

**Novática**, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** co-edita, asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE** European **NET**work).

<<http://www.ati.es/novatica/>>  
 <<http://www.ati.es/reicis/>>  
 <<http://www.cepis.org/upgrade/>>

**ATI** es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC**, **RITSI** e **HispaLinux**, junto a la que participa en **ProInnova**.

**Consejo Editorial**

Ignacio Aguillo Sousa, Guillem Aínsa González, María José Escalona Cuaremas, Rafael Fernández Calvo (presidente del Consejo), Jaime Fernández Martínez, Luis Fernández Sanz, Didac Lopez Viñas, Celestino Martín Alonso, José Onofre Montes Andrés, Francesc Noguera Puig, Ignacio Pérez Martínez, Andrés Pérez Payeras, Viktu Pons i Colomer, Juan Carlos Vilgo López

**Coordinación Editorial**

Llorenç Pagés Casas <pages@ati.es>

**Composición y autoedición**

Jorge Llácer Gil de Ramales

**Traducciones**

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>

**Administración**

Tomas Brunete, María José Fernández, Enric Camarero, Felicidad López

**Secciones Técnicas - Coordinadores**

**Acceso y recuperación de la Información**

José María Gómez Hidalgo (Optenet), <mgomez@yaho.com>

Manuel J. María López (Universidad de Huelva), <manuel.mana@diehsia.uhu.es>

**Administración Pública electrónica**

Francisco López Crespo (MAE), <flc@ati.es>

**Arquitecturas**

Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>

Jordi Tubella Morgados (DAC-UPC), <jordit@ac.upc.es>

**Auditoría SITIC**

Marina Touriño Troitiño, <marinatourino@marinatourino.com>

Manuel Palao García-Suelto (ATI), <manuel@palao.com>

**Derecho y tecnologías**

Isabel Hernández Collazos (Fac. Derecho de Donostia, UPV), <isabel.hernando@ehu.es>

Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

**Enseñanza Universitaria de la Informática**

Cristóbal Pareja Flores (DSIP-UJM), <cpajef@dsip.ujm.es>

J. Ángel Velázquez Turbide (DLSI, URJC), <angel.velazquez@urjc.es>

**Entorno digital personal**

Andrés Marín López (Univ. Carlos III), <amarin@ui3m.es>

Diego Gachet Pérez (Universidad Europea de Madrid), <gachet@uem.es>

**Estándares Web**

Encarna Quesada Ruiz (Virati), <encarna.quesada@virati.com>

José Carlos del Arco Prieto (TIC Sistemas e Ingeniería), <jcarco@gmail.com>

**Gestión del Conocimiento**

Juan Baiget Solé (Cap Gemini Ernst & Young), <jbaiget@ati.es>

**Informática y Filosofía**

José Ángel Olivas Varela (Escuela Superior de Informática, UCLM), <joseangel.olivas@uclm.es>

Karim Gherab Martin (Harvard University), <kgherab@gmail.com>

**Informática Gráfica**

Miguel Chover Selles (Universitat Jaume I de Castellón), <chover@lsi.uji.es>

Roberto Vivó Hernández (Eurographics, sección española), <rvivo@dsic.upv.es>

**Ingeniería del Software**

Javier Dolado Cosin (DLSI-UPV), <dolado@si.ehu.es>

Daniel Rodríguez García (Universidad de Alcalá), <daniel.rodriguez@uah.es>

**Inteligencia Artificial**

Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV), <vboti.vinglada@dsic.upv.es>

**Interacción Persona-Computador**

Pedro M. Latorre Andrés (Universidad de Zaragoza, AIPO), <platorre@unizar.es>

Francisco L. Gutiérrez Veja (Universidad de Granada, AIPO), <fgutierrez@ugr.es>

**Lengua e Informática**

M. del Carmen Ugarte García (ATI), <cugarte@ati.es>

**Lenguajes Informáticos**

Oscar Belmonte Fernández (Univ. Jaime I de Castellón), <bellm@lsi.uji.es>

Inmaculada Coma Tany (Univ. de Valencia), <inmaculada.coma@uv.es>

**Lingüística computacional**

Xavier Gómez Guinovart (Univ. de Vigo), <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante), <mpalomar@dsi.ua.es>

**Mundo estudiantil y jóvenes profesionales**

Federico G. Mon Trotti (RITSI), <gnu.fede@gmail.com>

Mikel Salazar Peña (Asoc. Jóvenes Profesionales, Junta de ATI Madrid), <mikelbo\_uni@yahoo.es>

**Profesión Informática**

Rafael Fernández Calvo (ATI), <rfoalvo@ati.es>

Miguel Santes Gurió (ATI), <msantes@ati.es>

**Redes y servicios telemáticos**

José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>

Juan Carlos López López (UCLM), <juancarlo@uclm.es>

**Robótica**

José Cortés Arenas (Sopra Group), <jccortese@gmail.com>

Juan González Gómez (Universidad Carlos III), <jian@learobotics.com>

**Seguridad**

Javier Arellito Bertolin (Univ. de Deusto), <jarellito@eside.deusto.es>

Javier López Muñoz (ETS Informática-UMA), <jlm@lcc.uma.es>

**Sistemas de Tiempo Real**

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <aalonso.jpunte@dit.upm.es>

**Software Libre**

Jesús M. González Barahona (Universidad Politécnica de Madrid), <israel.herraz@upm.es>

Israel Herraz Tabernero (UAJ), <isra@herraz.org>

**Tecnología de Objetos**

Jesús García Molina (DIS-UM), <jmolina@um.es>

Gustavo Rossi (LPIA-UNLP, Argentina), <gustavo@sol.inf.unlp.edu.ar>

**Tecnologías para la Educación**

Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>

César Pablo Córcoles Briogio (UOC), <ccorcoles@uoc.edu>

**Tecnologías y Empresa**

Didac López Vilas (Universitat de Girona), <didac.lopez@ati.es>

Francisco Javier Gantús Sánchez (Indra Sistemas), <jfgantus@gmail.com>

**Tendencias tecnológicas**

Alonso Álvarez García (TID), <aad@tid.es>

Gabriel Marín Fuentes (Interbits), <gabi@atinet.es>

**TID y Turismo**

Andrés Aguiayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga), <aguiayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

**Coordinación Editorial, Redacción Central y Redacción ATI Madrid**

Padilla 66, 3º dcha., 28006 Madrid

Tfn. 91 4029391; fax. 91 3093685 <novatica@ati.es>

**Composición, Edición y Redacción ATI Valencia**

Av. del Reino de Valencia 23, 46005 Valencia

Tfn./fax 963330392 <secreval@ati.es>

**Administración y Redacción ATI Cataluña**

Via Llaletana 46, ppal. 1º, 08003 Barcelona

Tfn. 93 41 25 235; fax. 93 41 27 713 <secregen@ati.es>

**Redacción ATI Aragón**

Lagasca 9, 3-B, 50006 Zaragoza.

Tfn. fax. 97 62 35 181 <seccara@ati.es>

**Redacción ATI Andalucía**

<secreand@ati.es>

**Redacción ATI Galicia**

<secregal@ati.es>

**Suscripción y Ventas**

<<http://www.ati.es/novatica/interes.html>>, ATI Cataluña, ATI Madrid

**Publicidad**

Padilla 66, 3º dcha., 28006 Madrid

Tfn. 91 4029391; fax. 91 3093685 <novatica@ati.es>

Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona.

Depósito legal: B 15.154-1975 - ISSN: 0211-2124, CODEN NOVAEC

Portada: Luces técnicas - Concha Aras Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

**editorial**

**La fuerza de trabajo en Informática**

> 02

**en resumen**

**Construcción sistemática de edificios del conocimiento**

> 02

*Llorenç Pagés Casas*

**Noticias de IFIP**

**Reunión del TC 13 (Human-Computer Interaction)**

> 03

*Julio Abascal González*

**Reunión del Consejo de IFIP**

> 06

*Ramon Puigjaner Trepal*

**Coordinación editorial**

> 07

**monografía**

**Ingeniería de Sistemas de Aprendizaje Electrónico**

(En colaboración con **UPGRADE**)

*Editores invitados: José Luis Sierra Rodríguez, Antonio Sarasa Cabezuolo, Demetrios G.*

*Sampson, Kinshuk, Ignacio Aedo Cuevas*

**Presentación. Diseño y construcción de Sistemas de Aprendizaje Electrónico**

> 08

*Ignacio Aedo Cuevas, Kinshuk, Demetrios G. Sampson, Antonio Sarasa Cabezuolo,*

*José Luis Sierra Rodríguez*

**Ingeniería de Sistemas de Aprendizaje Electrónico y esfuerzos de estandarización:**

**Un caso de estudio relativo a los Almacenes de Objetos Didácticos**

> 10

*Antonio Sarasa Cabezuolo, Daniel Rodríguez Cerezo, José Luis Sierra Rodríguez*

**Sistemas PDP de contenidos educativos audiovisuales:**

**Desarrollo de una solución para la Universidad de La Laguna**

> 13

*Carina S. González González, David Cabrera Primo, Daniel López Barrios, Antonio Barroso Díaz*

**Patrones de comunicación en sistemas tutores inteligentes basados**

**en componentes**

> 17

*Géraldine Ruddeck, Dennis Maciuszek, Martina Weicht, Alke Martens*

**Sistemas Tutores Inteligentes con SCORM**

> 23

*Gustavo Soares Santos, Álvaro Reis Figueira*

**Servicios eLearning en Moodle 2.0**

> 29

*Miguel A. Conde González, Alberto del Pozo de Dios, Francisco J. García Peñalvo*

**Un caso práctico de integración de agentes y servicios en entornos de**

**eLearning mediante espacios de tuplas**

> 34

*Francisco Jurado Monroy, Ana I. Molina Díaz, Miguel A. Redondo Duque, Manuel Ortega Cantero*

**Hacia la invocación de servicios de e-Learning adaptativo: Aplicación a un**

**simulador de navegación y pesca**

> 40

*Valérie Monfort, Maha Khemaja*

**Una metodología de diseño para el aprendizaje ubicuo**

> 46

*Jihen Malek, Mona Laroussi, Alain Derycke, Henda Ben Ghezala*

**secciones técnicas**

**Estándares web**

**Una aproximación basada en modelos para la ejecución automática**

**de modelos de procesos de negocio**

> 51

*Valeria de Castro, Javier Fabra Caro, Pedro Álvarez Pérez-Arados, Esperanza Marcos Martínez*

**Redes y servicios telemáticos**

**Diseño e Implementación de Video Streaming en Redes de Sensores Inalámbricas**

> 57

*David Rodenas Herráiz, Antonio Javier García Sánchez, Felipe García Sánchez*

**Robótica**

**Arquitecturas de control de robots**

> 62

*José Cortés Arena*

**Referencias autorizadas**

> 65

**sociedad de la información**

**Programar es crear**

**Mi número de Erdos (solución)**

> 72

*Julio Javier Castillo, Diego Javier Serrano*

**La Forja**

**Creación de un Clúster de Alta Disponibilidad con software libre (solución)**

> 74

*Miguel Vidal López, José Castro Luis*

**asuntos interiores**

**Socios Institucionales**

> 77

**Monografía del próximo número: "Business Intelligence"**

David Rodenas Herráiz, Antonio Javier García Sánchez, Felipe García Sánchez

Depto. de Tecnologías de la Información y Comunicaciones, Área de Ingeniería Telemática, Universidad Politécnica de Cartagena (Murcia)

drh@alu.upct.es,  
antoniojavier.garcia@upct.es,  
felipe.garcia@upct.es

## 1. Introducción

Una Red de Sensores Inalámbrica (*Wireless Sensor Networks*, WSN) [1] es un conjunto de dispositivos de bajo coste, reducido consumo y limitada capacidad de procesamiento cuya funcionalidad es la captura, transmisión y recepción de información procedente de sensores tales como temperatura, presión, humedad, etc. El objetivo es que un usuario final reciba la información de monitorización de uno o varios de estos parámetros de interés, donde las medidas se realizan en amplios intervalos de tiempo. Asimismo, y debido a la heterogeneidad de las variables de estudio, el empleo de las WSN abarca multitud de campos, tales como industria, domótica, agricultura, el campo militar, etc.

Adicionalmente, la disponibilidad de *hardware* multimedia de bajo coste, como micrófonos y cámaras CMOS, abre un nuevo campo de investigación dentro de este tipo de redes, dedicado a explotar la posibilidad de integración de sensores de sonido, vídeo o ambos (*Multimedia Streaming Application*, MSA) con las actuales aplicaciones WSN. El resultado son las denominadas Redes de Sensores Inalámbricas Multimedia (*Wireless Multimedia Sensor Networks*, WMSN) donde un usuario además de monitorizar los tradicionales parámetros de una red WSN puede visualizar y escuchar el emplazamiento físico de su aplicación.

Para llevar a cabo transmisiones eficientes de vídeo, el diseño de una WMSN debe garantizar un máximo de ancho de banda y un mínimo retardo. Estos requisitos se enfrentan a las actuales restricciones *hardware* de los dispositivos WSN. Un nodo WSN posee una capacidad de cómputo, comunicación y almacenamiento limitadas, y no es capaz de gestionar eficientemente la captura, procesamiento y transmisión de información multimedia.

La salida al mercado de la plataforma comercial *Imote2* [2] pretendía solucionar este problema. *Imote2* es un dispositivo de desarrollo dentro de las WMSN pensado para la gestión de contenidos multimedia. Esta plataforma es desarrollada y distribuida por la compañía *MEMSIC Inc.* en colaboración con *Intel Corporation* y la *Universidad de Berkeley*. Sus prestaciones de memoria y procesamiento son muy superiores a las ofrecidas por

# Diseño e Implementación de Vídeo Streaming en Redes de Sensores Inalámbricas

**Resumen:** Los dispositivos comerciales tradicionales que forman una red de sensores inalámbricos (WSN) están diseñados para monitorizar parámetros principalmente físicos (temperatura, humedad, presión, etc.), donde cada muestra es tomada en largos intervalos de tiempo. En contraposición, aplicaciones como las basadas en vídeo imponen una serie de requisitos hardware que hacen inviable el desarrollo de software streaming en los actuales dispositivos WSN. Sin embargo, la aparición de la plataforma *Imote2* resuelve teóricamente todas estas limitaciones abriendo nuevos campos de aplicación dentro de las WSN. En la práctica, el desarrollo de un programa de vídeo totalmente funcional y compatible con este dispositivo no ha sido todavía implementado por la comunidad científica internacional. Este artículo presenta un software consistente en la captura, procesamiento y transmisión de vídeo codificado vía radio totalmente compatible con los dispositivos *Imote2*. Además, se presenta un código para PC cuyo cometido es la recepción, decodificación y visualización del vídeo transmitido por el dispositivo *Imote2* al usuario. Finalmente, los resultados de la evaluación del funcionamiento experimental son mostrados y discutidos.

**Palabras clave:** *Imote2*, transmisión de vídeo, WSN.

otros dispositivos WSN tales como *Tmote Sky*, *MicaZ* o *TelosB*. Asimismo, la placa *IMB400* diseñada para *Imote2* es un dispositivo que facilita al implementador funcionalidades multimedia como captura de imágenes, captura y reproducción de sonido o la detección de movimiento.

Otra de las ventajas de usar *Imote2* frente a otras plataformas comerciales es que permite desarrollar aplicaciones bajo el sistema operativo *TinyOS* [3], diseñado por la *Universidad de Berkeley*. *TinyOS* es usado por un amplio número de grupos de investigación y comunidades de desarrollo por los siguientes motivos:

- *TinyOS* es un proyecto de código abierto (*Open Source*).
- El diseño de *TinyOS* se basa en una estructura de dos niveles de planificación: (1) eventos y (2) tareas. Este enfoque es idóneo para alcanzar un alto rendimiento de las aplicaciones diseñadas, usando las capacidades de CPU, memoria y energía de forma eficiente.
- *TinyOS* se ha programado en lenguaje *nesC*, un lenguaje diseñado para reflejar las ideas propias de un modelo de componentes. Además, incorpora un modelo de programación que soporta concurrencia y manejo de comunicaciones, permitiendo desarrollar aplicaciones en dispositivos limitados en recursos como los de las WSN.

Sin embargo, debido a lo novedoso de la plataforma *Imote2*, *TinyOS* presenta dos importantes desventajas: (1) el número de módulos implementados compatibles es mucho menor que en otros dispositivos *hardware* WSN; (2) la documentación dispo-

nible es escasa, no existiendo apenas información de aquellos módulos que ya han sido implementados. Estos inconvenientes han motivado que el número de implementaciones que usan la plataforma *Imote2* sea menor a la de otras plataformas comerciales.

En este artículo presentamos el diseño e implementación de nuevos módulos y componentes *nesC* capaces de desarrollar las características de captura, procesamiento, transmisión y recepción de vídeo en una WMSN. Ello ha dado lugar a un software pionero en este campo de aplicación que, a diferencia de otras aplicaciones, es capaz de resolver de una manera eficiente la transmisión y presentación al usuario de vídeo codificado dentro de una WSN.

El resto del artículo está organizado como sigue: la **sección 2** resume los trabajos relacionados encontrados en la literatura especializada; la **sección 3** introduce a la plataforma de trabajo; la **sección 4** describe la aplicación de *streaming* de vídeo *h263\_camera*, desarrollada en *TinyOS 2.x* para *Imote2*; la **sección 5** describe la aplicación *H263Viewer*, desarrollada en *Visual C++* para PC; la **sección 6** presenta los resultados de la evaluación del funcionamiento experimental, mostrando las imágenes obtenidas y un estudio sobre la latencia temporal, consumo de energía y tiempo de vida de las baterías del dispositivo *Imote2*; la **sección 7** concluye este artículo.

## 2. Trabajos relacionados

En los últimos años, ha surgido un gran interés en la comunicación multimedia con

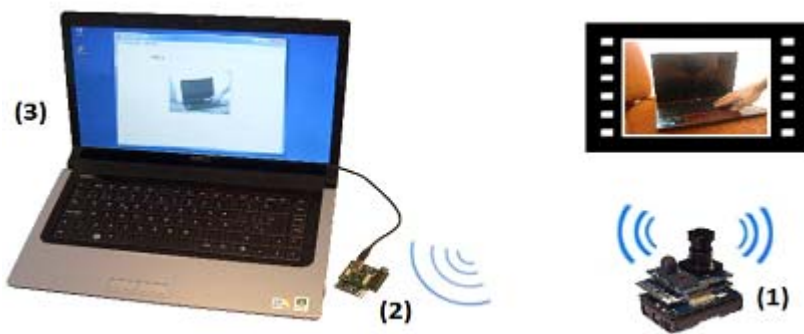


Figura 1. Esquema de red: (1) dispositivo *Imote2* emisor; (2) dispositivo *Imote2* estación base; (3) PC (*software H263Viewer*).

redes de sensores inalámbricas. Por esta razón, algunos trabajos previos recopilan el estado del conocimiento de las WMSN [4][5][6]. Estos estudios realizan una clasificación por capas de comunicaciones (enlace, red, transporte, etc.), presentando y discutiendo los requisitos de cada una de ellas y proponiendo líneas de trabajo futuro.

Es destacable que estos trabajos coinciden en que el paradigma de una codificación eficiente para estos sistemas es un problema aún no resuelto, debido a los requisitos de baja capacidad de procesamiento y almacenamiento de los dispositivos WSN. De esta forma, la elección del codificador se convierte en un factor de diseño crítico.

Los esquemas ISO MPEG o las equivalentes recomendaciones ITU-T H.263 y H.264 son las técnicas que tradicionalmente han dominado la codificación de vídeo en comunicaciones inalámbricas. Se basan en la idea de reducir la tasa de bits generada por la fuente codificadora explotando la redundancia entre las imágenes. Además, existen otros trabajos dedicados exclusivamente a diseños de nuevas técnicas de codificación.

En [7] los autores realizan un estudio teórico de un *codec* basado en el algoritmo *Winer-Ziv*

cuyo objetivo es reducir la complejidad del codificador. Emplean un bloque *Huffman* para conseguir un mayor ratio de compresión y programan dispositivos *Imote2* para medir el consumo de energía. Los resultados teóricos son eficientes, pero aunque existen implementaciones de este tipo de codificadores, aún se encuentra en fase de prueba. En [8] se presenta un nuevo codificador de vídeo sin pérdidas que trata la redundancia temporal entre imágenes. Tal y como los autores exponen en sus resultados, su ratio de compresión es menor que el de los codificadores tradicionales y sólo son apropiados para entornos estáticos.

En la literatura especializada, existen otras investigaciones dedicadas a solucionar otros requisitos diferentes a los de codificación. En Jung *et al.* [9] los autores desarrollan una herramienta de simulación con objeto de optimizar el consumo de energía en la transmisión de vídeo. Este simulador es validado mediante el desarrollo de una aplicación sobre el dispositivo *Imote2*, obviando el uso de cualquier tipo de *codec*.

En [10] se presenta un *software* desarrollado por la *Universidad de Stanford* para *Imote2* cuyo objetivo es la transmisión de imágenes JPEG. Sin embargo, paradójicamente, no

dispone de la implementación de un codificador JPEG eficiente, por lo que la aplicación sólo es capaz de enviar imágenes sin codificar.

En este trabajo se presenta la integración de distintos conceptos y tecnologías existentes (*codec* de vídeo H.263, almacenamiento, gestión de memoria, transmisión de vídeo, etc.) en el dispositivo *Imote2*, las cuales proporcionan una aplicación de envío/recepción de flujos de vídeo robusta y eficiente.

### 3. Descripción de la plataforma de trabajo

La plataforma de trabajo desarrollada consiste en un prototipo emisor-receptor de vídeo para WSN compuesto de una aplicación transmisora de vídeo codificado y una estación cliente que decodifica y presenta la secuencia de imágenes al usuario.

La estación transmisora está formada por un *Imote2* y una placa sensora *IMB400*. Hemos desarrollado un *software* de *streaming* de vídeo denominado *h263\_camera* que bajo la plataforma *Imote2* y el sistema operativo *TinyOS 2.x* permite capturar imágenes en tiempo real, codificarlas y enviarlas vía radio a otro dispositivo *Imote2* que actúa de estación base cliente.

La estación receptora implementa dos módulos *software*: *BaseStation* de *TinyOS 2.x* y *H263Viewer*. En primer lugar, los *frames* enviados por la estación transmisora son gestionados por la aplicación *BaseStation* de la estación base cliente (compuesta por un *Imote2*). En segundo lugar, los *frames* son transmitidos a un PC vía serie, el cual implementa el *software H263Viewer*. Este *software* implementa las funciones de decodificación y visualización al usuario de las imágenes.

La figura 1 muestra el esquema de la red *one-hop* formado por los elementos anteriores.

Además, la aplicación *H263Viewer* permite al usuario controlar el inicio/fin de la transmisión de vídeo en el sensor.

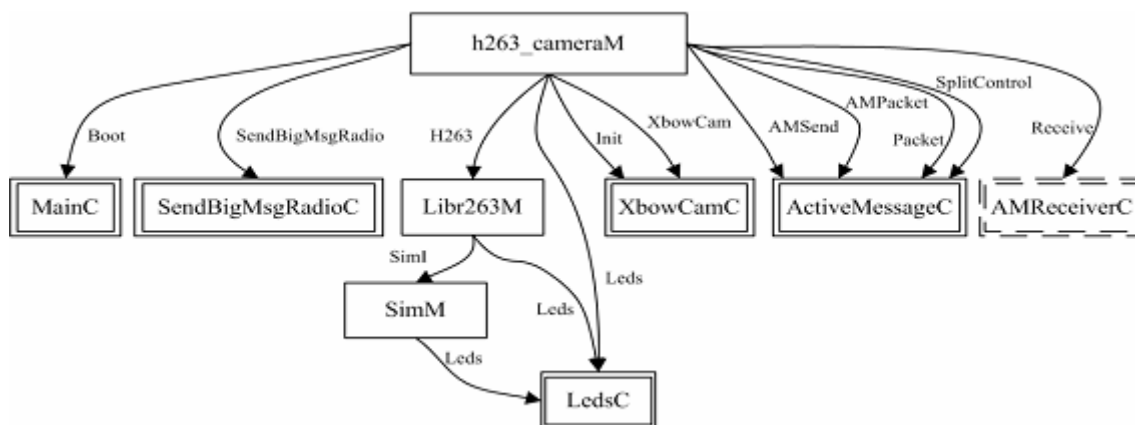


Figura 2. Diagrama de componentes de la aplicación *h263\_camera*.

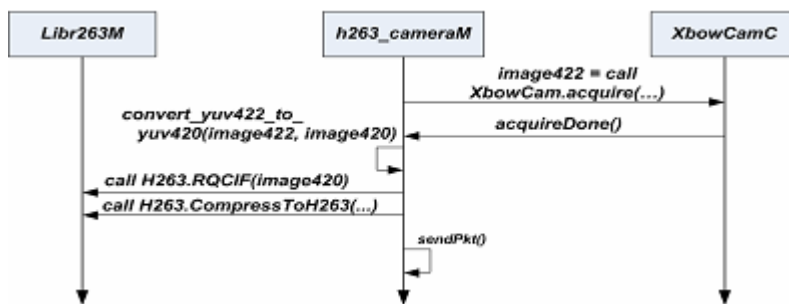


Figura 3. Secuencia de acciones de la aplicación h263\_camera.

#### 4. Diseño y detalles de implementación de la aplicación de transmisión de vídeo h263\_camera

La aplicación h263\_camera implementa los componentes software necesarios para realizar por una parte la captura de imágenes desde la placa IMB400 y por otra, la codificación H.263, versión en nesC de la implementación software de libre distribución Telenor TMN 1.6 (Test Model Near-Term) [11]. En relación al codificador, fiel a la Recomendación ITU-T H.263 [12], posee como principal característica la disminución de las exigencias computacionales en sistemas de tiempo real.

Este codificador permite incorporar las dos técnicas de compresión existentes: (1) intra-frame, que explota la redundancia espacial que existe en una imagen y (2) inter-frame, que aprovecha la correlación temporal existente entre imágenes consecutivas. En nuestro prototipo, sólo se ha considerado la técnica de compresión intra-frame (con pérdidas) dejando la técnica inter-frame para posteriores desarrollos. No es objetivo de este artículo profundizar en el proceso de codificación empleado.

Los recursos de memoria disponibles son un factor a tener en cuenta ya que el dispositivo Imote2 debe almacenar y procesar gran cantidad de información en tiempo real. Por tanto, es necesario implementar una gestión eficaz del almacenamiento en memoria, optimizando así los recursos existentes.

Finalmente, el envío de frames vía radio es una tarea también desarrollada por el software h263\_camera. En TinyOS 2.x, y a través del transceptor Chipcon 2420 que integra Imote2, sólo es posible enviar una cantidad de información limitada dentro de un frame de datos. Sin embargo, el tamaño del bloque de datos (imagen codificada) a enviar por la aplicación h263\_camera es superior a dicho límite. Este hecho conlleva la necesidad de fragmentar la imagen codificada.

La figura 2 muestra los módulos y componentes software que componen la aplicación h263\_camera y que son explicados brevemente a continuación:

- El módulo h263\_cameraM representa la abstracción de la aplicación h263\_camera. Gestiona la adquisición de imágenes, procesado y codificación, y envío radio.
- El componente MainC implementa la secuencia de inicio de un nodo.
- Los componentes ActiveMessageC y AMReceiverC permiten la transmisión y recepción de flujos de datos en TinyOS 2.x por el interfaz radio.
- El componente SendBigMsgRadioC proporciona servicios de tipo Active Message para el envío de bloques de datos en los que es necesario fragmentar.
- El componente XbowCamC permite adquirir imágenes con la placa IMB400.
- El componente Libr263M contiene la librería para la codificación H.263. Implementa el componente SimM, que contiene las fun-

ciones necesarias para realizar el proceso de codificación a través de la interfaz SimI.

- El componente LedsC tiene como objetivo controlar los LEDs del sensor. La secuencia de funcionamiento de la aplicación se muestra en la figura 3:

- La aplicación h263\_camera llama al comando acquire del componente XbowCamC para adquirir una imagen.

- El método acquire lanza el evento asíncrono acquireDone. Se obtiene una estructura de datos que constituye una imagen 'en bruto' de tipo QCIF (176 x 144 píxeles) con muestreo YUV 4:2:2.

- Sin embargo, el muestreo de entrada del codificador H.263 es YUV 4:2:0. Para obtenerlo la aplicación h263\_camera llama al método convert\_yuv422\_to\_yuv420.

- La aplicación h263\_camera llama al comando RQCIF del componente Libr263M para obtener la estructura de datos de entrada del codificador.

- La aplicación h263\_camera inicia el proceso de codificación con el comando CompressToH263 del componente Libr263M.

- La aplicación h263\_camera llama al método sendPkt para enviar la imagen codificada vía radio.

Este proceso se repite hasta que sea finalizado por el usuario o el sensor que lo activó.

#### 4.1. Captura de imágenes

La placa IMB400 permite capturar imágenes para su posterior codificación y envío radio. Esta función se consigue gracias al chip OV7670 de Omnivision que integra la propia placa. Consiste en un sensor CMOS de bajo voltaje que proporciona la funcionalidad completa de una cámara VGA y un procesador de imágenes. La aplicación h263\_camera opera con la máxima capacidad disponible que permite el sensor OV7670, 30 fotogramas por segundo (fps).

El componente XbowCamC (figura 4a) permite usar y controlar la placa IMB400. Por su parte, el módulo HplOV7670M gestiona todas las funcionalidades disponibles del sensor OV7670, como la calidad de la imagen

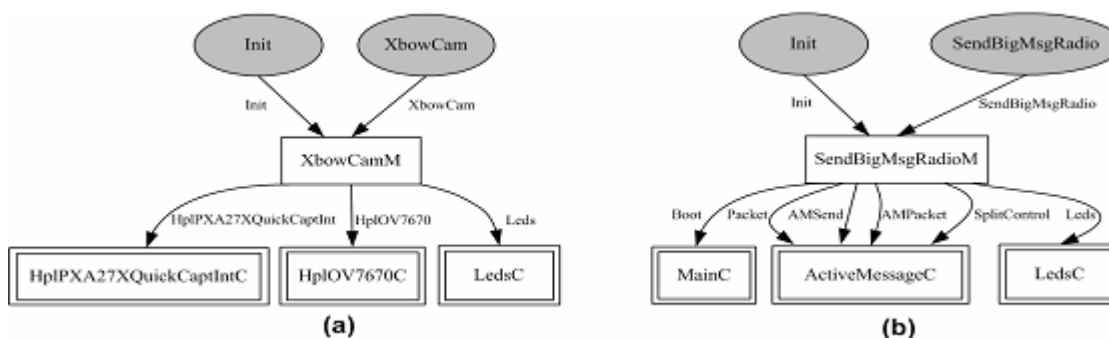


Figura 4. Diagrama de componentes de (a) XbowCamC y (b) SendBigMsgRadioC.



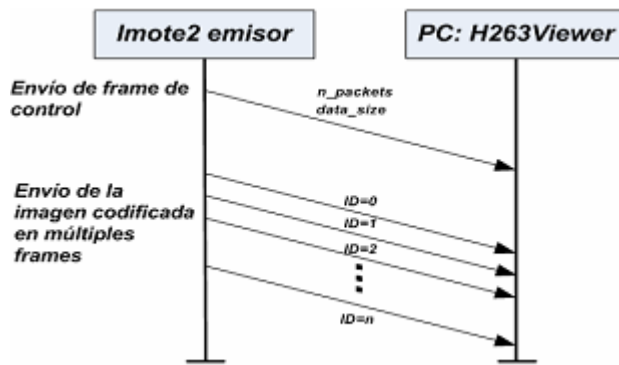


Figura 5. Diagrama UML. Proceso de envío radio.

y formato de salida, entre otros. Estos dos controladores se encuentran en una fase temprana de desarrollo, donde la información disponible es escasa, por lo que muchas características no han sido implementadas hasta ahora. Entre ellas, el escalado de imagen que nos interesa obtener, QCIF (176 x 144 píxeles). Los comandos y funciones actualmente desarrollados sólo permitían obtener imágenes de tipo VGA (640 x 480 píxeles) y QVGA (320 x 240 píxeles), por lo que hemos implementado la posibilidad de obtener imágenes de tipo QCIF.

#### 4.2. Gestión de memoria

Uno de los problemas a la hora de desarrollar aplicaciones de vídeo para WSN es la gestión de la memoria. Los dispositivos existentes en el mercado se encuentran limitados en cuanto a capacidad de memoria física disponible y por ello es importante realizar una correcta gestión de los recursos disponibles. Dentro de las plataformas *hardware* comerciales, *Imote2* presenta las mejores prestaciones en cuanto a memoria y procesado. Dispone de 3 memorias: 256 KB de SRAM (*Static Random Access Memory*), 32 MB de SDRAM (*Synchronous Dynamic Random Access Memory*) y 32 MB de *Flash*. En particular, las memorias SRAM y SDRAM son volátiles mientras que *Flash* almacena la información de forma permanente.

Las variables y estructuras de datos declaradas en *nesC* se almacenan por defecto en la memoria SRAM. Sin embargo, la aplicación *h263\_camera* trabaja con estructuras de datos de gran tamaño, asociadas al proceso de codificación y almacenamiento de la imagen. Por ello, los requerimientos de memoria superan a la disponible en SRAM. Se hace necesario el uso de una memoria de mayor capacidad. La memoria SDRAM cumple con los requisitos anteriores: su capacidad es suficiente y no es necesario almacenar la información de forma permanente (*Flash*).

#### 4.3. Transmisión mediante la conexión radio

El tamaño de datos de *payload* en *TinyOS 2.x*

para el transceptor radio *Chipcon 2420* de *Imote2* está definido por la constante *TOSH\_DATA\_LENGTH*, que por defecto es de 28 bytes, llegando a ser como máximo de 114 bytes. La salida del codificador constituye un bloque de datos de tamaño variable dependiendo de la imagen comprimida (máximo 3000 bytes). Este tamaño es superior al máximo valor de la constante *TOSH\_DATA\_LENGTH*. Por tanto, es necesario fragmentar. Para ello, se ha desarrollado el componente *SendBigMsgRadioC* (figura 4b). Este componente permite enviar vía radio bloques de información que necesiten ser fragmentados, y que en nuestro caso será en *frames* con el máximo *payload* posible, es decir, 114 bytes.

El bloque de datos codificado se envía en múltiples *frames*. Por tanto, es importante realizar una correcta gestión de la transmisión radio. Así, el envío radio se divide en dos fases consecutivas.

La figura 5 muestra un diagrama de flujo representativo de este proceso.

■ **Envío de un frame de control.** La aplicación *h263\_camera* envía un *frame* de datos a la estación receptora con los siguientes datos: (1) número de paquetes (fragmentos) que se van a enviar y (2) tamaño de la salida del codificador. Para ello, llama al comando *send* del componente *ActiveMessageC*. El envío de este *frame* permite a la aplicación *H263Viewer* controlar la posterior recepción, decodificación y visualización de la imagen sin perder la secuencia de vídeo.

■ **Envío de la imagen codificada en múltiples frames.** La aplicación *h263\_camera* llama al comando *send* del componente *SendBigMsgRadioC*. Este comando envía el bloque codificado en sucesivos *frames* hacia su destino. Para ello, cada *frame* contiene en el campo *payload* la porción de datos (máximo 108 bytes) correspondiente a la imagen codificada, junto con una cabecera (6 bytes) donde se incluye número de secuencia, ID del nodo emisor y tamaño de datos útiles.

## 5. Diseño y detalles de implementación de H263Viewer

En esta sección se describe la aplicación *H263Viewer*, desarrollada en *Visual C++* para la recepción y visualización de imágenes captadas por el sensor *Imote2* emisor. *H263Viewer* implementa un decodificador H.263 y está conectada vía serie a un dispositivo *Imote2* que actúa de estación base. No es objetivo de este artículo detallar las características del proceso de decodificación.

### 5.1. Comunicación entre PC y nodo estación base

El nodo estación base implementa la aplicación *BaseStation* de *TinyOS 2.x*. Su cometido es recibir información procedente del nodo emisor por su interfaz radio, y enviarla al PC por su interfaz serie, y viceversa.

*H263Viewer*, necesita abrir el puerto serie para poder recibir los datos de la estación base. Para ello, *H263Viewer* se conecta a la aplicación *SerialForwarder* de *TinyOS 2.x*, un servidor TCP en el puerto 9001 que permite la comunicación PC-*Imote2*-PC, intercambiándose datos mediante un protocolo del mismo nombre, el protocolo *Serial Forwarder*.

### 5.2. Recepción y visualización de la imagen

Los *frames* transmitidos por el nodo emisor se reciben de forma ordenada. El proceso de recepción ejecuta la acción adecuada según el tipo de *frame* recibido:

- *H263Viewer* recibe el *frame* de control. Se extraen los datos referentes a la posterior recepción de la imagen, es decir, el número de fragmentos que la componen y el tamaño de la imagen codificada que va a recibir.
- *H263Viewer* recibe un fragmento de datos. Este paso se repite hasta recibir el último fragmento. En este caso la secuencia de acciones es la siguiente:
  - Extracción del número de secuencia del *frame*. Si no es correcto, finaliza este paso, y la aplicación se queda a la espera del siguiente *frame* de control.
  - Obtención del tamaño de datos útiles contenidos en el *frame*.
  - Extracción de la cantidad de datos correspondiente y copia en un *buffer* (*RecvBuf*) a continuación de los extraídos antes. Así se reconstruye el bloque de datos que constituye la imagen completa.

Una vez recibidos todos los fragmentos y reconstruida la imagen codificada, *H263Viewer* decodifica la imagen en formato H.263 y la visualiza al usuario.

## 6. Evaluación funcional

Se ha desarrollado un test de pruebas con objeto de validar la herramienta *software* de transmisión/recepción de vídeo sobre redes WSN diseñada en este artículo. Para ello se ha realizado un estudio del retardo temporal

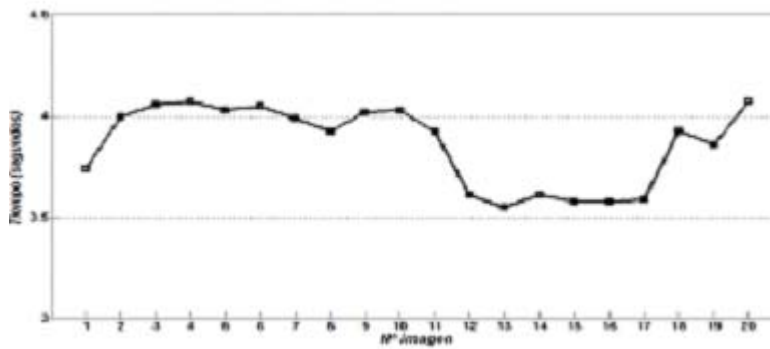


Figura 6. Retardo de imágenes.

asociado al proceso de captura-procesado-transmisión-recepción-visualización de ellas, del consumo de energía y tiempo de vida de las baterías del dispositivo *Imote2*. El escenario de pruebas es el mostrado en la figura 1.

Estas son las características del entorno donde se ha realizado el estudio:

- Entorno *indoor*.
- Visión directa entre dispositivos *Imote2* emisor y estación base, separados 20-30 m.
- Tamaño máximo de *payload* en los paquetes enviados por el nodo emisor (114 bytes).
- La máxima frecuencia de procesador y de bus de los sensores *Imote2* es 416 MHz. Sin embargo, *TinyOS 2.x* sólo permite trabajar hasta 104 MHz.

En la actualidad, las imágenes obtenidas con la placa *IMB400* y el sensor *OV7670* son en B/W. Los controladores de la placa *IMB400* (componente *XbowCamC*) no tienen implementada la funcionalidad de color en la captura de imágenes. A pesar de ello y de que se realiza un proceso de compresión con pérdidas, la visualización por parte del usuario es totalmente correcta.

El estudio temporal se ha realizado sobre un conjunto de 20 imágenes (figura 6). Se basa en medir el retardo total existente desde que se captura una imagen por el sensor *OV7670* hasta que ésta se visualiza en la aplicación *H263Viewer*. Así, los retardos temporales de interés son los siguientes: (1) adquisición de una imagen; (2) cambio de formatos YUV; (3) codificación; y (4) transmisión y visualización de la imagen de vídeo al usuario. El tiempo de decodificación y visualización puede considerarse despreciable.

Para una visualización adecuada por parte del usuario final, se observa que el retardo temporal se mantiene constante entre 3,5 y 4 segundos. Este tiempo se puede considerar aceptable, teniendo en cuenta la cantidad de cómputo que debe realizar un dispositivo de características muy reducidas.

Finalmente, sobre el consumo energético de estos dispositivos hay que tener en cuenta que la transmisión de vídeo en WSN se realiza durante un período acotado de tiempo y sólo cuando un usuario u otro sensor la activan.

Por ello, para cuantificar el consumo de energía, hemos desarrollado una aplicación de tele-vigilancia, desplegando dispositivos *Imote2* con sensores PIR (*Passive Infrared Sensor*) y de vídeo en cultivos ecológicos. Los sensores PIR detectan la posible entrada de intrusos y activan los sensores de vídeo, siendo el dispositivo *Imote2* quien envía al usuario final la alarma junto al flujo de vídeo. A partir de este estudio, el número medio de eventos diarios fue de 20, cada uno de los cuales activa la transmisión de vídeo durante 30 segundos. Según la especificación de consumos energéticos del *Imote2* [2], el consumo medio de éste particularizado a nuestro escenario es de 4,2 mA. Si el dispositivo *Imote2* dispone de 3 baterías tipo AAA de 1.200 mA/h (*Energizer-LR03*), el tiempo de vida del sistema es de unos 40 días.

## 7. Conclusiones y trabajo futuro

Hemos presentado el diseño de un prototipo que implementa una aplicación de *streaming* de vídeo, pionera dentro de las WSN. Este prototipo permite la transmisión de vídeo codificado entre un dispositivo WSN y un PC. Por tanto, a las ya características funciones de una WSN focalizada en exclusiva a la monitorización de datos de diferentes sensores espaciados en generosos intervalos temporales, se añade la posibilidad de que los usuarios monitoricen visualmente su aplicación.

Dada la originalidad de este trabajo, existen una serie de acciones planificadas para mejorar el funcionamiento de la aplicación. Entre ellas se encuentran:

- Expandir la funcionalidad *inter-frame*, explotando por tanto la redundancia temporal, para el algoritmo de codificación H.263 implementado en la aplicación *h263\_camera*. El propósito es reducir el retardo entre imágenes.

- Desarrollo de *drivers* explícitos y completos para la placa *IMB400*. El objetivo es permitir obtener imágenes a color con la máxima calidad.

- Extensión a una red *multi-hop*. Diseñar algoritmos *ad-hoc* que permitan distribuir labores de encaminamiento entre los diferentes nodos que forma la red con el objeto de disminuir los consumos de energía.

## Agradecimientos

Esta investigación ha sido financiada por el Ministerio de Ciencia e Innovación, dentro del proyecto TEC2010-21405-C02-02/TCM (CALM). Además, este trabajo se ha desarrollado dentro del marco de "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM".

## Referencias

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. "Wireless sensor networks: a survey", *Computer Networks*, vol. 38, pp. 393-422, 2002.
- [2] MEMSIC Inc. *Imote2 hardware bundle for Wireless Sensor Networks; Datasheet Imote2*.
- [3] P. A. Levis, D. Gay. *TinyOS Programming*. Cambridge University Press, 2009.
- [4] I. F. Akyildiz, T. Melodia, K. R. Chowdhury. A Survey on Wireless Multimedia Sensor Networks, *Computer Networks*, vol. 51, pp. 921-960, 2007.
- [5] S. Misra, M. Reisslein, X. Guoliang. A survey of multimedia streaming in wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 4, pp. 18-39, 2008.
- [6] I. T. Almkaw, M. Guerrero Zapata, J. N. Al-Karaki, J. Morillo-Pozo. Wireless Multimedia Sensor Networks: Current Trends and Future Directions. *Sensors*, vol. 10, no. 7, pp. 6662-6717, 2010.
- [7] H. Abid, S. Qaisar. *Distributed video coding for wireless visual sensor networks using low power Huffman coding*, 2010, pp. 1-6.
- [8] S. Pudlewski, T. Melodia. *On the Performance of Compressive Video Streaming for Wireless Multimedia Sensor Networks*, 2010, pp. 1-5.
- [9] J. Deokwoo, T. Thiago, S. Andreas. Sensor node lifetime analysis: Models and tools. *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 1-33, 2009.
- [10] E. Alavarez, Y. Jimenez, M. Pahn C. *Interconnexion d'une plateforme de réseaux de capteurs vidéo à l'Internet avec visualisation sur un navigateur Web*, 2010.
- [11] K. O. Lillevold et al. *H263 test model simulation software*, Telenor R&D, 1995.
- [12] ITU-T. *Recommendation H.263 - Video coding for low bit rate communication*.