundos. Este timer se puede programar para generar eventos de transmisión o recepción de información de forma autónoma, sin intervención de la CPU, o bien puede generar interrupciones de software para dar el control a rutinas específicas dentro del programa principal.

En el caso de Kindergarten se optó por usar este timer para delimitar la duración de un slot, definiéndolo en el tiempo de 9 ciclos de 320 microsegundos, lo que da un total de **2880 microsegundos**.

El ciclo de funcionamiento de la red asigna a cada una de las estaciones base un número de secuencia que determina el orden en que estas estaciones base, y sus estaciones móviles asociadas, utilizan el medio de transmisión. En cada uno de sus turnos cada estación base dispone de un conjunto de slots de tiempo para enviar y recibir información, y un período de inactividad para que nuevas estaciones móviles se unan a la red. El período de tiempo total asignado a una estación base recibe el nombre de *superframe*.

El formato del superframe consta de 5 zonas o etapas, donde la estación base envía y recibe 5 tipos distintos de información:

- En primer lugar, la estación base envía las tramas de datos, si las hubiera, a las estaciones móviles.
- Luego, la estación base realiza un broadcast de una trama de datos especial, donde se especifica en qué orden cada una de sus estaciones móviles asociadas deben transmitirle, si necesitaran, información a la estación base. Esta trama recibe el nombre de Order Of Transmission, abreviado OOT.
- Al recibir la trama de OOT las estaciones móviles transmiten su información, si la tuvieran, en el slot asignado.
- Por último, la estación base realiza un broadcast de una trama llamada Point Of Entry, o POE, marcando el inicio del período de guarda. En este frame también se incluye información necesaria para la estimación de localización, como se verá más adelante.
- Un período de inactividad, o guarda, de 1000 microsegundos para que nuevas estaciones transmitan su pedido de entrada a la red.

El número de slots de datos por superframe no está fijo en la implementación. Se han hecho experimentos con 4, 6 y 8 slots de datos de transmisión en ambos sentidos (8, 12 y 16 slots de datos en total). El valor final depende de la cantidad de estaciones móviles que se prevén en funcionamiento, y de la cantidad de superframes por segundo que se espera obtener de la red.

Por ejemplo, utilizando 8 slots de datos en cada dirección por superframe, se tienen los siguientes tiempos:

Tipo de trama	Tiempo asignado (μ s)
Tramas de datos (base \rightarrow móvil)	$8 * 2880 = 23040$
Order of transmission	2880
Tramas de datos (móvil \rightarrow base)	$8 * 2880 = 23040$
Point of Entry	2880
Período de guarda	1000
Total	52840

Tabla 8: Tiempo total de un superframe

El tiempo de un superframe en este caso se redondea a 53120 microsegundos para ajustarlo a la duración de 320 microsegundos del timer MAC del CC2431 con el que se hace toda la temporización de la red.

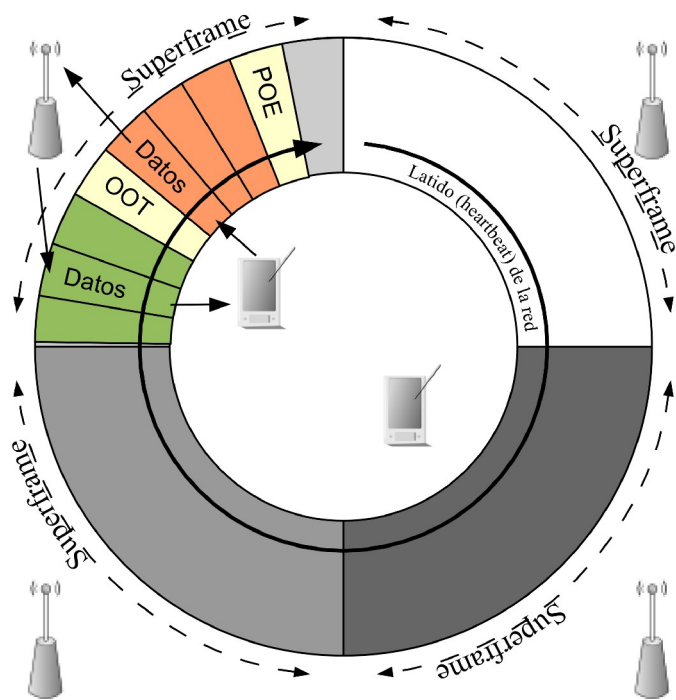


Ilustración 9: Esquema temporal de funcionamiento de Kindergarten

Así como al conjunto de tramas de datos que se transmiten o reciben a través de una estación base se le dio el nombre de Superframe, en el resto del texto se utilizará el nombre *latido* o *heartbeat* para referirse al conjunto de todos los superframes de una determinada red, en un ciclo de operación.

Este ciclo de latido de la red es lo que le da a Kindergarten su nombre. La analogía es la siguiente: imaginemos un grupo de niños pequeños, correteando en el jardín, gritándose unos a otros mientras juegan. En esencia, las estaciones inalámbricas son como estos niños: ellas deambulan libremente, gritándose unas a otras (a través del canal inalámbrico) cuando lo desean sin control (interfiriéndose mutuamente). El objetivo del diseño de Kindergarten fue crear un ambiente en el que los niños (estaciones móviles) fueran supervisados por maestros (estaciones base) que los hacen jugar por turnos.

Sub-división de la red

Cuando una red Kindergarten abarca un área muy amplia es posible que haya zonas donde el nivel de señal de una o más estaciones base sea demasiado bajo para ser utilizado como referencia de localización, o bien existan en el interior de la zona obstáculos para el paso de la señal como pueden ser construcciones u objetos metálicos grandes. En estos casos se hace imprescindible dividir esta única área de grandes dimensiones en dos o más áreas más pequeñas que sean cubiertas por sendos grupos de estaciones base.

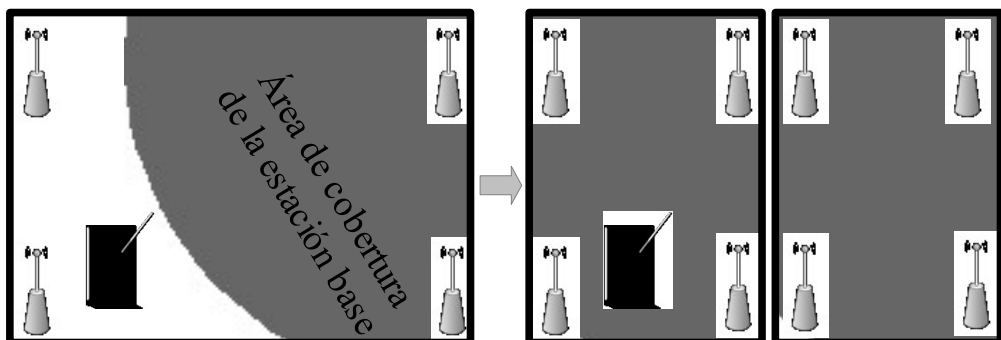


Ilustración 10: División de una red en dos subredes con cobertura total

El problema de dividir las áreas radica en la naturaleza síncrona de Kindergarten, que obliga a tomar precauciones para que los superframes de dos estaciones base no colisionen entre sí en las áreas cubiertas por ellas. Una colisión entre superframes de estaciones base implicaría la pérdida no sólo de tramas de datos, sino de las tramas de señalización OOT y POE que llevan la sincronización y la información de localización a las estaciones móviles.

La solución al problema consiste en sincronizar los latidos de las dos sub-áreas para que los superframes sólo coincidan en el tiempo si sus estaciones base no pueden interferirse entre sí. Esto es, sólo hacer que dos estaciones usen el canal a la vez cuando no hay forma de que la transmisión de una pueda interferir con la otra.

Esta sincronización sólo es posible gracias a que cada slot de tiempo para cada trama, y por lo tanto cada superframe, tienen exactamente la misma duración de tiempo, de manera que si dos sub-redes comienzan un latido al mismo tiempo, todos los superframes y slots estarán perfectamente sincronizados entre sí en ambas. Para asegurar que esto ocurra, y aprovechando el hecho de que las estaciones base están cableadas entre sí, se definió una conexión de datos muy sencilla que une a todas las estaciones base y señala a todas a la vez el comienzo de un superframe. Esta conexión, a la que llamamos *línea de latido*, es simplemente un cable de datos de un solo conductor que cambia de estado en el comienzo del latido de la red, y que puede estar controlado por una estación base especial o por un circuito de reloj especialmente dedicado a ese efecto.

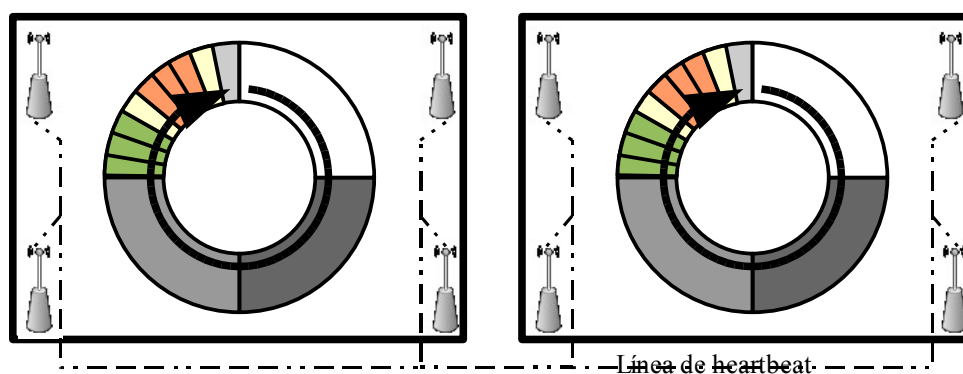


Ilustración 11: Línea de Heartbeat sincronizando las estaciones base y sus superframes

Cada estación base espera el disparo del latido monitoreando el cambio de estado en la línea de latido. Al ocurrir el disparo las distintas estaciones base esperan el turno de su superframe y cumplen con su función, y luego quedan a la espera de un nuevo latido. El ciclo se repite una y otra vez. El único requisito es que cada cambio de estado de la línea de latido, y por ende cada inicio del latido de la red, se produzca en el tiempo correcto: una vez que todas las estaciones base han cumplido su función y el latido anterior ha terminado.

Formato de las tramas de Kindergarten

En esta sección se hará una reseña del formato de la trama física y de la información agregada por la subcapa MAC de 802.15.4. Luego se presentará el formato de los mensajes específicos de Kindergarten

Los mensajes de Kindergarten

Kindergarten se basa en la trama MAC de IEEE 802.15.4 definiendo un espacio de direcciones plano de 16 bits (empleando las direcciones de 16 bits descritas anteriormente) para mantener las tramas cortas y concisas, pero este espacio puede fácilmente ampliarse empleando direcciones IEEE de 64 bits si fuera necesario en un futuro.

Todas las tramas de Kindergarten siguen el siguiente formato:

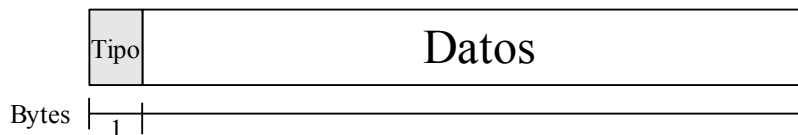


Ilustración 12: El formato del mensaje básico de Kindergarten

Esencialmente, constan de un único byte que determina el servicio de Kindergarten (datos, POE, OOT) al que pertenecen, seguido por los datos en sí. Esta trama se encapsula en una trama MAC 802.15.4.

Los mensajes de datos

Los mensajes de datos de Kindergarten se forman simplemente dividiendo la información que la capa de aplicación desee enviar en paquetes de hasta 64 bytes y encapsulándolos en una trama de Kindergarten de tipo 0 (cero).

Si los datos a enviar fueran más de 64 bytes, es responsabilidad de la aplicación realizar la fragmentación y reensamblado de los datos. Se determinó que estas operaciones fueran responsabilidad de las aplicaciones debido a que suelen estar desarrolladas utilizando lenguajes y herramientas de mayor nivel, y que al residir en una CPU externa al microcontrolador que corre Kindergarten, tienen a su disposición una mayor cantidad de memoria y poder de cómputo. Con el mismo criterio se ha decidido descargar sobre las capas superiores de software la tarea de retransmisión de mensajes perdidos o corruptos.

El mensaje OOT

El formato del mensaje OOT es:

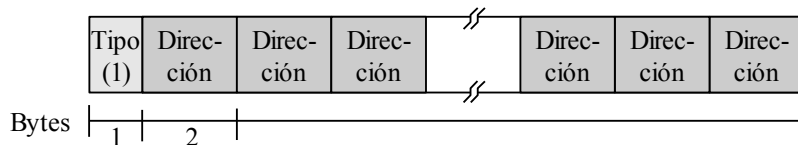


Ilustración 13: Formato del mensaje OOT de Kindergarten

El paquete de datos OOT es una colección de hasta 32 direcciones Kindergarten encapsuladas en una trama Kindergarten de tipo 1. La cantidad real de direcciones presentes es determinada por la cantidad de slots de datos en el superframe con el que está configurada la red Kindergarten, y se fija al momento de montar la red.

El mensaje POE

El formato del mensaje POE es:

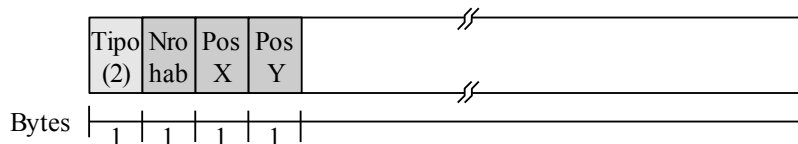


Ilustración 14: Formato del mensaje POE de Kindergarten

El mensaje POE es simplemente una baliza para señalar el fin del conjunto de slots de datos y el comienzo del período donde estaciones que deseen unirse a la red pueden solicitarlo a través de los mensajes de Joining Request, descritos más abajo. En su forma básica, es un mensaje Kindergarten con código 2 (dos).

Dado que el mensaje de POE se emite en todos los latidos de la red por cada una de las estaciones base, es un excelente medio para hacer llegar a las estaciones móviles información para que estas puedan estimar su localización. En el área de datos del POE cada estación envía, entonces, una estructura de información que consta de un número de habitación o área,

y la posición X e Y donde se encuentra. La forma en que estos datos son utilizados se describirá en el capítulo 5.

El mensaje de joining Request:

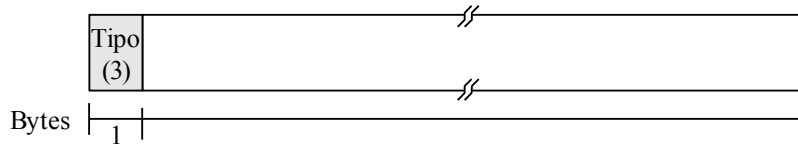


Ilustración 15: Formato del mensaje JR de Kindergarten

Cuando una estación móvil desea comenzar a participar en una red, sólo tiene que esperar a recibir un mensaje POE de una estación base cercana, y transmitir su mensaje Join Request (JR) en respuesta. Si el pedido de unión es aceptado, la estación base envía en el siguiente latido un mensaje del tipo Join Acknowledge, o JA, dirigido a la estación móvil en uno de los slots de datos. A partir del siguiente latido la estación móvil podrá recibir y enviar información a través de Kindergarten.

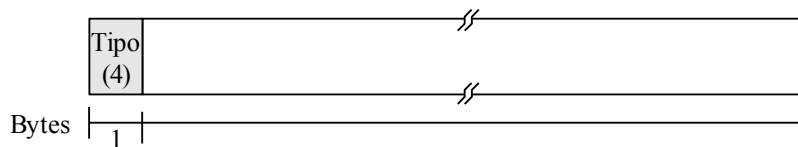
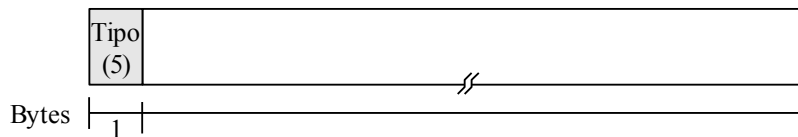


Ilustración 16: Formato del mensaje JA de Kindergarten

Si por alguna razón la red Kindergarten no puede soportar que la estación móvil se asocie a la estación base solicitada, en lugar de un Join Acknowledge se le envía un Join NAK (Join Negative Acknowledge o JN):



Dibujos 1: Formato del mensaje JN de Kindergarten

La estación móvil espera entonces un nuevo POE para volver a intentar el proceso de asociación.

Ruteo de mensajes entre estaciones

Dado que tanto las estaciones base como las estaciones móviles de Kindergarten pertenecen a un único espacio plano de direcciones de 16 bits se hace necesario crear un mecanismo de ruteo de mensajes. Este mecanismo debe tener en cuenta el camino que sigue un mensaje de datos de Kindergarten desde que es creado en una estación móvil, transmitido a través del aire hasta la estación base a la que la estación origen está asociada, el viaje a través de la red cableada hasta la estación base de destino, y su transmisión final a la estación móvil que lo recibirá.

Esto obliga a crear una infraestructura que contenga la información actualizada de qué estación móvil está asociada a cuál estación base, y que tome las decisiones sobre cómo encaminar los mensajes basado en ella. La infraestructura debe funcionar aún teniendo en cuenta las prestaciones limitadas del hardware utilizado en las estaciones base y móviles de Kindergarten.

La solución adoptada es similar a la empleada por los diseñadores del propio protocolo IEEE 802.15.4: una estación dentro de la red opera como “master router” y direcciona los mensajes a la estación base correspondiente. El master router es el encargado de recibir los mensajes de Join Request emitidos por las estaciones móviles y con ellos actualiza la tabla maestra de asociaciones que él mismo posee.

Los mensajes Kindergarten son recibidos por las estaciones móviles y encapsulados en datagramas UDP que son reenviados al master router. El master router reenvía luego los datagramas a la estación base destino, que lo desensambla y obtiene el mensaje Kindergarten. Este mensaje es luego transmitido a la estación móvil.

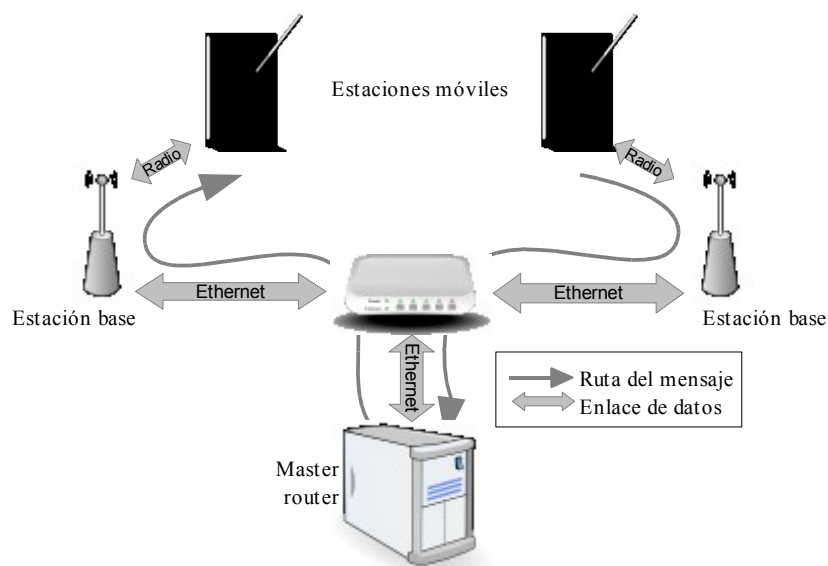


Ilustración 17: Camino recorrido por un mensaje Kindergarten entre dos estaciones móviles

Este master router puede ser cualquier sistema informático capaz de conectarse a la infraestructura de comunicaciones que une a las estaciones base Kindergarten. En el caso de la implementación que se describe en el presente trabajo, donde la infraestructura es una red Ethernet, el master router es simplemente un equipo de escritorio conectado a la misma red que las estaciones base.

El principio de trabajo del master router es sumamente sencillo: posee una tabla maestra donde almacena cada una de las direcciones Kindergarten de las estaciones móviles, y la dirección ethernet de la estación base a la que ella está asociada. Al llegar un mensaje de Join Request, el master router registra la dirección Kindergarten de la estación que emitió el request y la dirección Ethernet de la estación base que lo recibió, y actualiza la tabla. Cuando el master router recibe un datagrama conteniendo un mensaje Kindergarten de datos, determina a partir de la dirección Kindergarten de destino la dirección IP de la estación base a la que debe reenviar el mensaje.

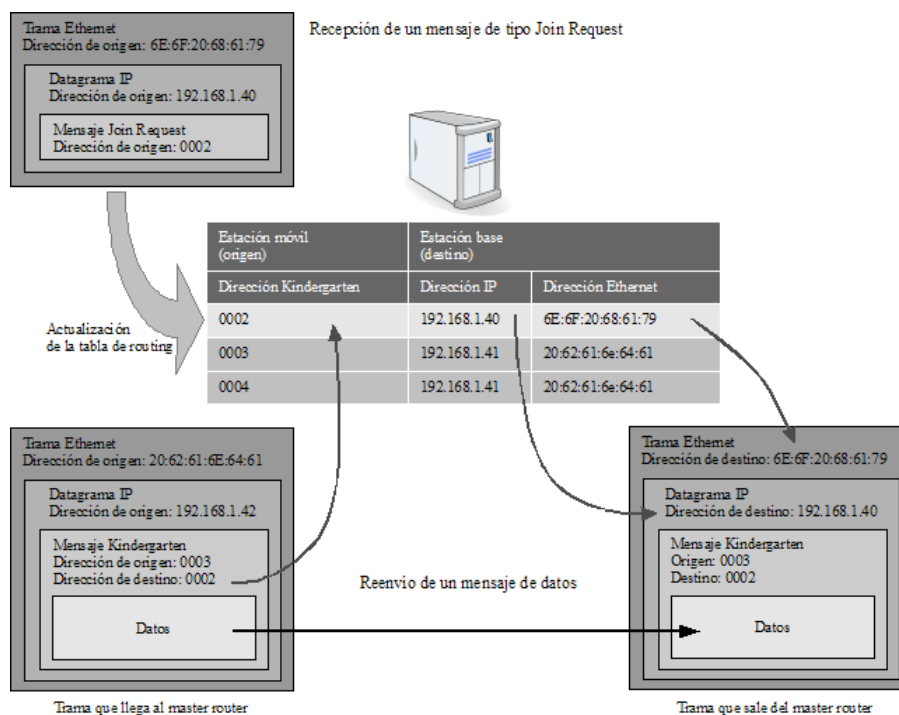


Ilustración 18: Operaciones utilizando la tabla de routing del Master Router

Cuando una estación móvil se desplaza, es posible que abandone una zona cubierta por una sub-red y entre en una sub-red distinta, por ejemplo cuando el usuario que utiliza la estación móvil abandona una habitación para pasar a otra. Cada una de las estaciones móviles monitorea por esta razón el nivel de señal que recibe de la estación base a la que está asociada; si el nivel de señal cae por debajo del de otras estaciones base, la estación móvil envía un Join Request a la estación base con mejor nivel de señal y comienza a cursar sus comunicaciones a través de ella.

Organización del stack de Kindergarten

Kindergarten, siendo un protocolo master-slave, posee diferentes implementaciones en las estaciones móviles y en las estaciones base. En los siguientes párrafos se hará una descripción de los módulos de software comunes a ambas implementaciones y los específicos de estaciones móviles y estaciones base.

Código común

Ambos tipos de estaciones, móviles y base, comparten las capas inferiores del stack de Kindergarten encargadas de la comunicación con el módem de radio integrado en el microcontrolador CC2431, y el código de encapsulamiento de mensajes Kindergarten en tramas IEEE 802.15.4.

Para la interacción con el hardware se utiliza la biblioteca de funciones Simple Packet Protocol (SPP) provista por la misma empresa Texas Instruments, fabricante del CC2431. Esta biblioteca inicializa el módem de radio, comanda las transmisiones y recepciones asíncronamente y utiliza DMA para la transmisión y recepción de información de forma desatendida. También utiliza un mecanismo de callback al código de la aplicación cada vez que una trama es recibida.

El nombre Simple Packet Protocol deriva de su naturaleza minimalista: su única función es la transmisión y recepción de tramas 802.15.4 nodo-a-nodo o broadcast de la forma más eficiente y simple para el desarrollador. La biblioteca de SPP representa una trama a transmitir y las tramas recibidas con estructuras que contienen los datos de los encabezados y los datos:

```
typedef struct{
```

```

    BYTE payloadLength;
    BYTE destAddress;
    BYTE flags;
    BYTE *payload;
}SPP_TX_STRUCT;

typedef struct{
    BYTE payloadLength;
    BYTE destAddress;
    BYTE srcAddress;
    BYTE flags;
    BYTE payload[SPP_MAX_PAYLOAD_LENGTH + SPP_FOOTER_LENGTH];
}SPP_RX_STRUCT;

```

A través de las funciones provistas por SPP es posible controlar todos los aspectos del funcionamiento:

```

BOOL sppInit(UINT32 frequency, BYTE address)

```

SppInit hace el trabajo de inicialización de los módulos de radio del CC2431 y la máquina de estados que gobierna el protocolo SPP, configura el hardware con la frecuencia seleccionada y almacena internamente la dirección local de la estación que se utilizará en las tramas enviadas.

```

void sppSetRxCallbackFunction(FUNCTION* callBackFunction)

```

A través de sppSetRxCallbackFunction la aplicación envía a la biblioteca de SPP la dirección de la función que será invocada cuando el hardware recibe una trama. La función declarada como callback debe ser lo suficientemente rápida para terminar y retornar el control al stack en menos tiempo que el que demora una nueva trama en arribar, dado que ni el hardware ni la biblioteca de SPP tienen la capacidad de almacenar múltiples tramas en memoria.

```

BYTE sppSend(SPP_TX_STRUCT* pPacketPointer)

```

SppSend envía a través de la interfaz de radio la trama representada por la estructura pasada como parámetro. Aunque en Kindergarten puro no se dé la situación, sppSend está preparada para utilizar carrier-sense multiple access (CSMA) para sensar la actividad del canal y transmitir cuando este está libre.

La llamada a sppSend es bloqueante, y retorna un código de error representando el resultado de la operación:

Valor de retorno	Nombre simbólico	Significado
0	OK	Operación exitosa
4	CHANNEL_BUSY	El canal está ocupado (sin usar CSMA)
8	TOO_LONG	La operación tomó demasiado tiempo para completarse

```

void sppReceive(SPP_RX_STRUCT* pReceiveData)

```

La función inicia el proceso de recepción de una trama por las rutinas de SPP. SppReceive no

espera a una trama, sino que inicializa las interrupciones y canales de DMA necesarios y retorna el control a la función llamante. Cuando el módem de radio del CC2431 recibe una trama, esta es procesada asincrónicamente con el programa principal y se invoca a la función de callback dentro del propio stack de Kindergarten.

Código específico de las estaciones móviles

El código de Kindergarten en las estaciones móviles está contenido en el archivo `protocol.c` y define una máquina de estados que reacciona a los eventos que se producen en el conjunto CC2431-PDA, internos o externos, a medida que se vayan produciendo.

La máquina de estados de Kindergarten opera completamente disparada por eventos, sin siquiera ser llamada por el lazo de control principal del programa:

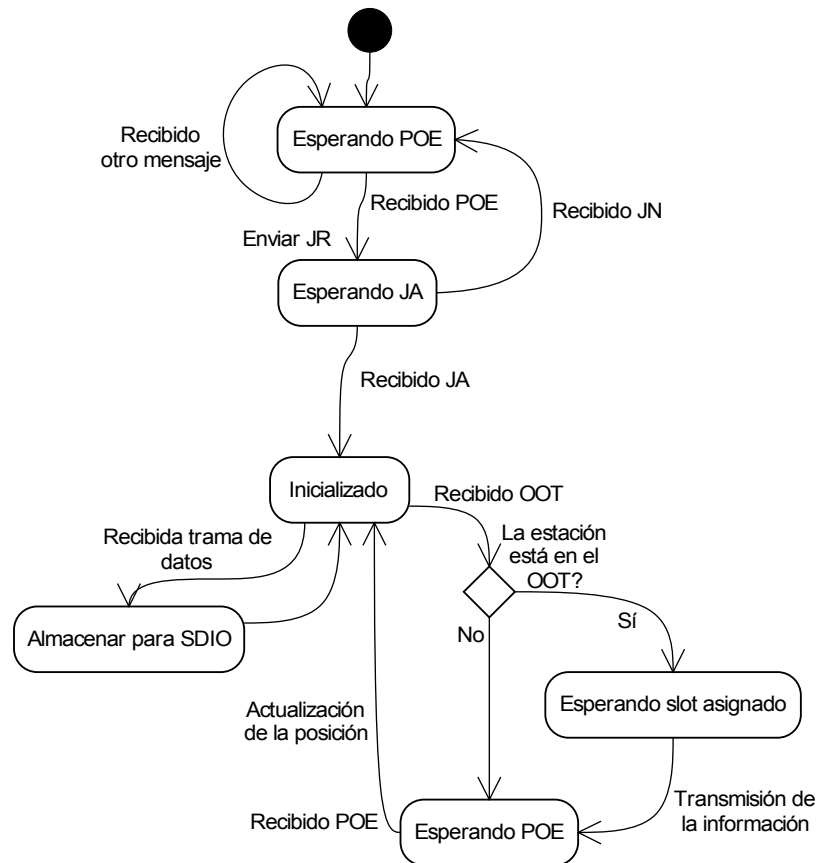


Ilustración 19: Máquina de estados de una estación móvil Kindergarten

Cada evento es disparado por la llegada de una trama a través de la biblioteca de SPP, que invoca a la función `protocol_packet_received()`.

Código específico de las estaciones base

Las estaciones base son más sencillas desde el punto de vista de Kindergarten. Su única función es retransmitir mensajes entre la red cableada (y el Master Router) y las estaciones móviles asociadas a ellas. La estación base también trabaja de forma casi totalmente asíncrona al flujo de control del programa principal, siendo sus principales funciones disparadas por la línea de heartbeat y timers internos que inician la transmisión de los mensajes de datos para las estaciones móviles, y los mensajes de OOT y POE.

El ciclo de funcionamiento de la estación base se centra en los archivos `protocol.c` y `wireless.c`. Una vez que la máquina de estados ha sido inicializada por la llamada a la función `protocol_init()` y `wireless_init()` el punto de entrada de la máquina

de estados son las funciones `wireless_packet_received()` y `heartbeat_interrupt_handler()` en el módulo de wireless.

El diagrama de estados de la estación base es el siguiente. Nótese que el envío y la recepción de los mensajes de datos se hace mientras la máquina de estados está esperando que se cumplan plazos relacionados con la temporización del superframe.

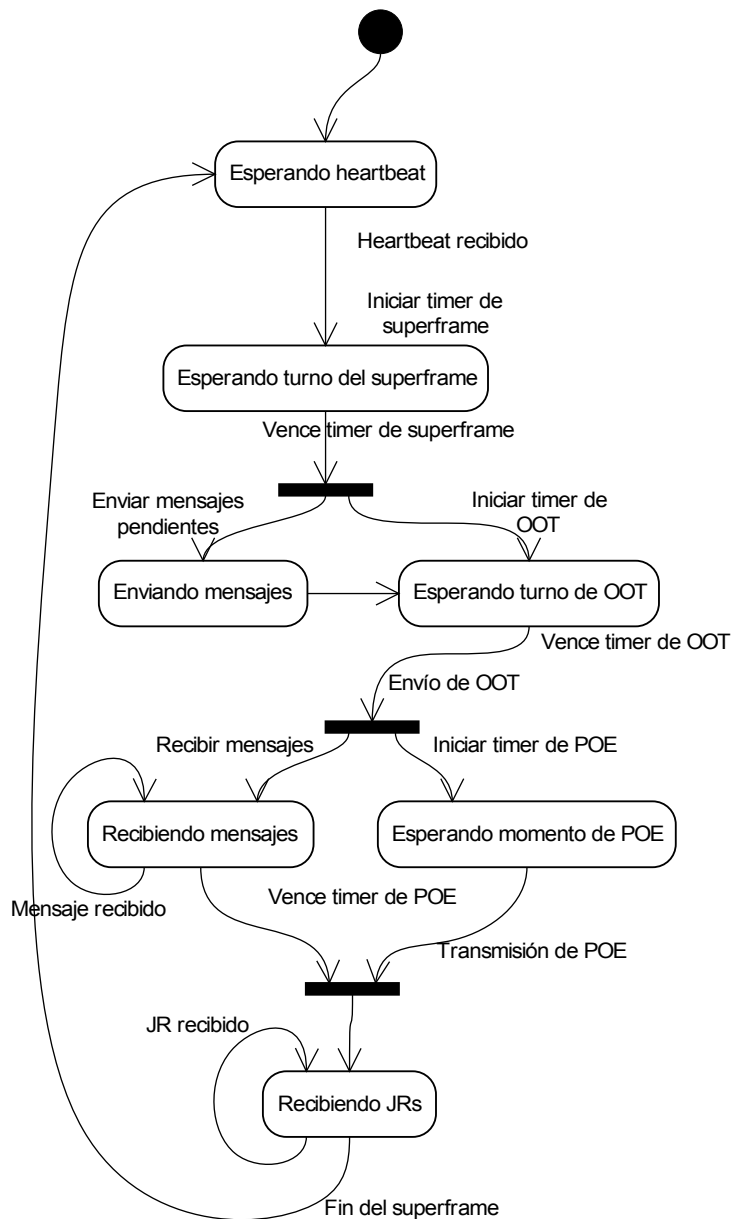


Ilustración 20: Máquina de estados de una estación base Kindergarten

`Heartbeat_interrupt_handler()` es la función invocada por el controlador de interrupciones del CC2431 en el evento de recepción de un nuevo heartbeat. Esta función pone a la máquina de estados del protocolo Kindergarten en su estado inicial y configura el timer MAC del CC2431 para que espere la cantidad necesaria de tiempo que corresponden al inicio del superframe de la estación.

Por ejemplo, una estación que tenga asignado el primer superframe comenzará inmediatamente con las operaciones, mientras que una que tenga asignado el segundo superframe programará el timer MAC para generar un evento 53120 microsegundos (166 ciclos de 320 microsegundos) después.

Como puede verse en la ilustración de la máquina de estados de la estación base, todas las demás transiciones de estados son disparadas por eventos del timer MAC. Esta metodología

permite programar un evento para el futuro y utilizar el flujo de control principal del programa para hacer envío y recepción de tramas y mensajes de datos entre estaciones, e interacción con la red cableada mientras se espera.

Capítulo 6

La estimación de posición en Kindergarten

En este capítulo se describe brevemente el principio físico por el que es posible estimar la distancia entre un transmisor y un receptor de radio, el funcionamiento del módulo de estimación de posición del CC2431, su aplicación a Kindergarten y posibles alternativas al mismo.

Kindergarten basa su capacidad de estimación de posición en la propiedad física de las ondas de radio de variar su intensidad de acuerdo a la distancia del emisor de la señal con respecto al receptor. Esto se debe a que en el espacio, todas las ondas electromagnéticas (radio, luz, rayos X) obedecen la denominada ley de la inversa del cuadrado, que dice que la densidad de potencia de una onda electromagnética es inversamente proporcional al cuadrado de la distancia de la fuente[24]:

$$I = \frac{S}{4\pi r^2}$$

Donde el valor S representa la intensidad de la señal en su origen e I la intensidad de la señal en cualquier punto de la esfera de radio r centrada en el transmisor. Duplicar la distancia al transmisor implica que la densidad de potencia de las ondas irradiadas en la nueva posición se reduce a la cuarta parte de su valor anterior.

Si se grafica la curva distancia-síñal para un transmisor hipotético de una potencia arbitraria “1” en alguna unidad determinada, se obtendrá la siguiente curva:

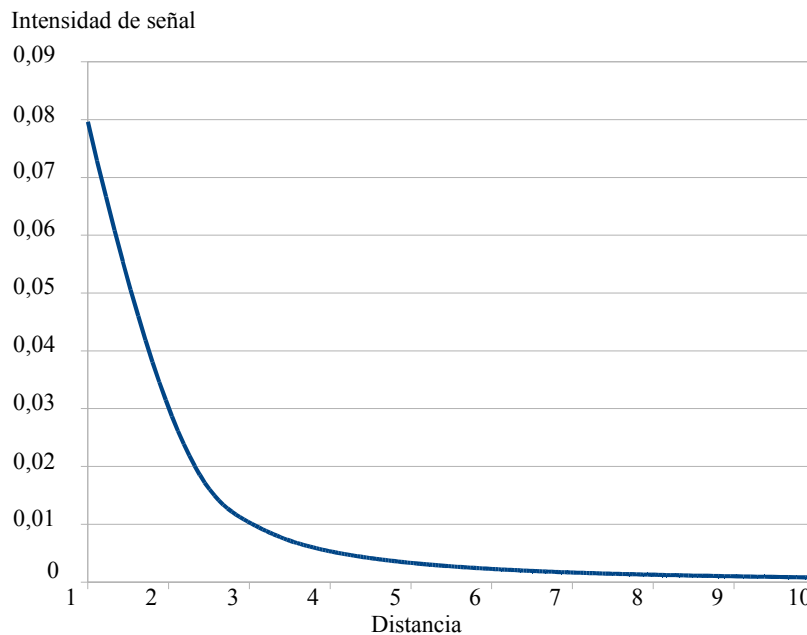


Ilustración 21: Relación distancia-nivel de señal

Descripción del módulo de location awareness

El módulo de localización del CC2431 se usa para estimar la posición de nodos en una red inalámbrica creada a tal efecto. En ella existen nodos de referencia con coordenadas conocidas, típicamente por ser parte de la infraestructura de la instalación.

Otros nodos actúan como “nodos ciegos”, cuyas coordenadas necesitan ser estimadas. Estos nodos son los que en Kindergarten actúan como estaciones móviles, actuando aislados o asociados a terminales PDA o Smartphone.

El módulo de localización implementa un algoritmo de cómputo distribuido que utiliza el valor de la intensidad de señal recibida (RSSI) de nodos de referencia conocidos (estaciones base). Realizar el cálculo de la posición a nivel de nodo reduce el tráfico de red y las demoras de comunicación que estarían presentes en un enfoque más centralizado.

El módulo de localización presente en el CC2431 tiene las siguientes características:

- El algoritmo utiliza entre 3 y 8 nodos de referencia
- La estimación de posición tiene una resolución de 50 centímetros cuadrados
- El tiempo necesario para estimar la posición de un nodo es de menos de 40 μ s
- Área de cobertura de 64 * 64 metros
- Error de localización menor a 3 metros
- La estimación ocupa mínimos recursos de CPU

Para obtener la mejor precisión posible es necesario emplear antenas isotrópicas. Esto es, que irradian la misma cantidad de energía en todas las direcciones. El error de localización depende del entorno de la señal y el número y la localización de las estaciones base que sirven como referencia en el área. En general, tener más estaciones de referencia hace que la estimación de posición sea más precisa.

Operación del módulo de localización

Esta sección describe los pasos básicos para obtener estimaciones de posición del módulo de localización del CC2431.

El módulo de localización requiere la provisión de un conjunto de entre 3 y 8 coordenadas de referencia junto con un conjunto similar de parámetros medidos. La salida del módulo de localización es un par de coordenadas de localización.

El módulo de localización requiere que se le indique las coordenadas de cada una de las estaciones de referencia como una serie de entre 3 y 8 pares de valores de 8 bits [x0, y0, x1, y1... x7, y7] donde cada valor es la coordenada X o Y de un nodo expresada en múltiplos de 25 centímetros. Así en 8 bits es posible expresar el rango (0 .. 63,75) con resolución de 0,25 metros. El formato utilizado es binario de punto fijo, donde los dos bits menos significativos representan la parte fraccional y los seis restantes la parte entera.

Los parámetros A y N

Dado que las características de propagación de señal varían enormemente en las distintas instalaciones, y a través del uso de diferentes tipos y calidades de antenas para la transmisión y recepción de las señales, el módulo de localización permite parametrizar estas variables a través de dos valores llamados A y N:

El parámetro A

El parámetro A se define como el valor absoluto del promedio la intensidad de señal recibida, expresada en dBm, a una distancia de un metro del transmisor, asumiendo un patrón de radiación omnidireccional. Por ejemplo, si la intensidad promedio recibida es de -40 dBm, el parámetro A tendrá un valor de 40. El módulo espera que el valor A esté en el rango [30 – 50] con una precisión de 0,5.

El valor de A se da como un valor de 8 bits sin signo con punto fijo donde el bit menos significativo es el dígito fraccional y los 7 bits más significativos son la parte entera. Un valor típico de A es 40,0

El parámetro N

El parámetro de radio N se define como el exponente de pérdida del camino que describe la velocidad con la que decae la intensidad de la señal cuando se aumenta la distancia entre el

transmisor y el receptor. La caída es proporcional a $d-N$ donde el valor d corresponde a la distancia entre transmisor y receptor. El valor final del parámetro N es un índice entero que corresponde a un valor dentro de la siguiente tabla:

Índice de N	N	Índice de N	N
0	1,000	16	3,375
1	1,25	17	3,50
2	1,50	18	3,625
3	1,75	19	3,75
4	1,875	20	3,875
5	2,00	21	4,00
6	2,125	22	4,125
7	2,25	23	4,25
8	2,375	24	4,375
9	2,50	25	4,50
10	2,625	26	4,625
11	2,75	27	5,00
12	2,875	28	5,500
13	3,00	29	6,00
14	3,125	30	7,00
15	3,250	31	8,00

Por ejemplo, si el valor N medido fuera de $N=2,98$ se utiliza la entrada en la tabla más cercana a ese valor, que corresponde a 3,00 y al índice 13. El 13 es entonces el valor del parámetro que se ingresa al módulo de posicionamiento.

Todo el funcionamiento del módulo se controla a través de cuatro registros de 8 bits dentro del CC2431: LOCENG, REFCOORD, LOCX y LOCY. Estos registros se encuentran dentro del espacio de memoria de la CPU del microcontrolador, y son accesibles desde el código de usuario como variables globales.

El registro LOCENG es un mapa de bits que contiene flags de control y status del estado del módulo de localización:

Bit	Nombre	Función
4	EN	Controla el módulo de localización 0: Deshabilitado 1: Habilitado
3	DONE	(Sólo lectura) Este bit es un flag que indica cuando el módulo terminó el cálculo de una nueva posición. 0: El módulo se encuentra ocupado procesando 1: Hay un resultado listo
2	PARLD	Al poner en 1 este bit se inicia la carga de parámetros de señal en el módulo.
1	REFLD	Al poner en 1 este bit se inicia la carga de las coordenadas de las estaciones de referencia.
0	RUN	(Sólo escritura) Poniendo este bit en 1 se inicia un nuevo cálculo de posición a partir de los parámetros ingresados.

Un ciclo de estimación de posición se inicia activando los bits EN y REFLD del registro LOCENG. Al activar REFLD el registro REFCOORD queda listo para recibir los 8 pares de coordenadas de las estaciones de referencia en el orden x0, y0.. x7, y7. Siempre deben cargarse 8 pares de coordenadas; si las estaciones de referencia son menos de 8 las coordenadas faltantes se representan con pares (0, 0).

Al terminar de cargar las coordenadas se pone en 0 el bit REFLD y se pone en 1 el bit PARLD del registro LOCENG. Tras este paso se cargan los parámetros A y N, y 8 valores del RSSI de las estaciones de referencia en el registro PARLD. También en este caso es necesario cargar en el registro 8 juegos de datos, rellenando con ceros en caso de tener un número menor de estaciones de referencia.

Por último se pone en 0 el bit PARLD y se activa el bit RUN del registro LOCENG, con lo que el módulo inicia un nuevo ciclo de estimación de posición. El software puede determinar cuándo un cálculo ha terminado inspeccionando el valor del bit DONE de LOCENG, o bien activando una interrupción especial asignada al módulo de localización.

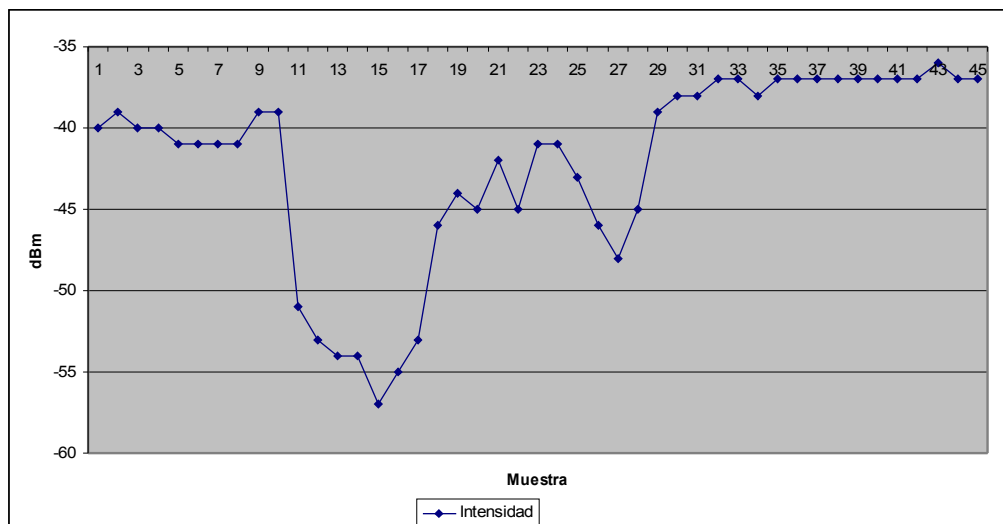
Integración con el resto de la estación móvil

En los siguientes párrafos se describirá el código que administra y actualiza la información de posicionamiento dentro de la estación móvil.

En el capítulo 4 se mencionó que en el mensaje POE se incluye información para la actualización de la localización. Esta información abarca un número que identifica una habitación o espacio cerrado donde está la estación móvil que emite el mensaje POE, y su posición X e Y en ese espacio.

La intensidad de la señal con la que se recibe el mensaje POE de cada estación se utiliza para alimentar el mecanismo de estimación de posición de Kindergarten; sin embargo el valor instantáneo de la señal dista mucho de ser totalmente representativo de la distancia: factores como interferencias transitorias o ruido originados en artefactos eléctricos y electrónicos cercanos pueden distorsionar el valor instantáneo de la señal y así generar lecturas erróneas.

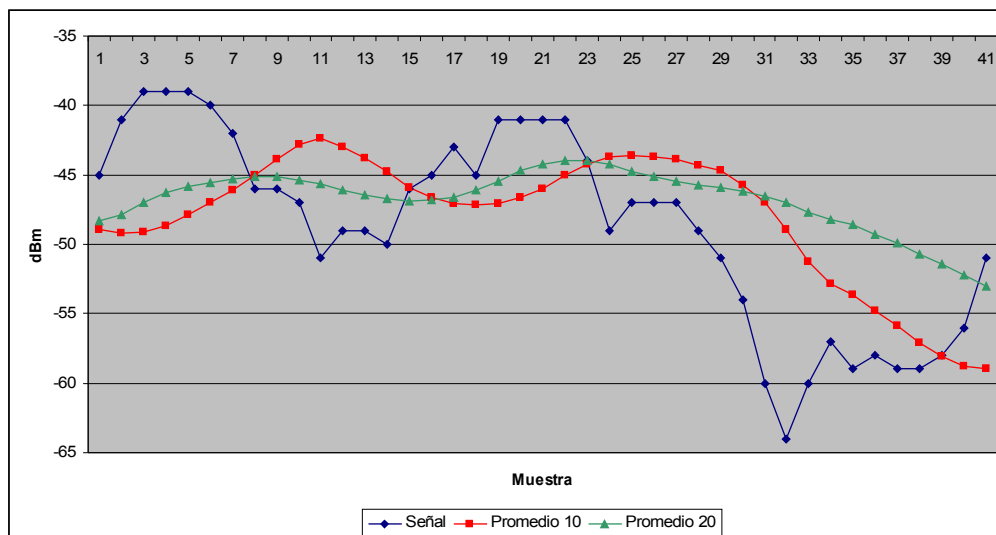
Como ejemplo puede tomarse el siguiente conjunto de lecturas tomadas cada 0,2 segundos con una distancia de un metro entre emisor y receptor, realizado dentro de un laboratorio de la Facultad de Informática, en un ambiente con 10 computadoras e iluminación fluorescente (los valores están expresados en dBm):



Para mitigar los efectos de interferencias transitorias, el valor efectivo de intensidad de señal que se emplea para alimentar el módulo de localización es el *promedio móvil* de 10 valores instantáneos de cada estación. El uso de un promedio móvil permite filtrar interferencias

transitorias al mismo tiempo que mantiene una correlación con la variación de señal ante el movimiento de la estación móvil.

El siguiente gráfico ilustra la caída de señal recibida cuando un usuario se desplaza de una distancia de 2 metros a una distancia de 4 metros de una estación base llevando consigo una estación móvil. Las condiciones de medición son las mismas que en el caso anterior. Se ilustran los valores de intensidad de señal medidos, y los promedios móviles obtenidos con 10 y 20 muestras:



Estructura del software de localización

El código de estimación de posición en las estaciones móviles está contenido en el archivo `location.c`. Cada estación que el software es capaz de utilizar como referencia está descrita por una estructura de datos que almacena toda la información pertinente:

```
struct _station_data {
    BYTE station_address;
    INT8 rssi_history[CONFIG_STATION_REMEMBERED_STRENGTHS];
    BYTE used_history; // Cuántos datos se tienen de la estación
    BYTE next_history; // Puntero al próximo lugar en rssi_history libre
    INT8 calculated_strength;
    INT8 xpos;
    INT8 ypos;
    BYTE room_number;
    bool valid;
};
```

Los valores de `station_address`, `room_number`, `xpos` e `ypos` describen a la estación en sí, su dirección y localización física, mientras que en el array `rssi_history` se almacena el historial de las últimas N intensidades de señal registradas. RSSI es la sigla que denota el Indicador de Intensidad de Señal Recibida⁶. Cada vez que el array `rssi_history` se actualiza, se recalcula el promedio y este se almacena en el campo `calculated_strength`.

Los puntos de entrada a este módulo son las funciones `location_update_from_packet()`, `location_cycle_stations()` y `location_update_position()`.

La primera de las funciones, `location_update_from_packet()` es invocada por el

6 Del inglés Received Signal Strength Indicator

código de procesamiento del mensaje POE cada vez que uno de estos mensajes llega a la estación móvil. Su función es actualizar el historial de las intensidades de señal medidas para la estación base que originó el mensaje POE. Si la estación base no se encuentra entre las estaciones registradas, `location_update_from_packet()` sólo la registra si pertenece al mismo número de habitación que la estación base a la que la estación móvil está asociada. En otras palabras, sólo se toman en cuenta las intensidades de señal de las estaciones base que están en la misma zona o habitación que la estación móvil.

`Location_cycle_stations()` se encarga de hacer un seguimiento de los niveles de señal de la estación base a la que está asociada la estación móvil y a los de otras estaciones base en la cercanía. Esta función compara los niveles de señal actuales recibidos de las distintas estaciones base en el área con el historial de nivel de señal de la estación base a la que está asociada la estación móvil.

Si `location_cycle_stations()` encuentra una estación base que consistentemente ha tenido mejor nivel de señal que la estación base a la que actualmente está asociado, inicia el mecanismo de re-asociación a la estación base con mejor nivel de señal. Esto permite que las estaciones móviles permanezcan asociadas a las estaciones base con mayor calidad de señal en todo momento, y las conexiones de datos no se pierdan cuando el usuario se mueve de una zona o habitación a otra.

Por último, `location_update_position()` es también invocada por el programa principal y es la encargada de alimentar el módulo de localización del CC2431 con los datos de las estaciones almacenados en la memoria. Esta función se encarga de inicializar el módulo de posicionamiento, cargar en su memoria la información a través del mecanismo explicado previamente, y almacenar los resultados obtenidos en memoria de programa para su posterior uso.

Soporte para otros modelos de localización

La implementación del modelo de estimación de posición descrita anteriormente está basada en el módulo de hardware incluido en el microcontrolador CC2431, pero el software que implementa el modelo de localización está totalmente autocontenido en el archivo `location.c` y sus diferentes puntos de entrada, ya sean en forma de callback o como invocación directa, están bien definidos. Esto permite reemplazar el código contenido en `location.c` por otra implementación que en el futuro se estime conveniente sin modificar el resto de los módulos de software de la estación móvil.

Capítulo 7

La comunicación PDA-CC2431 en las estaciones móviles

En este capítulo se aborda el problema del intercambio de datos entre el microcontrolador que implementa el protocolo Kindergarten y la PDA o Smartphone donde se ejecuta el sistema context-aware.

Al momento de abordar la integración del hardware de la PDA o Smartphone con el microcontrolador CC2431 donde corre el stack de Kindergarten surge el problema del mecanismo de comunicación a utilizar entre ambos. La elección no es sencilla dada la falta de uniformidad entre los distintos modelos de PDA en el mercado.

La solución presentada utiliza el bus Secure Digital como mecanismo de conexión por ser prácticamente ubicuo en los dispositivos existentes en el mercado, pero también por su alta velocidad de transferencia y fiabilidad.

Descripción de Secure Digital

Secure Digital (en adelante SD) es el nombre de un tipo de tarjeta de memoria desarrollada conjuntamente por Panasonic, SanDisk y Toshiba en 1999, creada como una mejora de las tarjetas MultiMedia Card, o MMC. Mecánicamente, las tarjetas SD tienen una forma rectangular de 32mm por 24mm y se comunican a través de una hilera de 9 islas de cobre que corresponden a las 9 señales del bus SD:

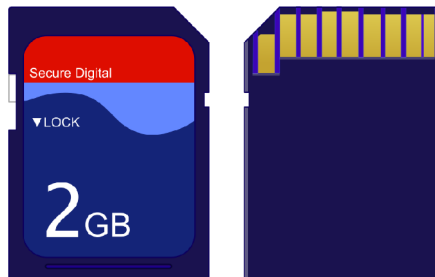


Ilustración 22: Frente y dorso de una tarjeta SD

Físicamente el bus SD es un bus master/slave síncrono, donde la tarea de control y arbitraje de las transferencias recae sobre un dispositivo **host** con el que se comunican las **tarjetas SD**. Host y tarjetas intercambian información en forma de paquetes de datos comando/respuesta. Sólo el host puede iniciar un diálogo con una tarjeta, y las transferencias de datos se inician exclusivamente a petición del host. Posee tres modos de funcionamiento, dependiendo de las capacidades de las tarjetas y del host en cada caso.

Los dos primeros modos de funcionamiento corresponden a los modos SD. En estos modos, el bus se comporta como dos interfaces seriales síncronas que comparten una única señal de reloj. Estas dos interfaces reciben los nombres CMD (command) y DAT (data). A través de CMD el host envía los comandos a las tarjetas, y recibe las respuestas de status si correspondiera. La interfaz DAT se utiliza para transferir bloques de datos entre host y tarjeta, cuando la cantidad de información a transferir excede lo que pueden transportar las estructuras de comando o respuesta; típicamente 4 bytes.

La diferencia entre los modos de SD radica en el ancho de la interfaz DAT, que puede ser de 1 ó 4 bits dependiendo de las capacidades de la tarjeta y el host. El soporte para SD de 1 bit es obligatorio tanto para hosts SD como para tarjetas, y se utiliza como mecanismo de fallback en caso de que el host o la tarjeta no soporten modo de transferencia de 4 bits, que la especificación SD establece como opcional en ambos dispositivos.

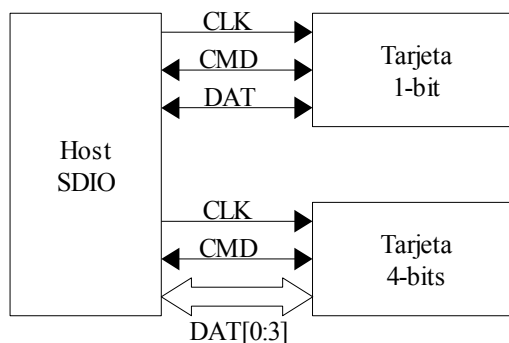


Ilustración 23: Distintos anchos de DAT en tarjetas SD

El tercer modo de funcionamiento es el denominado SPI (Serial Peripheral Interface). Todas las tarjetas deben soportar este modo de forma obligatoria pero el soporte de SPI por parte de los hosts es opcional. En este modo, la tarjeta utiliza las líneas CMD y DAT como un puerto serial SPI por donde se envían y reciben tanto comandos como datos. Este modo fue creado para permitir que las tarjetas fueran utilizadas por dispositivos microcontrolados simples, donde el costo y la complejidad no justifican incluir un controlador SD completo. Más adelante se abordará SPI cuando se hable de la implementación de SDIO con el CC2431.

Función			
Pin	Modo SD 1-bit	Modo SD 4-bit	Modo SPI
1	Card detect	Card Detect / DAT bit 3	Chip select
2	CMD	CMD	MOSI
3	Vss	Vss	Vss
4	Vdd	Vdd	Vdd
5	Clock	Clock	Clock
6	Vdd2	Vdd2	Vdd2
7	DAT	DAT bit 0	MISO
8	n/c	DAT bit 1	n/c
9	n/c	DAT bit 2	n/c

En adelante este documento se concentrará en el modo SD de 1 bit exclusivamente, por ser el implementado en el trabajo que aquí se describe. Para más información acerca de los otros dos modos de funcionamiento de las tarjetas SD, puede descargarse el texto de la especificación física de SD del sitio de la Asociación Secure Digital Card[25].

Todas las operaciones SD son iniciadas por el host enviando un comando a través de la línea CMD, y a través de esta misma línea es que la tarjeta envía el mensaje de status (si lo hubiera) que reporta el resultado de la operación. Si el comando genera por parte de la tarjeta una transferencia de datos, la respuesta es enviada o recibida a través de las líneas de datos en simultáneo con la respuesta de status.

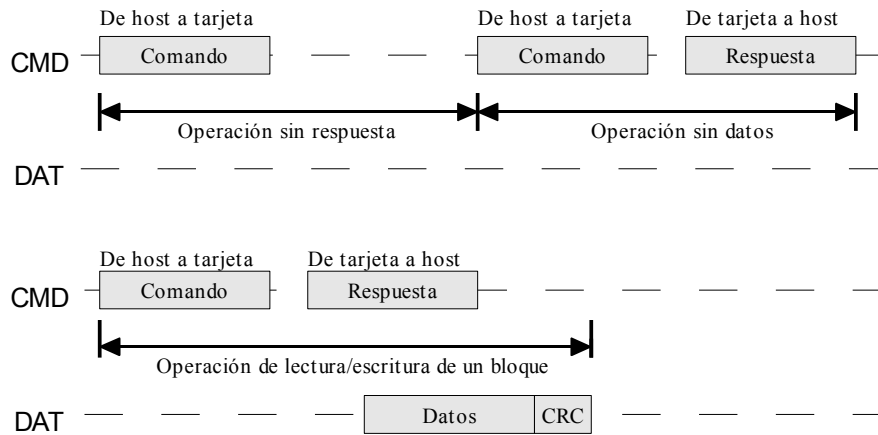


Ilustración 24: Líneas de tiempo de comandos/respuesta de SD

Como se observa en el gráfico, al ser CMD y DAT independientes una de la otra es posible comenzar una transferencia de datos cuando aún se está utilizando la línea CMD para transmitir el mensaje de respuesta al comando. Los comandos y respuestas de status SD son paquetes de 48 bits con el siguiente formato:

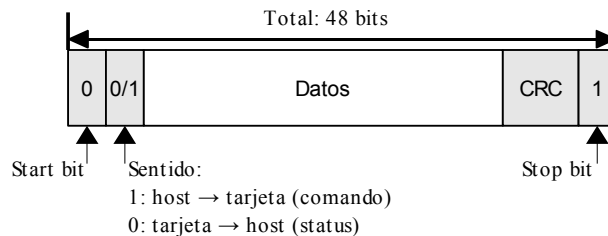


Ilustración 25: Formato de los comandos y respuestas del bus SD

Mientras que los bloques de datos se transfieren agregando al inicio un bit de start, y al final un CRC y el bit de stop:

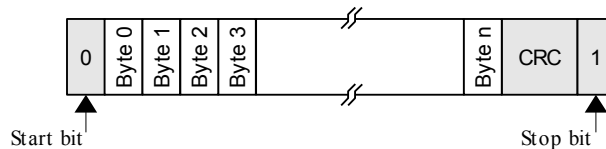


Ilustración 26: Formato de los bloques de datos de SD

Cuando las líneas de datos y comandos están en estado inactivo, se mantienen en estado lógico "1" (alto). De ahí la razón de que el bit de inicio sea un 0 lógico.

SDIO

El estándar Secure Digital se diseñó como una forma de añadir espacios de almacenamiento (las tarjetas SD) a dispositivos electrónicos, pero en el año 2000 las empresas fundadoras de la Asociación SD Card, la entidad que nuclea a las empresas que controlan el estándar SD, comenzó a trabajar en extender la especificación original para permitir conectar además tarjetas que cumplieran el rol de periféricos de entrada/salida. Este estándar se dio a llamar Secure Digital Input/Output, o SDIO.

La tarjeta SDIO está basada en (y es compatible con) la tarjeta de memoria SD. Esta compatibilidad incluye dimensiones, conexiones eléctricas, alimentación, señalización y software. El objetivo de SDIO es proveer funciones de E/S de alta velocidad y poco consumo de energía a los dispositivos electrónicos móviles. Uno de los objetivos de SDIO es que una tarjeta SDIO conectada a un host no compatible con SDIO no debe causar daño a la

circuitería, ni ninguna interrupción en el dispositivo o su software, sino que debe simplemente ser ignorada.

SDIO usa el mismo tipo de bus y los mismos modos de transferencia de información que SD, pero cambia la forma en que la tarjeta se identifica con el host al momento de la inserción y agrega comandos nuevos para hacer operaciones de lectura/escritura de registros de memoria. Una única tarjeta SDIO puede poseer hasta 7 **funciones** SDIO, equivalentes a diferentes sub-dispositivos integrados en una misma tarjeta física. Una tarjeta SDIO puede, además, integrar memoria de la misma forma que una tarjeta SD estándar en adición a las funciones de SDIO.

Durante la inicialización e interrogación de la tarjeta por parte del host, la tarjeta se identifica como un dispositivo SDIO; el software entonces puede leer una serie de tuplas enlazadas desde la memoria de la tarjeta que identifican a cada una de las funciones de la tarjeta tras lo cual puede decidir a cuál o cuáles de estas funciones activar. La decisión puede estar basada en funcionalidad requerida, presencia en el sistema operativo del host de los controladores necesarios, o simplemente por razones de consumo de energía.

En adelante se dejará de lado el aspecto de almacenamiento de las tarjetas y el bus SD, para abordar específicamente el funcionamiento de SDIO. Puede encontrarse más información acerca de las tarjetas de memoria SD en el sitio de la Asociación SD[25]

Mapa de memoria

Las tarjetas SDIO poseen un espacio de registros de I/O a través del cual el host programa a las diferentes funciones de la tarjeta. Es a través de este espacio de registros, y de los comandos que leen y escriben información en él, que el host puede identificar el tipo de tarjeta, las funciones que implementa y el modo de bus a utilizar.

El mapa de memoria de una tarjeta SDIO (que no debe confundirse con una tarjeta SD de almacenamiento) está dividido en 8 bancos de 128 KB cada uno. El banco número 0 corresponde a los descriptores de la tarjeta y sus funciones, y está dividido en las siguientes áreas:

Common I/O Area (CIA)

La CIA está presente en todas las tarjetas SDIO. Se accede a través de operaciones de lectura/escritura a la función 0. Los registros de la CIA proveen información acerca de las funciones de la tarjeta y sus requerimientos, permiten habilitar o deshabilitar cada una de esas funciones, y controlar la generación de interrupciones de hardware por parte de la tarjeta. En esta área existen tres estructuras de datos diferentes:

1. Card Common Control Registers (CCCR)
2. Function Basic Registers (FBR)
3. Card Information Structure (CIS)

Card Common Control Registers (CCCR)

Esta área permite que el host revise y controle la activación de las funciones de la tarjeta y la generación de interrupciones a nivel global y de función. Los CCCRs son una combinación de registros de información (sólo lectura) y control (lectura/escritura).

Function Basic Registers (FBR)

Además de los CCCRs, cada función de I/O soportada por la tarjeta tiene un área de 256 bytes que el host utiliza para determinar rápidamente las características y requerimientos de cada función, y para habilitar la carga del software. La dirección de cada una de estas áreas es la región comprendida entre 0x00n00 y 0x00nFF, donde n es el número de función (1 a 7)

Card Information Structure (CIS)

La CIS provee información más detallada acerca de la tarjeta y sus funciones individuales. Es el área común para leer toda la información acerca de las funciones de I/O que existen en una tarjeta. Su diseño está basada en el diseño de las tarjetas PCMCIA, que poseen una estructura similar. Todas las tarjetas que soporten I/O deben tener una CIS común y una CIS para cada función ubicadas en las áreas de memoria preestablecidas y se acceden a través de operaciones de lectura al igual que los CCCRs y FBRs. Cada uno de los FBRs tiene un puntero a la dirección donde se encuentra su CIS dentro de este espacio de memoria.

Embedded I/O Code Storage Area (CSA)

Para dar soporte al concepto de “plug-and-play” de las tarjetas SDIO, cada función puede necesitar un área de memoria donde almacenar controladores o aplicaciones. La Code Storage Area es un área de 16 MB que está pensada para almacenar esta información, y se accede utilizando operaciones de lectura/escritura sobre los FBRs de cada una de las funciones. Esto permite que cada función tenga su propia CSA y además hacer cada una de ellas independientes de la presencia o no de la funcionalidad de “SD Memory” en la tarjeta SDIO.

A continuación se presenta el mapa de memoria de una tarjeta SDIO genérica:

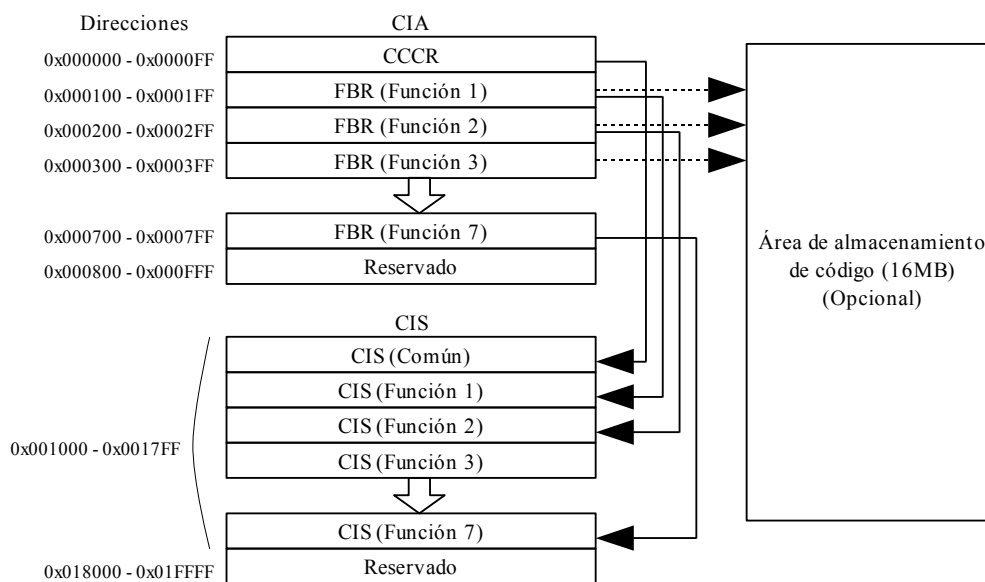


Ilustración 27: Mapa de memoria de una tarjeta SDIO

Los comandos de SDIO

SDIO posee una gran variedad de comandos que los diferentes tipos de tarjeta (SD Memory y SDIO) deben reconocer. A continuación se sumarán los comandos relevantes para SDIO, y más específicamente los relevantes para el caso que se describe.

Todos los comandos SDIO poseen un formato común de 5 bytes. A esto se le agrega un CRC de 7 bits y un octavo bit de stop. El primer byte corresponde al comando que se está transmitiendo, y de su valor depende la interpretación de los siguientes 4 bytes, que sirven como parámetros del mismo.



Ilustración 28: Formato de un comando SDIO

En el vocabulario de SD se nombra a los comandos por su código anteponiéndoles las letras

CMD. De esta forma, por ejemplo, el comando cuyo código es 0 y corresponde a la operación “Reset” se nombra CMD0. Las respuestas, por su parte, se denominan Rn, con n siendo un dígito entre 1 y 5 y se describirán cuando se hable de los comandos que las generan. Esta es la nomenclatura que se utilizará en la siguiente descripción:

Comando: CMD5

Nombre: IO_SEND_OP_COND

Este es el primer comando que un host SDIO envía a una tarjeta, y está destinado a obtener las características básicas de la misma. Tiene el siguiente formato:

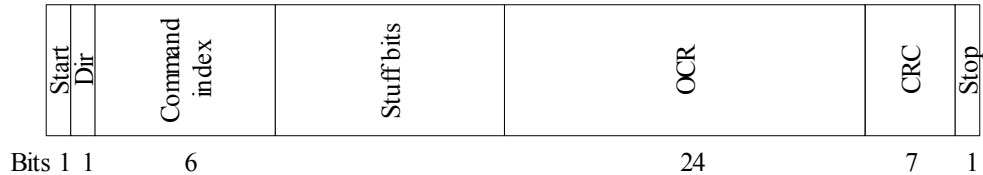


Ilustración 29: Estructura del comando CMD5

Solicita a la tarjeta que envíe una respuesta de 48 bits con el siguiente formato:

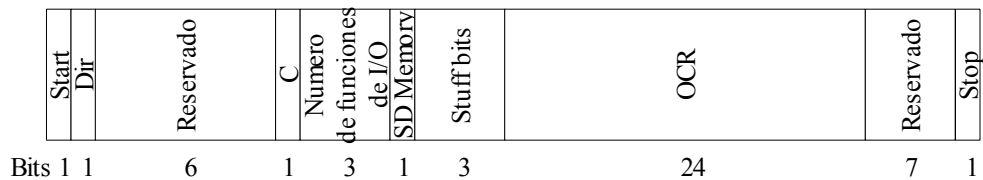


Ilustración 30: El formato de respuesta R4

Los campos son:

C: indica si la tarjeta está lista para operar (Card ready) o debe esperarse aún más para que complete su inicialización.

Número de funciones de I/O: la cantidad de funciones (1 a 7) distintas de I/O que implementa esta tarjeta.

SD Memory: si es 1, indica que la tarjeta además implementa memoria SD estándar.

OCR: codifica en cada uno de los 24 bits un rango de tensión de alimentación. La tarjeta informa a través de este campo qué valores de voltaje de entrada es capaz de manejar. Si bien SDIO requiere que todas las tarjetas sean capaces de funcionar a 3,3V, una tarjeta puede solicitar ser alimentada con 2,7V para reducir el consumo.

Comando: CMD3

Nombre: SEND_RELATIVE_ADDR

A través de este comando el host pide a la tarjeta que le envíe una dirección (Relative Card Address) por la cual el host se referirá a ella cada vez que quiera enviar o recibir información. La respuesta que envía la tarjeta se denomina R6:

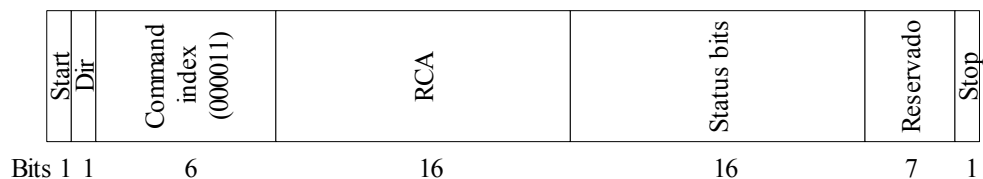


Ilustración 31: El formato de respuesta R6

La nueva dirección se envía en el campo RCA.

Los status bits codifican el estado de la tarjeta al momento de la operación como sigue:

Bits	Descripción	Valores
15	Error de CRC	0 = CRC correcto 1 = error de CRC
14	Comando ilegal	0 = comando válido 1 = error
13	Error	0 = el comando se ejecutó correctamente 1 = ocurrió un error
12:0	Estado de la tarjeta al momento de recibir el comando	0 = libre 1 = lista 2 = durante la identificación 3 = en stand-by 4 = en una transacción 5 = transfiriendo información 6 = recibiendo datos 7 = en modo programación 8 = desconectada 9 – 14 = reservados 15 = reservado para SDIO

Comando: CMD7

Nombre: SELECT_DESELECT_CARD

CMD7 permite al host direccionar una tarjeta en especial. Cuando un CMD7 es transmitido por el bus SD, todas las tarjetas lo reciben y sólo aquella cuya dirección es la seleccionada por el parámetro del comando cambia a estado activo. Las demás tarjetas pasan a estar en estado inactivo con respecto al bus SD, e ignoran cualquier otro comando siguiente con la excepción de un nuevo CMD7.

Su formato es:

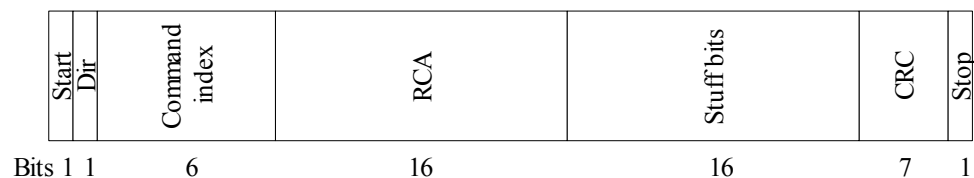


Ilustración 32: Estructura del comando CMD7

La tarjeta seleccionada transmite al host un mensaje de respuesta de status de tipo R1:

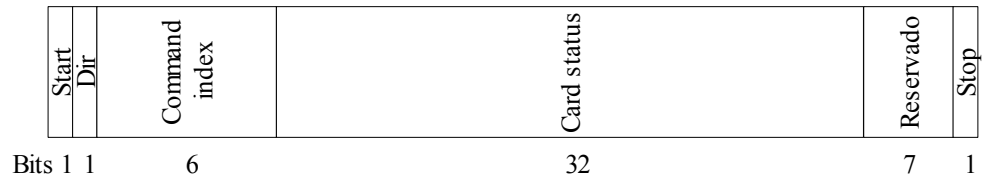


Ilustración 33: El formato de la respuesta R1

En la respuesta de tipo R1 el campo command index contiene el número de comando al que la respuesta hace referencia. En este caso es el número 7.

El valor de Card Status se fija para SDIO en 0x00000F00, que el estándar reserva para las respuestas de status de SDIO.

Comando: CMD52

Nombre: IO_RW_DIRECT

Parámetros: campo de bits.

El comando CMD52 lee o escribe un registro de un byte dentro del espacio de direcciones de 128KB de cualquiera de las 8 funciones, incluyendo la CIA. Un uso común de este comando es durante la inicialización de la tarjeta, para leer las distintas estructuras presentes en la CIA e inicializar los valores de los registros de configuración.

Los 32 bits correspondientes a los parámetros del comando se interpretan como una serie de 5 campos de información:



Ilustración 34: Parámetros del CMD52

Donde los parámetros son:

R/W: selecciona si el comando es de lectura (0) o escritura (1).

Function number: selecciona la función de la tarjeta sobre la que se hace la lectura/escritura

RAW Flag: (read-after-write) Selecciona si el valor retornado en el resultado de una operación de escritura es el valor previo a la modificación, o el valor que se acaba de escribir.

Register address: la dirección dentro del banco de 128KB del byte que se desea leer o escribir

Write data: el valor que se desea escribir en una operación de escritura. Si la operación es de lectura el dato es ignorado.

CMD52 retorna una respuesta de tipo R5:

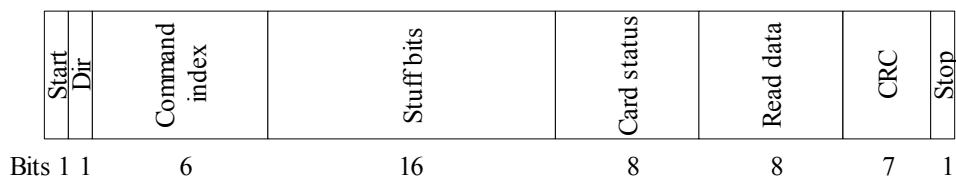


Ilustración 35: El formato de la respuesta R5

Donde el *Card Status* es un bit-field con los siguientes valores:

Bits	Descripción	Valores
7	Error de CRC	0: el CRC es correcto 1: error de CRC
6	Comando ilegal	0: el comando es correcto 1: el comando es ilegal para el estado de la tarjeta
5 – 4	Estado actual	0: tarjeta deshabilitada 1: ejecutando un comando 2: transfiriendo información 3: reservado
3	Error general	0: no hubo ningún error en la ejecución del comando 1: el comando produjo un error
2	Reservado para uso futuro	
1	Número de función	0: la función es correcta 1: la función direccionada es ilegal
0	Fuera de rango	0: los parámetros son correctos 1: alguno de los parámetros está fuera de rango

Secuencia de inicialización

Al insertar una tarjeta SDIO en un host, este responde enviando a la tarjeta un CMD5 para obtener los parámetros operativos de la tarjeta. En este comando el host llena el campo OCR del comando con sus propias condiciones de operación.

Si al recibir la respuesta de la tarjeta el host determina que puede satisfacer sus condiciones de operación, envía un segundo CMD5 donde el campo OCR es el resultado de hacer un AND lógico entre el OCR del host y el retornado por la tarjeta al comando anterior.

A continuación el host envía un CMD3 para conocer la dirección por la que se referirá a la tarjeta. Si la dirección retornada pertenece a una tarjeta ya presente en el host esta operación se repite, retornando la tarjeta una dirección diferente cada vez, hasta que la duplicación de direcciones desaparezca.

Cuando el host ha obtenido una dirección (RCA) de la tarjeta le envía un CMD7 con su dirección para seleccionarla. Luego lee todos los registros de información presentes en la CIA byte a byte, utilizando CMD52, para obtener los datos necesarios de fabricante, tipo y modelo de tarjeta necesarios para cargar el controlador o los controladores correspondientes.

El resto de la inicialización de la tarjeta es específico del modelo particular y está a cargo del controlador de la tarjeta presente en el sistema operativo del host.

Implementación del bus SDIO sobre el CC2431

A continuación se describirá cómo se implementó la funcionalidad de tarjeta SDIO descrita anteriormente utilizando el hardware del CC2431, tanto desde el punto de vista físico como del stack de SDIO creado para correr en el microcontrolador.

El hardware

Para la implementación de la comunicación física de los bits se utilizaron los dos transmisores/receptores seriales síncronos/asíncronos (USARTs) que provee el chip CC2431. Estos módulos funcionan independientemente y es posible utilizarlos de forma concurrente. Sus características principales son:

- Soporte de palabra de 8 ó 9 bits.
- Paridad par, impar o ninguna.
- Nivel de bits de inicio y parada configurables.
- Pueden transmitir/recibir en orden LSB⁷ primero, o MSB⁸ primero.
- Interrupciones independientes para transmisión y recepción.
- Triggers de DMA independientes para transmisión y recepción.
- Detección de errores de paridad y framing automáticos.

Para la implementación de SDIO utilizada en Kindergarten se usó la USART0 para el manejo de la línea CMD de SDIO, y la USART1 para el manejo de la línea DAT. Ambas interfaces se configuraron para actuar como puertos seriales síncronos SPI para descargar sobre el hardware la tarea de envío y recepción de bits individuales.

SPI y su adaptación a las conexiones SDIO

El protocolo SPI (Serial Peripheral Interface) es un estándar creado por Motorola para un bus serie síncrono full-duplex que utiliza cuatro líneas de datos unidireccionales para comunicar dos o más dispositivos electrónicos. SPI es un bus asimétrico, donde un dispositivo master controla el bus y habilita o inhabilita a uno o más dispositivos esclavos que lo comparten.

Las líneas de datos que forman el bus SPI son:

Línea	Sentido	Descripción
MOSI	Master → slave	Master Output Slave Input: a través de esta línea el dispositivo master envía los bits a los dispositivos esclavos
MISO	Slave → master	Master Input Slave Output: por esta línea los dispositivos esclavos envían la información al master
SCK	Master → slave	Serial Clock: el reloj maestro del bus.
SS	Master → slave	Slave Select: esta señal actúa como línea de direccionamiento de dispositivo para el caso en que múltiples esclavos compartan un mismo bus. Sólo el dispositivo cuyo SS lo habilita puede recibir la información que transmite el master, y transmitir la suya. Si el dispositivo esclavo no es habilitado por su SS, debe permanecer “desconectado” eléctricamente del bus.

El siguiente diagrama muestra un bus SPI compartido por un dispositivo master y dos slaves:

7 Least Significant Bit, bit menos significativo

8 Most Significant Bit, bit más significativo

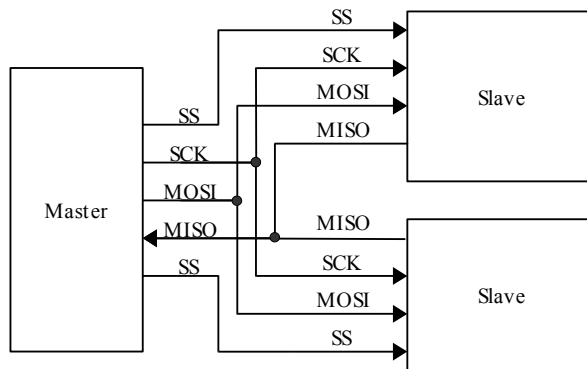


Ilustración 36: Diagrama de bus SPI con dos dispositivos esclavos

Los puertos SPI se conectaron back-to-back con sí mismos para generar de cada par de líneas unidireccionales (MISO Y MOSI) una única línea bidireccional (CMD y DAT). Se usaron dos bits de entrada/salida de propósito general para servir como Slave Select de ambos puertos SPI de forma de controlar a través del software la habilitación de las interfaces de hardware:

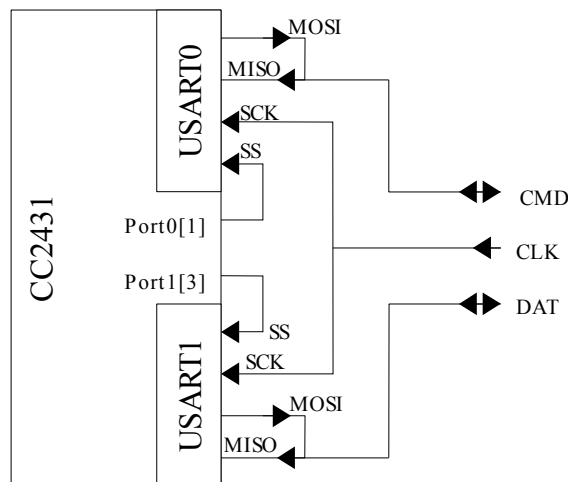


Ilustración 37: Conexión de los puertos SPI para simular una interfaz SDIO

El ciclo de trabajo de la interfaz SDIO exige que ambos puertos SPI y sus correspondientes pines de E/S cambien de estado, función y dirección varias veces durante la recepción de un comando desde el host y la transmisión de la respuesta. En los próximos párrafos se describirá en forma detallada este proceso.

Recepción de un comando transmitido desde el host

En su estado de reposo, el hardware de la tarjeta SDIO emulada por el CC2431 se encuentra en lo que se denomina un estado de alta impedancia. En este estado, si bien el microcontrolador está físicamente cableado al bus, sus líneas de datos de entrada y salida se encuentran desconectadas electrónicamente de los terminales. En este estado las líneas de datos de salida no conducen electricidad desde o hacia el microcontrolador, y las líneas de entrada no drenan energía del host.

En modo de reposo los bits que comandan las líneas Slave Select de ambos puertos SPI, el bit 1 del port 0 (Port0[1]) y el bit 3 del port 1 (Port1[3]), mantienen a los puertos en estado inactivo. La línea MISO del USART0 se asigna al puerto de GPIO y se programa como fuente de interrupción en el flanco descendiente de la línea CMD de SDIO.

Al empezar el host a transmitir un comando, la transición 1 → 0 de la línea de CMD correspondiente al start-bit del comando dispara la interrupción asociada a la línea MISO. En este momento el microcontrolador pasa a ejecutar el manejador de la interrupción de inicio de

comando. Esta rutina hace los siguientes cambios en los puertos:

- Desconecta la línea MISO del puerto SPI de la USART0 asignando el pin a la función de GPIO
- Configura el puerto GPIO en alta impedancia, para que el CC2431 no afecte el estado lógico de la línea CMD
- Conecta el pin de clock de SDIO a la entrada SCK del puerto SPI de la USART0
- Activa la línea SS del puerto SPI para hacer que la USART0 comience a recibir los bits del comando.
- Activa la interrupción de “byte recibido” de la USART0

Todos estos pasos consiguen conectar la USART0 a la línea CMD de SDIO, ponerla en modo recepción y hacer que el mismo hardware notifique a la CPU cada vez que se reciba un byte. El manejador de interrupción de “byte recibido” para la USART0 simplemente almacena en un buffer global el comando a medida que se va recibiendo byte a byte, y al completar la recepción vuelve la USART0 a su estado de reposo. Al terminar de recibir un comando, el manejador de interrupción invoca al resto del stack SDIO para que procese el comando.

Si el comando genera una respuesta, la transmisión de la misma se hace configurando la USART0 de forma que todas sus conexiones (MISO, MOSI y SCK) queden conectadas a la línea de datos CMD. El software de Kindergarten programa el puerto SPI para transmitir la información de la respuesta y lo alimenta byte a byte a medida que se transmite. El hardware se encarga entonces de serializar los bits de la respuesta.

La transferencia de datos se hace por un mecanismo similar, por lo que no será abordada aquí.

Stack de SDIO en el CC2431

El stack de SDIO creado para Kindergarten y que corre en el CC2431 está dividido en tres módulos principales:

- Hardware Abstraction Layer
- Memory Space
- Core

La HAL

La Hardware Abstraction Layer (HAL) de SDIO está contenida en el archivo `sdio_hal_dual_uart.c` y es un conjunto de funciones destinado a aislar al núcleo de SDIO de los detalles del hardware en particular. La HAL se encarga de enviar y recibir comandos y respuestas, y de realizar transferencias de datos.

Las funciones más importantes de la HAL son:

```
void sdio_hal_init(void)
```

Como su nombre lo indica, inicializa el hardware y lo deja a la espera de un comando. Esta función coloca las dos USARTs del CC2431 en espera de cualquier transferencia, las inicializa en modo SPI y hace la inicialización de las dos interrupciones que utiliza para atender a los eventos asíncronos.

```
__interrupt void sdio_hal_receive_command_from_int(void)
```

Esta es la función que es invocada por el microcontrolador cuando se recibe la transición 1 → 0 en la línea CMD de la interfaz SDIO, marcando el comienzo de un comando.

```
__interrupt void receive_command_byte_from_int(void)
```

receive_command_byte_from_int es la función invocada cuando la USART0 recibe un byte.

```
void sdio_hal_send_response(BYTE length, BYTE r1b, BYTE  
waitclocks);
```

```
void sdio_hal_send_response_with_data(BYTE response_length,  
                                     BYTE length, BYTE r1b, BYTE waitclocks);
```

Este par de funciones se encarga de enviar una respuesta al host SDIO, con o sin un paquete de datos asociado. En ambos casos la operación es bloqueante.

El módulo de memory space (memspace) está contenido en el archivo sdio_memspace.c y representa el espacio de memoria de la tarjeta, comprendiendo la CIA (los CCCRs, los FBRs de la tarjeta, la CIS de la tarjeta y la CIS de la función implementada) y el espacio de memoria de la función implementada.

El módulo de memory space implementa una única función, sdio_memspace_read_write(), que toma un comando de tipo CMD52 (IO_RW_REGISTER_DIRECT) recibido por la tarjeta, decodifica la función (0 a 7), operación (lectura / escritura) y la dirección contenidas en él, y accede a la memoria simulada para obtener o cambiar el byte al que se hace referencia.

El Core

En el core de SDIO está definido en el archivo sdio.c y en él están las funciones que interpretan los comandos y crean los distintos tipos de respuesta. En este módulo está toda la lógica de inicialización de la tarjeta SDIO que implementa el software sobre el CC2431, y las rutinas que disparan cada uno de los comandos del host SDIO.

Las dos funciones más destacadas de este módulo son:

```
sdio_startup()
```

Realiza toda la secuencia de inicialización y registro de la tarjeta SDIO con el sistema host según lo explicado más arriba en “Secuencia de inicialización”

```
sdio_process_command()
```

El punto de entrada de la segunda capa del stack de SDIO, esta función es invocada desde receive_command_byte_from_int() de la HAL cada vez que un comando haya terminado de ser recibido. Es la encargada de decodificar el número de comando y delegar en subrutinas cada comando en particular.

Capítulo 8

La comunicación cableada de las estaciones base

En este capítulo se abordará la comunicación entre estaciones base, describiéndose los elementos de hardware y software que intervienen en el proceso.

Como se ha hablado en el capítulo 4, Kindergarten requiere que las estaciones base que están fijas en el ambiente se comuniquen entre sí y con un coordinador central de forma eficiente y sencilla. Debido a que una red Kindergarten puede abarcar áreas extensas, potencialmente edificios enteros con múltiples sub-redes, uno de los requisitos para el mecanismo de comunicaciones fue su escalabilidad.

El controlador enc28J60

Para la comunicación entre estaciones base se eligió utilizar la red Ethernet por su ubicuidad, gran velocidad y confiabilidad. Esto obligó a buscar un medio para comunicar el microcontrolador CC2431 de las estaciones base con el cableado UTP de la red Ethernet. Para este rol se eligió utilizar el chip ENC28J60 de la empresa Microchip, un controlador Ethernet especialmente desarrollado para aplicaciones como la aquí descrita.

Sus principales características son:

- Controlador Ethernet compatible con la norma IEEE 802.3
- Integra las capas MAC y PHY para 10 BASE-T
- Soporta cableado UTP con detección y corrección automática de polaridad
- Soporta conexiones semi-dúplex y full-duplex
- Se comunica a través de SPI con velocidades de hasta 20 Mhz
- Posee un buffer de datos de 8KB de RAM dual-port⁹
- Integra un controlador de DMA para movimiento de datos
- Posee 6 fuentes de interrupción
- Permite filtrar tramas unicast, multicast o broadcast independientemente

La gran ventaja del ENC28J60 es que no necesita ningún tipo de interfaz extra para comunicarse con el CC2431. Sólo es necesario cablear las 4 líneas de la interfaz SPI entre ambos circuitos para disponer de una comunicación de datos entre ambos.

Microchip también entrega como parte de su documentación un conjunto de funciones escritas en lenguaje C que realizan el envío y recepción de la información a través del ENC28J60, de manera que los desarrolladores que utilicen este chip no necesitan programar su propio controlador. Este código se encuentra en el archivo `enc28j60.c`.

⁹ La RAM dual-port posee dos buses de datos y direcciones independientes, permitiendo efectivamente dos accesos simultáneos a la misma memoria, por ejemplo para que la electrónica de la PHY escriba los datos de una trama que se está recibiendo mientras el host lee los datos de otra trama ya recibida.

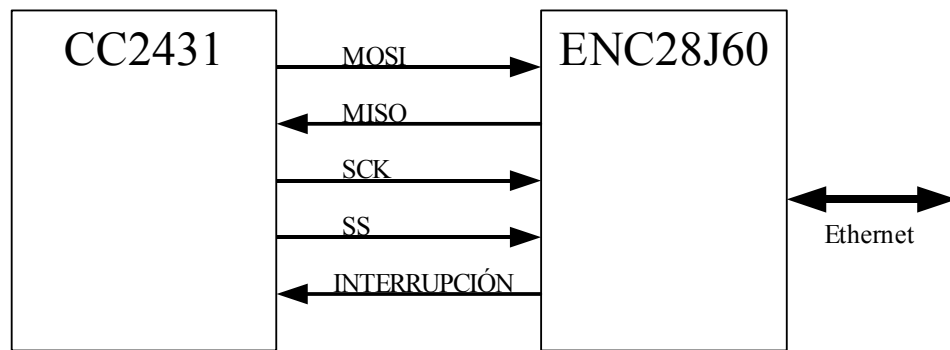


Ilustración 38: Conexión entre el CC2431 y el ENC28J60

Además de las líneas estándar de SPI, el ENC28J60 incluye una línea de interrupción a través de la cual informa a la CPU del CC2431 de eventos tales como la recepción de una trama o una condición de error.

μIP

El uso del ENC28J60 permite al software de la estación base conectarse a la red Ethernet y enviar y recibir tramas de datos a través de ella. Esto es de una utilidad enorme para comunicar sistemas similares entre sí por su gran simplicidad: basta encapsular la información en una trama Ethernet y enviarla a la dirección de destino.

Por desgracia este enfoque tiene dos limitaciones: la primera es que al encapsular información en tramas Ethernet puras se está asumiendo que toda la red Kindergarten estará vinculada por la misma capa de switching. Dicho de otra forma, todas las estaciones base deben estar conectadas a uno o más switches ethernet conectados entre sí, con lo que es imposible aislar dominios de broadcast si la misma red Ethernet es compartida por equipos informáticos tradicionales.

La segunda desventaja involucra al sistema del nodo coordinador: los entornos de alto nivel, como el empleado para implementar el algoritmo de routing del nodo coordinador, no permiten interactuar a tan bajo nivel con las capas de red. En ellos es mucho más conveniente utilizar las APIs de TCP/IP.

Estas razones llevaron a buscar un medio por el que las estaciones base de Kindergarten fueran capaces de comunicarse utilizando protocolos de más alto nivel; específicamente TCP/IP.

Al igual que en el caso del controlador de bajo nivel del ENC28J60 se buscó una solución ya desarrollada para el stack de TCP/IP, encontrándose esta en el paquete “μIP” desarrollado en el Instituto Sueco de Ciencias de la Computación (SICS). μIP es un stack de TCP/IP especialmente pensado y desarrollado para microcontroladores y microprocesadores de 8 bits, bajo licencia Open Source.

Sus principales características son:

- Implementación de protocolos TCP e IP estándar, incluyendo control de flujo, fragmentación y reensamblado, y retransmisión por time-out.
- Tamaño de código muy reducido.
- Muy bajo uso de RAM, configurable en tiempo de compilación.
- Soporte de protocolos ARP, SLIP, IP, UDP, ICMP (ping) y TCP.
- Soporta cualquier número de conexiones TCP simultáneas, hasta un máximo configurable en tiempo de compilación.
- Soporta cualquier número de escuchas de conexiones TCP (servidores TCP) hasta un máximo configurable en tiempo de ejecución.
- Libre para uso comercial y no comercial.

Integración de μ IP con el stack de Kindergarten

A través de μ IP la estación base establece la escucha por datagramas UDP en el puerto 45000. El coordinador central posee la lista de direcciones IP o hostnames de todas las estaciones base y utiliza este número de puerto para enviarles mensajes Kindergarten como se describe en el capítulo 4.

Cuando una trama Ethernet es recibida por el ENC28J60, éste envía al microcontrolador un pedido de interrupción a través de la línea de interrupción, conectada a uno de los pins del PORT1. Esta línea dispara la interrupción correspondiente en el núcleo de CPU que invoca a la función `heartbeat_interrupt_handler()`, debido a que las interrupciones de heartbeat y de evento de Ethernet comparten el mismo vector de interrupción. Dentro de esta función se determina cuál es la fuente de la misma y se invoca al handler correspondiente. En el caso de las interrupciones disparadas por el ENC28J60, esa función es `ethernet_packet_received()`.

`Ethernet_packet_received()` invoca a las funciones provistas en el archivo `enc28j60.c` para transferir la trama recibida desde la memoria del cotrolador Ethernet a la memoria del microcontrolador CC2431, y decide en base al tipo de trama declarado en la cabecera Ethernet (IP o ARP) a qué punto de entrada dentro del stack μ IP debe invocar. Las funciones llamadas son `uip_arpin()` en caso de que se reciba una trama ARP y `uip_input()` si es una trama IP.

μ IP trabaja con dos variables globales (`uip_buf` y `uip_len`) que almacenan una trama y su longitud, y sobre este buffer hace las operaciones que sean necesarias. En el caso de recibir una solicitud de ARP, μ IP lee la dirección IP de la solicitud y responde con el mensaje correspondiente en caso de que fuera dirigida a la propia estación. En caso de recibir una trama IP determina el tipo de capa de enlace y hace el callback que corresponde a las funciones de espacio de usuario, terminando por emitir un mensaje de `acknowledge` en las conexiones TCP.

Capítulo 9

Creación y resultados de una red Kindergarten real

En este capítulo se describe la creación de una red Kindergarten para experimentación, llevada a cabo en la Facultad de Informática, sobre la base del hardware y software descritos en los capítulos anteriores. Se hará una reseña de los modelos de hardware adquiridos para el proyecto y de los circuitos desarrollados específicamente.

Como se ha descrito en los capítulos anteriores, se eligió como base de hardware para el desarrollo de Kindergarten al chip CC2431 de la empresa Texas Instruments. En el caso de la PDA donde se ejecuta el sistema context aware se eligió una PocketPC HP iPaq rx5900.

Se adquirió un kit de desarrollo provisto por la misma empresa Texas Instruments como plataforma base. Dicho kit incluye 10 ejemplares de CC2431 instalados en circuitos impresos (llamados Evaluation Boards) y 10 adaptadores de esos circuitos a hileras de pines para fácil acceso a los pines de E/S, que reciben el nombre de Battery Boards. Además el kit incluye dos placas programadoras que se conectan por un lado a los Battery Boards y por otro a la PC a través del bus USB; permitiendo la programación del microcontrolador y el debugging in-circuit del programa.

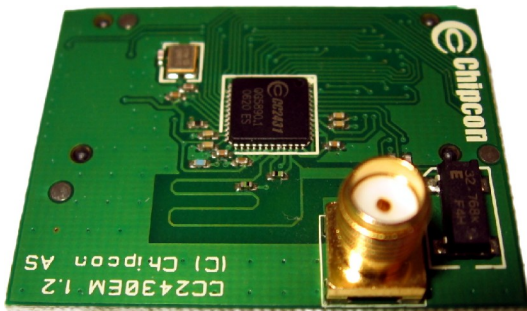


Ilustración 39: Evaluation board del chip CC2431

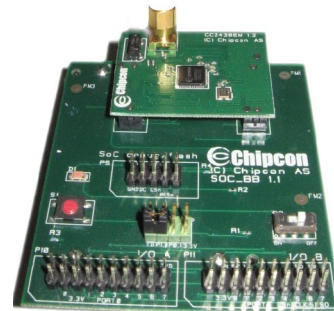


Ilustración 40: Battery board del chip CC2431

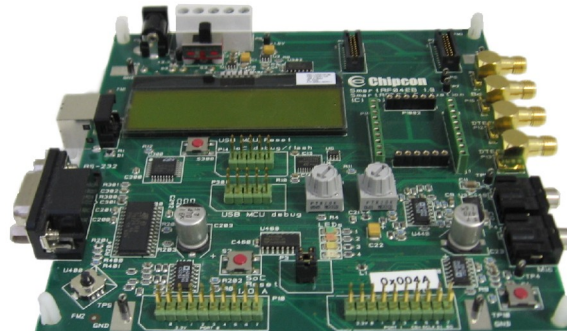


Ilustración 41: Programador de los evaluation boards del chip CC2431

Para la conexión de los Evaluation Boards a la PDA se desarrolló un adaptador ad-hoc consistente en una pieza de circuito impreso con la forma y conexiones de una tarjeta SD, junto a un cable que lo une al Evaluation Board. Este adaptador no tiene electrónica propia, siendo simplemente una extensión de las líneas de datos y energía del microcontrolador. A través del adaptador el CC2431 no sólo se comunica, sino que además recibe energía del slot SD de la PDA.



Ilustración 42: Adaptador CC2431 - SDIO, junto a una placa Wi-Fi SDIO para comparación

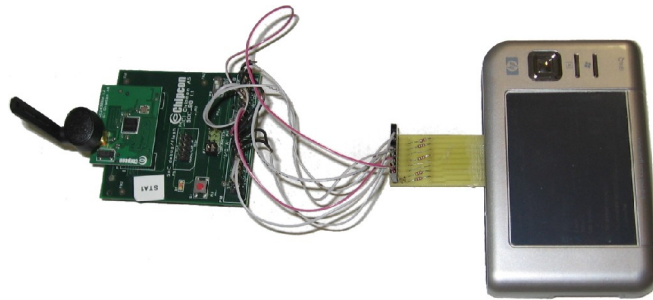


Ilustración 43: Prototipo de estación móvil conectada a una PocketPC iPaq

Para la conexión de las estaciones base a la red debió realizar un trabajo similar a las estaciones móviles: se adquirieron cuatro adaptadores ENC28J60-H a la empresa Olimex. Estos adaptadores incluyen el chip ENC28J60 y toda la electrónica necesaria para su funcionamiento, incluyendo el conector Ethernet que aloja al cable UTP. Sus únicos requerimientos son una alimentación de 3,3V y las señales de datos y reloj discutidas con anterioridad. El módulo está listo para insertarse sobre un conector hembra de 10 pines.

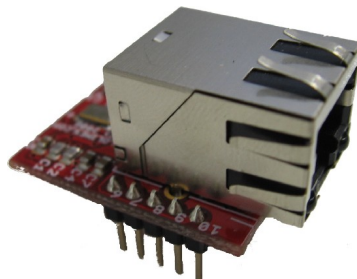


Ilustración 44: Módulo ENC28J60-H

La conexión entre el Battery Board del CC2431 y el módulo ENC28J60-H es llevada a cabo por una placa desarrollada específicamente por el autor del presente trabajo. Consiste simplemente en un adaptador entre los pines del Port1 del Battery Board al conector donde se inserta el ENC28J60-H. El mismo adaptador incluye la circuitería que se conecta a la línea de heartbeat y la dirige al pin correspondiente del CC2431.

una prueba de campo del subsistema de localización de las estaciones móviles. Se creó una red de 4 estaciones base en un espacio despejado de 10 por 10 metros. Las estaciones se colocaron en las cuatro esquinas del área.

Para la prueba se utilizó la estación móvil mostrada en la ilustración 43, con un Battery Board de CC2431 más una PDA HP rx5900. Se procedió a recorrer el área por su perímetro y registrar las posiciones estimadas por el software y hardware de Kindergarten. Los resultados se muestran en el siguiente gráfico como un ploteo del recorrido detectado por el sistema de localización junto con el recorrido real:

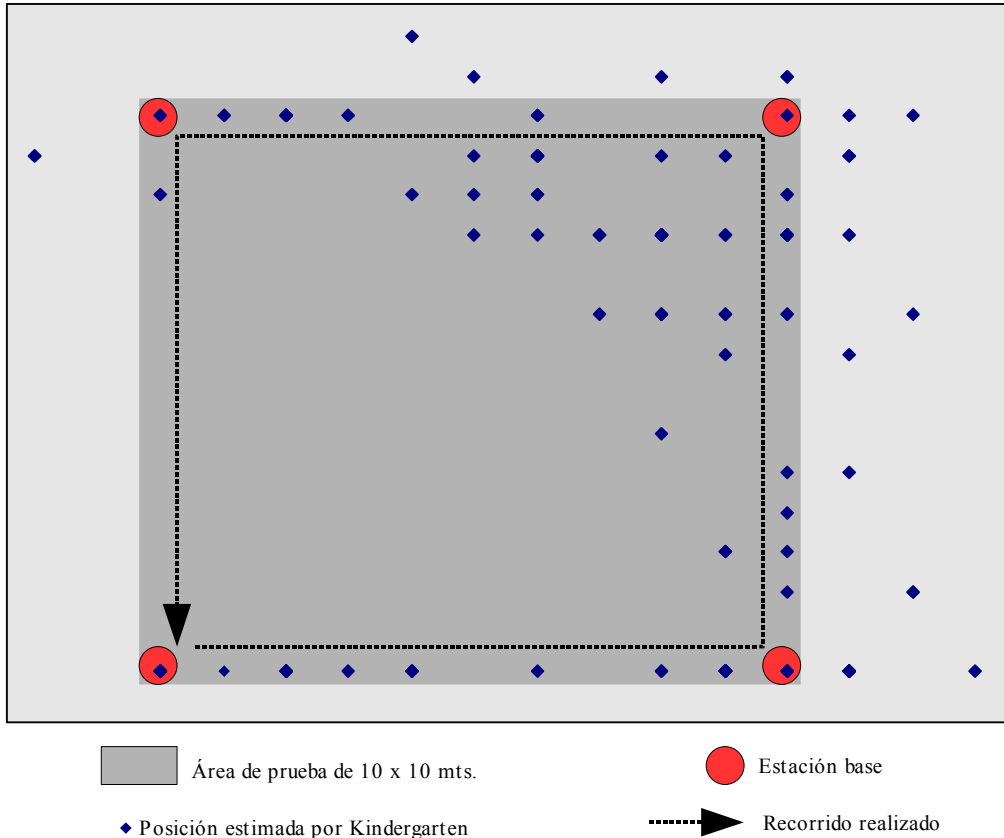


Ilustración 48: Diagrama de la prueba de campo realizada con Kindergarten

Como puede observarse la posición estimada no se corresponde exactamente con la real debido a errores de medición e interferencias, aún utilizando el filtrado rudimentario descrito en el capítulo 6. Sin embargo los errores con sólo 4 estaciones en esta primera prueba son menores a $\pm 2,5$ metros y están perfectamente dentro de lo esperado.

Capítulo 10

Conclusiones

*A learning experience is one of those things that tells you
"You see what you just did? Don't ever do that again."
Douglas Adams, "The salmon of doubt"*

En el presente capítulo se realiza un resumen final de lo expuesto en los capítulos anteriores de este trabajo de grado. Se mencionan los resultados obtenidos por la implementación de la red Kindergarten realizada a efectos de prueba, y se exploran posibles rumbos para la extensión y mejora de la red.

En los últimos años ha habido un creciente interés por lograr que las aplicaciones adapten su comportamiento a diversos factores externos. En respuesta a esta motivación han surgido diversas áreas de investigación como personalización y aplicaciones sensibles al contexto.

En el presente trabajo de grado se realizó una introducción a las aplicaciones sensibles al contexto, su importancia en el mundo real. Se describió el diseño, desarrollo e implementación de una red de comunicaciones y localización inalámbricas especialmente diseñada con el propósito de soportar aplicaciones sensibles al contexto, llamada Kindergarten.

Se describió el hardware sobre el que corre el protocolo de red, el microcontrolador CC2431 elegido por su capacidad de estimación de posición, y se hizo una reseña de la forma en que dicho microcontrolador se comunica a través de su módem de radio interno. También se explicaron los conceptos básicos de la norma de comunicaciones inalámbrica IEEE 802.15.4 que sirve de base a Kindergarten.

A continuación se habló de la red Kindergarten. Se mencionaron los requerimientos de la red, y las decisiones de diseño tomadas para satisfacerlos. Se describió el funcionamiento de cada uno de los integrantes de la red: el coordinador, la estación base y la estación móvil, y la forma en que el protocolo Kindergarten los mantiene sincronizados para aprovechar todo el ancho de banda de la red. También se describió el mecanismo por el cual cada una de las estaciones móviles es capaz de estimar su posición a través del módulo de hardware integrado en el CC2431 a tal efecto.

En el siguiente capítulo se detalló la forma en que el microcontrolador CC2431 que actúa como controlador de Kindergarten en las estaciones base se comunica con el sistema PDA o Smartphone sobre el que corre la aplicación context-aware: el bus SDIO. Se hizo una introducción al bus SD mencionando sus características básicas y se explicó la estructura de comunicación y comandos pertinentes del bus SDIO, detallándose la forma en que se utilizó el hardware del CC2431 para simular el hardware de una tarjeta SDIO.

Por último se mencionó el mecanismo de hardware y software sobre el que se comunican las estaciones base entre sí, y se describió el chip controlador de Ethernet y el stack de TCP/IP utilizado.

Trabajo futuro

Durante el desarrollo del tema que abarca este trabajo de grado se creó la implementación física y lógica de la red Kindergarten descrita en los capítulos anteriores. Sin embargo resulta demasiado complejo intentar una solución total y óptima en una primera iteración de desarrollo. En base a esto es posible considerar múltiples oportunidades de mejora en el futuro, las cuales incluyen;

- **Performance:** mucho trabajo queda aún hasta hacer que todos los mecanismos de la red funcionen óptimamente, sobre todo en las áreas de optimización de algoritmos,
- **Precisión:** debe evaluarse si la implementación por defecto del mecanismo de estimación de posición que provee el hardware del microcontrolador CC2431 es adecuada para las necesidades, y llegado el caso mejorarse o reemplazarse por otra para hacerla más precisa y confiable.

- **Filtrado de la información de localización:** relacionado con el punto anterior; en este momento todas las mediciones de nivel de señal recibida se promedian como se describe en el capítulo 6 y ese promedio se pasa al estimador. Deberá explorarse la posibilidad de hacer filtrado estadístico de los valores medidos para descartar mediciones erróneas.
- **Más estaciones base:** resta por analizar el efecto que tendrá el uso de más de 4 estaciones base por área cubierta. Se espera que la precisión y confiabilidad del sistema aumente enormemente con el uso de 6 u 8 estaciones base como referencia de posición.
- **Ahorro de energía:** el chip CC2431 provee características avanzadas de ahorro de energía que pueden aprovecharse para bajar aún más el consumo. Un ejemplo es entrar en modos de bajo consumo mientras el programa controlador de Kindergarten no reciba eventos y se encuentre ejecutando el lazo de control del programa principal.
- **Benchmarking:** debe analizarse el throughput final y real de la red en condiciones normales de utilización para poder determinar cómo influyen las demoras y pérdidas de señal en la confiabilidad del protocolo.
- **Ampliación de los servicios:** en el coordinador central podría actuar como un servicio de directorio para las estaciones móviles, o como un application server que proveería servicios a las aplicaciones context aware.

Bibliografía

- 1: *Estándares oficiales de telefonía 3G*, <http://www.3gpp.org/specifications>
- 2: *Procesador Intel ATOM*, <http://www.intel.com/technology/atom/index.htm>
- 3: Schilit, B., *Context-aware computing applications*, Proceedings of the Workshop on Mobile Computing Systems and Applications, 1994
- 4: Anind K. Dey, *Providing architectural support for building context-aware applications*, tesis doctoral, 2000
- 5: Virrantaus K., Markkula J., Garmash A., Terziyan V., Veijalainen J., Katasonov A., Tirri H., *Developing GIS-Supported Location-Based Services*, Proceedings of the Second International Conference on Web Information Systems Engineering, 2001
- 6: Huang, A., *A privacy conscious bluetooth infrastructure for location aware computing*, <http://people.csail.mit.edu/albert/pubs/2004-albert-infrastructure-forlocation-aware-computing.pdf>
- 7: *Estándar IEEE 802.11*, <http://standards.ieee.org/wireless/standards.html>
- 8: *Especificación de Bluetooth*, <http://www.bluetooth.com/Bluetooth/Technology/Works/>
- 9: *Historia del sistema GPS*, <http://www.kowoma.de/en/gps/history.htm>
- 10: *Especificaciones de IrDA*, <http://www.irda.org/displaycommon.cfm?an=1&subarticlenbr=7>
- 11: *Patente de la primera aplicación de offloading de procesamiento*, <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fmetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=5355453.PN.&OS=PN/5355453&RS=PN/5355453>
- 12: *Familia Picowatt de Atmel*, <http://www.atmel.com/ad/picopower/>
- 13: *Atmel's AVR Microcontroller Ships 500 Million Units*, <http://www.atmel.com/dyn/resources/press/ships500m.html>
- 14: *Historia del microcontrolador 8051*, http://en.wikipedia.org/wiki/Intel_8051
- 15: *Especificación del CC2431*, <http://focus.ti.com/docs/prod/folders/print/cc2431.html>
- 16: *IEEE 802.15.4 Protocol Specification*, <http://standards.ieee.org/getieee802/download/802.15.4a-2007.pdf>
- 17: *Spread-spectrum techniques for wireless communication*, Signal Processing Magazine, 1997
- 18: Shafi, M. Ogose, S. Hattori, T., *Wireless Communications in the 21st Century*, Wiley & Sons, 2002
- 19: Allen, B., *Ultra-Wideband Wireless Sensor Networks*, IEE Seminar on Ultra Wideband Communications Technologies and System Design, 2004
- 20: Berni, A. J., & Gregg, W. D., *On the utility of chirp modulation for digital signaling*, IEEE Transactions on Communications, 1973
- 21: *Zigbee Alliance*, <http://www.zigbee.org>
- 22: *Estándar IEEE 802.3*, <http://www.ieee802.org/3/>
- 23: *Estándar IEEE 802.5*, <http://www.ieee802.org/5/www8025org/>
- 24: Westman, H.P., *Reference Data for Radio Engineers*, Howard W. Sams and Co., 1968
- 25: *SD Technology*, <http://www.sdcard.org/developers/tech/>