

Plataforma para monitorización y control distribuidos basada en 802.15.4

Conrado Arquer, Luis Orihuela, Francisco R. Rubio
Dpto. Ingeniería de Sistemas y Automática, Universidad de Sevilla, España
conarqcob@alum.us.es, orihuela@us.es, rubio@us.es

Resumen

Este artículo presenta una plataforma basada en redes de sensores inalámbricas que se caracteriza por su modularidad y bajo consumo. Para mostrar su utilidad en sistemas de control, se han construido dos aplicaciones diferentes: monitorización remota de sistemas; y control distribuido de flotas de vehículos autónomos. La solución tecnológica utiliza dispositivos Waspote de Libelium que se comunican a través del protocolo 802.15.4 (ZigBee).

Palabras Clave: Redes de sensores inalámbricas, Monitorización de sistemas, Control distribuido, Flotas de vehículos autónomos.

1. Introducción

Las redes de sensores inalámbricas, también conocidas por sus siglas en inglés WSN –Wireless Sensor Networks– están alcanzando en los últimos años una gran aceptación en la comunidad científica y en la sociedad en general [1]. El principal motivo para este mayor alcance es el abaratamiento de los sistemas microelectrónicos y de comunicación.

Aunque las tecnologías derivadas de las normas 802.11x son aplicables en WSN, son los protocolos de la familia 802.15.4 (la base de ZigBee) los que han logrado mejor encuadre, ya que implementan accesos al medio basados en contienda que acotan los retrasos de comunicación. Además, están pensados para reducir el consumo de energía, factor crucial en estos sistemas. Estas características los hacen adecuados para su uso en sistemas de control y en tiempo real.

Dentro de la comunidad del control, las WSN se han utilizado principalmente en dos grandes líneas. En primer lugar, cabe destacar su uso para la monitorización de sistemas [2]. Una línea similar es la que se encarga de la estimación distribuida de sistemas de gran tamaño [3] [4].

Por otro lado, es prolífica también la investigación relacionada con el control distribuido o

descentralizado [5] [6]. En ambos casos, el uso de la red se hace necesario porque los diferentes elementos (sensores/actuadores) están localizados en posiciones remotas, y el cableado punto a punto conllevaría grandes costes. Además, algunos sistemas, como las flotas de vehículos autónomos, requieren el uso de este tipo de sistemas por la propia naturaleza de la aplicación [7].

En este trabajo se presenta una plataforma que permite ser utilizada tanto para monitorización de sistemas como para el control distribuido.

Para construir la red de sensores se han empleado las soluciones ofrecidas por Libelium, a través de sus Waspotes [8]. Esta tecnología se caracteriza por su bajo consumo, facilidad de implementación y desarrollo, así como por su variedad de módulos que permiten añadir grandes capacidades sensoriales utilizando productos un mismo fabricante.

Se proponen dos aplicaciones. En primer lugar, se ha construido un sistema de monitorización remoto que permite acceder a las medidas de los diferentes sensores a través de un teléfono móvil. Este tipo de solución es interesante en sistemas que requieran la respuesta rápida de un operador ante diversas situaciones. Por ejemplo, en aplicaciones sanitarias o en determinados procesos críticos de fabricación.

En segundo lugar, dicha red de sensores se ha acoplado a una flota de vehículos, de forma que se dispone de un banco de trabajo para probar diferentes controladores distribuidos para flotas de vehículos.

El artículo está estructurado de la siguiente manera. A continuación se dan las principales características de los Waspotes, así como sus posibilidades de creación de redes de sensores. Las secciones 3 y 4 detallan las aplicaciones de monitorización y control, respectivamente. Finalmente, se termina con algunas conclusiones y posibles trabajos de ampliación.

2. Redes de sensores usando Waspotes

Esta sección presenta en primer lugar una descripción del hardware que se va a emplear. Seguidamente, se

detalla la creación de redes de sensores inalámbricas usando estos dispositivos.

1. Wasp mote

Wasp mote es una plataforma de código abierto para la construcción de redes de sensores inalámbricas que permite varios protocolos de comunicación (ZigBee, Bluetooth, GPRS) y cuyo objetivo principal es la implementación de modos de bajo consumo que permitan a los nodos sensoriales o *motes* comportarse de forma totalmente autónoma.

Los *motes* pueden operar en cuatro modos, como ilustra la Tabla 1. Cabe señalar el consumo mínimo del dispositivo de 0.06 micro amperios mientras se encuentra en el estado “Hibernate”, un consumo mínimo inferior al de la mayoría de dispositivos que realizan funciones similares, véase como ejemplo Mica con un consumo mínimo superior a 1 micro amperio o CM5000 con un consumo mínimo de 1 micro amperio. Gracias a esto, los nodos sensoriales pueden llegar a vivir hasta 5 años de forma ininterrumpida.

Los Wasp motes se alimentan de una pequeña batería de litio que puede recargarse a través de un panel solar, lo que aumenta la autonomía de los dispositivos.

Tabla 1: Modos de operación en los que puede trabajar Wasp mote

	Consumo	Micro	Interrupciones aceptadas
ON	15 mA	ON	Síncronas (Watchdog) y asíncronas
Sleep	55 μ A	ON	Síncronas (RTC) y asíncronas
Deep Sleep	55 μ A	ON	Síncronas y asíncronas
Hibernate	0.06 μ A	OFF	Síncronas

Otra característica a destacar es su arquitectura modular. La placa de Wasp mote (Figura 1) proporciona una serie de *sockets* en los cuales se conectan los diferentes módulos disponibles [8], ampliando la funcionalidad del nodo.

Cada una de estas placas incorpora sensores para la medición de parámetros en un ámbito específico. Por ejemplo la placa “GASES” incorpora sensores para

medir el nivel de oxígeno, dióxido de carbono, monóxido de carbono, etc. De esta forma se podría controlar el nivel de polución en la ciudad, las emisiones de granjas, viveros y las resultantes de procesos industriales, o detectar incendios forestales.

Wasp mote también dispone de dos sensores integrados en la placa base: un sensor de temperatura asociado al RTC (Real-Time Clock) y un acelerómetro de tres ejes (X,Y,Z). Finalmente, cabe señalar que dispone de un bus de comunicaciones I2C.

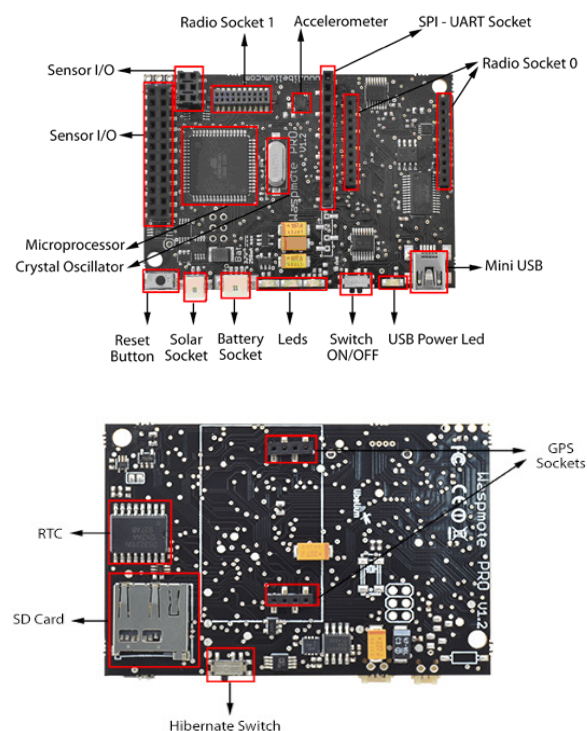


Figura 1: Vista frontal y trasera de la placa de Wasp mote

Para su programación, Libelium proporciona un IDE que incorpora un compilador y una serie de librerías con funciones básicas. Este IDE está basado enteramente en el compilador y las librerías que proporciona Arduino en forma de código abierto.

2. Redes de sensores

Lo que hace más interesantes a los Wasp motes son sus capacidades de comunicación para formar redes de sensores inalámbricos [10]. Para que esta comunicación sea posible el Wasp mote cuenta con un *socket* para la conexión de módulos XBee que se comunican con el microcontrolador a través de la UART.

Otro de los objetivos que persigue Libelium es el de permitir una comunicación de varios kilómetros de

distancia entre los dispositivos. Para ello se integraron módulos de comunicación de alta sensibilidad que utilizan sistemas inalámbricos de distintas frecuencias (2.4GHz, 868MHz, 900MHz) a fin de evitar interferencias con redes Wifi en zonas urbanas. Hay siete tipos de módulos XBee [8] y todos ellos se caracterizan por tener una alta sensibilidad de recepción y gran potencia de transmisión, con un rango de alcance que varía desde los 500 metros en el caso del módulo XBee-ZigBee (ver Figura 2) hasta los 12 Km en el caso del módulo XBee-868.



Figura 2: Módulo XBee-ZigBee

Cada módulo XBee puede funcionar como “Coordinador”, “Router” o “End Device” dentro de la red.

- **Coordinador:** será el encargado de crear la red, esto es buscar un canal de frecuencia libre para la comunicación inalámbrica y establecer un identificador de red denominado PAN ID. Una vez seleccionado el canal y el PAN ID, el Coordinador permitirá a los Routers y End Devices unirse a la red para que puedan transmitir o recibir datos, o encaminarlos.
- **Router:** puede, una vez conectado a la red, permitir a otros Routers y End Devices unirse a la red así como ejercer tareas de encaminamiento, transmisión y recepción de datos.
- **End Device:** necesita de un padre, bien sea un Router o un Coordinador, para poder transmitir y recibir datos, sin embargo, es el único que puede “dormir”, es decir, entrar en modo de bajo consumo. El padre deberá por tanto almacenar en un *buffer* los datos destinados al End Device mientras éste se encuentre dormido y entregárselos cuando despierte.

A través de la herramienta X-CTU se pueden programar los módulos para que funcionen de una de estas tres formas.

Con respecto a la topología, existen principalmente dos posibilidades: árbol (Figura 3) y anillo.

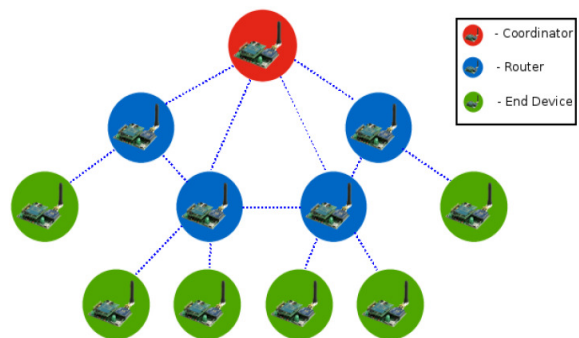


Figura 3: Ejemplo de topología en árbol

3. Aplicación: Monitorización de sistemas remoto

Para mostrar las utilidades que ofrece Waspnote y la posibilidad de integración con otras tecnologías se ha desarrollado una aplicación que consiste en monitorizar el estado de una red de sensores inalámbrica desde un dispositivo móvil. El objetivo es tener acceso desde un teléfono móvil a las medidas de los sensores en tiempo real. Para conectar los *nodes* (ZigBee) con el teléfono (Wifi/3G/GPRS) es necesario utilizar un equipo pasarela, concretamente Meshlium [9].

Meshlium es un router que ejecuta un sistema Linux embebido y que cuenta con 5 interfaces de radio diferentes: Wifi 2.4 GHz, Wifi 5GHz, 3G/GPRS, Bluetooth y ZigBee. Además puede integrar un módulo GPS y ser alimentado por luz solar y batería. Meshlium, a diferencia de los routers convencionales, está diseñado para ser resistente al agua y soportar condiciones medioambientales adversas. Por otra parte, la potencia de transmisión de la interfaz Wifi es significativamente mayor que en los routers de uso doméstico pudiendo alcanzar localizaciones muy remotas.

Meshlium utilizará su interfaz ZigBee para convertirse en el nodo Coordinador de una red de Waspnotes. Una vez formada la red, Meshlium podrá recoger datos enviados por los *nodes* y almacenarlos en una base local o externa o enviarlos a través de internet usando una conexión Wifi, Ethernet o 3G/GPRS [10].

Aprovechando la capacidad de Meshlium de funcionar simultáneamente como Coordinador de una red de *nodes* y como punto de acceso Wifi, se ha decidido implementar una aplicación para un teléfono Android que permita capturar los valores medidos por los sensores del Waspnote y mostrarlos en la pantalla (Figura 4). Haciendo uso de Android SDK, que proporciona las librerías y herramientas de

desarrollo necesarias para la creación de aplicaciones para la plataforma Android, se ha implementado una aplicación que se conecta a Meshlium a través de su interfaz Wifi y le solicita información acerca un determinado nodo de la red de *motes*. Por su parte Meshlium, al recibir la petición de información, la traslada al *Mote* correspondiente haciendo uso de su interfaz ZigBee.

Para que todo esto sea posible, se ha programado una rutina en lenguaje C que el sistema embebido Linux ejecuta en *background*. Esta rutina se encarga de abrir la interfaz Wifi y se mantiene a la espera de peticiones por parte del dispositivo que las solicite, en este caso un Smartphone. Al recibir dicha petición, la procesará y la reenviará a través de la interfaz ZigBee al nodo del que se desee obtener la información. Este nodo obtendrá los valores requeridos de los sensores y se los entregará a Meshlium, el cual a su vez se los proporcionará al Smartphone que los mostrará en la pantalla. Para implementar la comunicación Wifi entre el Smartphone y Meshlium se hace uso del protocolo TCP/IP.

Cada uno de los Wasmotes por su parte tendrá programada una rutina que se mantendrá a la espera de peticiones por parte de Meshlium, leerá los valores de los sensores pertinentes, y los enviará a Meshlium a través de ZigBee.

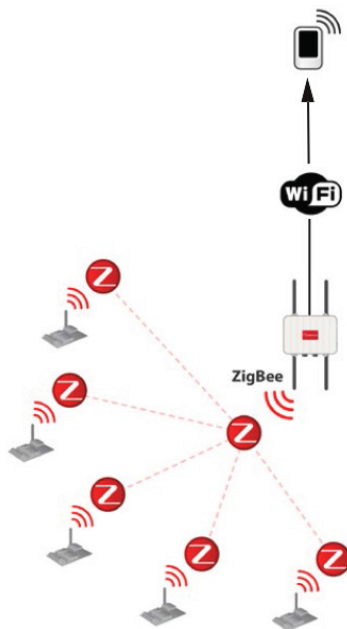


Figura 4: Esquema de funcionamiento de la aplicación

La Figura 5 muestra algunas capacidades de la aplicación. A la izquierda se observa la pantalla principal que nos da acceso a cada uno de los *motes*. Las medidas que se obtienen de un *mote* en particular

se reciben en el teléfono, tal y como se observa en la figura de la derecha.



Figura 5: Aplicación para la monitorización de sistemas remotos

4. Aplicación: Plataforma para el control de flotas de vehículos

La segunda aplicación que se ha realizado utilizando esta tecnología consiste en acoplar los *motes* a una serie de vehículos autónomos, en este caso pequeños coches impulsados por servomotores Pololu (Figuras 6 y 7).

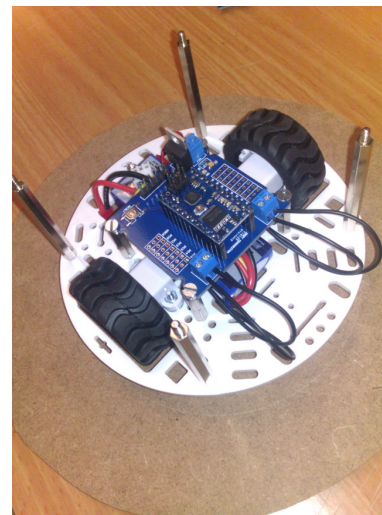


Figura 6: Vista de la parte inferior del vehículo

El movimiento de estos vehículos en el plano XY se describe mediante el conocido modelo del unicycle:

$$\dot{x} = F(t)\cos(\theta(t)),$$

$$\dot{y} = F(t)\sin(\theta(t)),$$

$$\dot{\theta} = T(t),$$

donde $x(t)$, $y(t)$ y $\theta(t)$ son, respectivamente, las coordenadas y el ángulo de orientación. Las señales

de entrada $F(t)$ y $T(t)$ representan la velocidad lineal y angular. Además de utilizarse para vehículos terrestres, el unicyclo se ha empleado para modelar submarinos autónomos del tipo deslizadores [7].

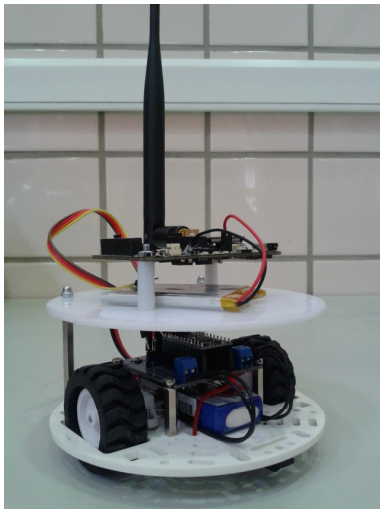


Figura 7: Vista lateral del vehículo con el mote

Las señales de entrada a los servos son generadas por el microcontrolador local mediante PWM, de forma que para realizar giros se les aplica una señal diferente a cada uno de los ejes de giro.

Además, el microcontrolador local debe comunicarse con el Wasmote (ver Figura 7), ya que será éste el encargado de generar las consignas de alto nivel. Para llevar a cabo dicha conexión entre micros, se utiliza el bus I2C de comunicación serie.

Las aplicaciones son variadas. Por ejemplo, haciendo uso del acelerómetro tri-axial incluido en Wasmote, se podrán detectar por ejemplo golpes y caídas, o cuándo el robot está subiendo y bajando cuestas.

El objetivo final que se persigue es coordinar una flota de vehículos para que maniobren en conjunto persiguiendo algún objetivo global (Figura 8).



Figura 8: Flota de vehículos autónomos

En este tipo de aplicaciones es de especial importancia mantener una correcta formación de los vehículos. Para ello, se hace indispensable conocer la posición de los robots, si no de forma absoluta, al menos su posición relativa a sus vecinos. Para resolver este problema, existen las siguientes soluciones:

- Sensores de ultrasonidos: miden la diferencia de tiempos entre la emisión y la recepción de un sonido. Aunque son bastante precisos, hay que tener en cuenta las zonas ciegas y las falsas alarmas.
- Sistemas de localización GPS: miden posiciones absolutas, pero con un margen de error que no es despreciable.
- Sistemas de visión artificial: de gran carga computacional, pueden obtener resultados muy adecuados.

La elección es muy dependiente del tipo de problema que quiera resolverse.

5. Conclusiones

Este artículo ha presentado las posibilidades que ofrece Libelium para desarrollar diversas aplicaciones basadas en redes de sensores. En primer lugar, se ha gestionado el acceso a una red de sensores inalámbricos mediante un *smartphone*. Finalmente, se ha construido una plataforma adecuada para el control de flotas de vehículos. En los siguientes meses, se probarán diversos algoritmos de control, como el propuesto por los autores en [11].

Agradecimientos

Los autores agradecen al Ministerio de Ciencia y Tecnología (DPI2010-19154) y la Junta de Andalucía (P09-AGR-4785) por financiar este trabajo. Igualmente, agradecen a ESIBot su colaboración en el desarrollo de los vehículos.

Referencias

- [1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002), "Wireless sensor Networks: a survey", *Computer Networks*, 38, pp: 393-422.
- [2] Martínez, M., Blasco, X., Herrero, J. M., Ramos, C., Sanchis, C. (2005), "Monitorización y Control de Procesos. Una Visión Teórico-Práctica Aplicada a Invernaderos", *Revista Iberoamericana de Informática y Robótica Industriales*, 2(4), pp: 5-24.
- [3] Millán, P., Orihuela, L., and Jurado, I., Vivas, C., Rubio, F. R. (2013), "Distributed estimation in networked systems under periodic and event-based

communication policies”, International Journal of Systems Science, DOI: 10.1080/00207721.2013.775387.

[4] Olfati-Saber, R. (2007), “Distributed Kalman filtering for sensor networks”, 46th IEEE Conference on Decision and Control, New Orleans, USA, pp: 5492-5498.

[5] Maestre, J. M., Muñoz de la Peña, D., Camacho, E. F. (2011), “Wireless sensor network analysis through a coalitional game: application to a distributed Kalman filter”, International Conference on Networking, Sensing and Control, Delft, The Netherlands, pp: 227-232.

[6] Millán, P., Orihuela, L., Vivas, C., Rubio, F. R., Dimarogonas, D. V., Johansson, K. H. (2013), “Sensor-network-based robust distributed control and estimation”, Control Engineering Practice, <http://dx.doi.org/10.1016/j.conengprac.2013.05.002i>

[7] Briñón-Arranz, L., Seuret, A., Canudas de Wit, C. (2009), “Translation control of a fleet circular formation of AUVs under finite communication range”, 48th IEEE Conference on Decision and Control y 28th Chinese Control Conference, Shanghai, China, pp. 8345–8350.

[8] Libelium Comunicaciones Distribuidas S.L. (2013), “Waspote Technical Guide”.

[9] Libelium Comunicaciones Distribuidas S.L. (2012), “Meshlium Technical Guide”.

[10] Libelium Comunicaciones Distribuidas S.L. (2013), “Wireless Sensors Networks with Waspote and Meshlium”.

[11] Millán, P., Orihuela, L., Jurado, I., Rubio, F. R. (2013), “Formation control of autonomous underwater vehicles subject to communication delays”, IEEE Transactions on Control Systems Technology, DOI: 10.1109/TCST.2013.2262768.