

# Estructura para modelado, simulación y generación automática de código para aplicaciones de **redes inalámbricas de sensores**

A pesar de que el diseño basado en modelos se está convirtiendo en un estándar para varios dominios que tienen limitaciones similares a las de las redes inalámbricas de sensores, hay una falta de herramientas para este propósito en el mundo de las redes inalámbricas de sensores.



Por Stefano Olivieri, The MathWorks, escrito con la colaboración de Mostafizur Mozumdar, Luciano Lavagno y Laura Vanzago.

Las redes inalámbricas de sensores (en inglés, Wireless Sensor Networks, WSN) se han convertido en uno de los temas de investigación con mayor actividad y pueden convertirse en uno de los factores de desarrollo de las tecnologías microelectrónicas tanto en la actualidad como en el futuro. Esta tecnología emergente ofrece un enorme potencial para muchas

áreas de aplicación, entre las que se incluyen el medioambiente, la medicina, el sector militar, el transporte, el entretenimiento, la gestión de crisis, la defensa y la domótica.

Como planteamiento general, los usuarios recogen información de una WSN inyectándole consultas y recolectando los datos de los nodos sensores, también llamados “motes” (del inglés, Mobile Transmission Element). Un solo mote puede tener diferentes componentes (sensor, procesador, dispositivo de comunicación, dispositivo de actuación y batería) integrados y empaquetados en un mote miniaturizado. Una WSN generalmente consta de miles de estos nodos, que se comunican a través de canales inalámbricos para compartir información y llevar a cabo procesos cooperativos.

Debido a la necesidad de funcionamiento sin operador en entornos remotos e incluso potencialmente hostiles, las redes inalámbricas de sensores deben tener un consumo de energía mínimo. Los mayores retos en el campo de las redes inalámbricas de sensores son la medición y recolección de datos de los sensores y su evaluación para elaborar información fiable (que permita, por ejemplo, generar alarmas o tomar ciertas decisiones) y la reducción, al mismo tiempo, del consumo de energía.

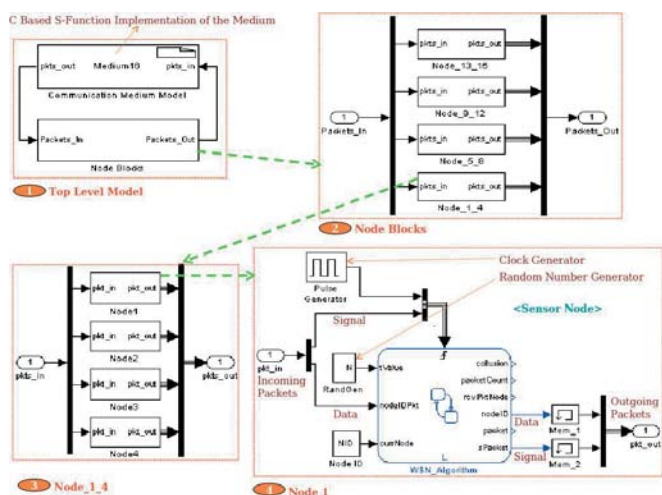


Figura 1: estructura simple de simulación de WSN.

## Los mayores retos del desarrollo de aplicaciones de redes inalámbricas de sensores

En la actualidad, los investigadores están desarrollando aplicaciones de software a diferentes niveles para redes inalámbricas de sensores. No obstante, aún persisten serios inconvenientes para el éxito de las WSN en aplicaciones comerciales.

A día de hoy hay disponibles dos tipos primarios de paquetes de análisis funcional. Uno de ellos es muy específico para una plataforma y sistema operativo (en adelante SO) (por ejemplo, TOSSIM), que ofrece una API binaria para modelar el SO y los motes, con capacidades limitadas para reutilizar los modelos de canales, el seguimiento, las estadísticas de recolección, etc., existentes. El otro tipo incluye simuladores de red genéricos (tales como OmNet, NS, etc.), algunas veces ampliados con modelos adaptados a las radios y canales usados por las WSN.

Ambos tienen desventajas importantes en la fase de desarrollo de la aplicación. Con el primer grupo, es difícil portar una aplicación a una plataforma diferente, por ejemplo de TinyOS a MANTIS, o a una plataforma compatible con ZigBee, sin tener que reescribir mucha cantidad de código y hacer una gran cantidad de depuraciones. El segundo grupo aún deja mucho código detallado dependiente de la plataforma que se debe desarrollar y depurar. El uso integrado de un simulador de red seguido de un simulador de plataforma es la ruta más comúnmente utilizada, pero también requiere que se porte el código entre los diferentes entornos. Además,

si se encuentra un error al final de la fase de desarrollo e integración, la única solución es la depuración mediante los LED que es terriblemente tediosa.

Para resolver estos problemas, los ingenieros de sistemas necesitan poder modelar aplicaciones utilizando abstracciones de alto nivel y simularlas usando topologías configurables y realistas para la propia red. Además, sería deseable contar con herramientas apropiadas para generar código automáticamente para varios sistemas operativos de destino.

**“ En TinyOS, una entidad conceptual se representa mediante dos tipos de componentes: módulo y configuración**

## Una estructura de simulación de referencia

Como resultado del proyecto de investigación conjunta llevado a cabo por STMicroelectronics, Politecnico di Torino y The MathWorks, hemos definido un prototipo de estructura para modelado, simulación y generación de código para aplicaciones de redes de sensores basada en las herramientas de The MathWorks. En el nivel superior, se puede construir un modelo de red de sensores con Simulink® que sirve como simulador basado en el tiempo para sistemas dinámicos. Este modelo tiene dos componentes principales: el modelo del canal inalámbrico (en adelante, llamado “medio de comunicación”) y el bloque de nodo.

Se puede implementar el bloque de medio de comunicación en C, de modo que pueda modificarse para reutilizar los modelos de canal y conectividad existentes. Ahora estamos trabajando, por ejemplo, en refinarlo usando los estándares físicos de ZigBee y Medium Access Control.



Figura 2: estructura para modelado, simulación y generación de código para aplicaciones de WSN.

El bloque de nodo contiene un número de motes, cada uno de los cuales está completamente parametrizado y contiene un modelo de la plataforma de hardware y software que incluye, por ejemplo, un temporizador y los LED, así como un bloque parametrizado de Stateflow® que representa la aplicación. Stateflow es un lenguaje de máquina de estado finito que se utiliza para modelar y simular los diagramas de flujo y la lógica de control. Aquí, implementa un algoritmo (aplicación, middleware o controladores de dispositivos) que se ejecuta en cada mote. El temporizador, por ejemplo, se utiliza para generar eventos temporales para el algoritmo de Stateflow.

Es posible modelar redes de sensores que ejecuten aplicaciones idénticas o diferentes en cada nodo. Para modelar una nueva aplicación de red de sensores basada en esta estructura, un desarrollador de aplicaciones sólo necesita modificar la implementación del algoritmo modelo (objeto de biblioteca de Stateflow) y establecer los parámetros de conectividad de los nodos en el bloque de medio de comunicación. A continuación, se puede iniciar la simulación y se pueden recopilar datos estadísticos usando diagramas de estado animados y visualizaciones gráficas y numéricas para llevar a cabo el análisis funcional del algoritmo. La implementación del algoritmo se puede refinar si el análisis de los resultados sugiere que debe hacerse. Finalmente, el desarrollador obtendrá un modelo refinado que represente el comportamiento deseado.

### Generación automática de código multiplataforma

Después del análisis funcional del algoritmo, el paso siguiente es la generación automática de código para cada una de las plataformas admitidas por la WSN a partir de la representación del algoritmo en Stateflow, utilizando una personalización del software RealTime Workshop Embedded Coder, que puede generar código ANSI C embebible para los bloques de Simulink y Stateflow.

### Ejemplos de sistemas operativos para aplicaciones de WSN: TinyOS y MANTIS

TinyOS y Mantis son dos sistemas operativos que representan paradigmas ejemplares para el desarrollo de aplicaciones embebidas en redes inalámbricas de sensores.

El modelo de programación de TinyOS se basa en componentes. En TinyOS, una entidad conceptual se representa mediante dos tipos de componentes: módulo y configuración. Cada componente implementa interfaces. La interfaz declara la firma de los comandos y eventos que deben implementar el proveedor y el usuario de la interfaz, respectivamente. Los eventos son las abstracciones de software de eventos de hardware o software, tales como la recepción de un paquete, la finalización del muestreo de un sensor, etc. Por otro lado, los comandos se utilizan para desencadenar una opera-

ción, como iniciar la lectura de un sensor o la comunicación por radio para recepción o transmisión, etc. Las aplicaciones de TinyOS están escritas en nesC, que es una extensión del lenguaje C.

MANTIS es un sistema operativo ligero, multiproceso y multitarea, diseñado para redes de sensores distribuidas con limitaciones de energía. El planificador de MANTIS admite la preferencia de procesos, que permite al sistema operativo cambiar entre procesos activos sin esperar. De este modo, la capacidad de respuesta del sistema operativo a los eventos críticos puede ser más rápida que la de TinyOS, que no la admite. El planificador de MANTIS determina la prioridad en base al método de circuito

cíclico (round robin). El kernel y las API de MANTIS están escritos en C estándar.

“ MANTIS es un sistema operativo ligero, multiproceso y multitarea, diseñado para redes de sensores distribuidas con limitaciones de energía

### Implementación de la aplicación de WSN

El cuerpo computacional, tanto de los procesos y las tareas de MANTIS como de los comandos de TinyOS, está escrito básicamente en C. En MANTIS, la integración es sencilla, debido a que la salida ANSI C de RealTime Workshop Embedded Coder se puede usar directamente como un proceso del usuario. En TinyOS, los programas están compuestos por abstracciones de alto nivel, tales como componentes, módulos e interfaces que no existen en ANSI C. Los cuerpos de estas abstracciones, sin embargo, están escritos en C, de modo que todo lo que se necesita es un convertidor de código que coja el código C como entrada y lo divida en secciones (inclusiones, definiciones, funciones, etc.) y utilice estas secciones para generar el código nesC.

Se puede utilizar el Target Language Compiler (TLC) de The MathWorks para implementar este tipo de convertidor de código. Proporciona mecanismos para generar código específico para cada plataforma tomando secciones (tales como inclusiones, definiciones, funciones, etc.) del código ANSI C y añadiendo también código personalizado para la plataforma de destino. Para facilitar el desarrollo independiente de la plataforma, también se puede suministrar un conjunto de funciones de biblioteca genéricas que pueden utilizarse desde Stateflow para acceder a la funcionalidad específica del sistema operativo de la plataforma (como por ejemplo, led\_toggle, led\_on, led\_off, sendPacket, receivePacket). Al trabajar con Stateflow, los desarrolladores de aplicaciones no tienen que pensar en la implementación real de estas funciones genéricas en TinyOS o en MANTIS, dado que ya han sido implementadas en la biblioteca TLC y pueden destinarse a cualquier sistema operativo. Utilizando los scripts de TLC, un desarrollador puede generar automáticamente una aplicación de TinyOS o MANTIS y, a continuación, puede compilar y ejecutar esa aplicación para la plataforma de destino sin ninguna modificación. Esto es más potente que una API de abstracción independiente del sistema operativo y del hardware, porque TLC puede reescribir y modificar realmente el código fuente para dividirlo y adecuarlo a la plataforma subyacente. De este modo, se logra una mayor opti-

mización, por ejemplo, que la que puede conseguir un compilador y permite evitar la cantidad de reestructuración manual de código que un programador tendría que realizar para portar una aplicación entre plataformas de hardware y software muy diferentes

Los resultados de la implementación de un algoritmo de WSN típico en una plataforma TinyOS muestran que el incremento del tamaño del código debido a la generación automática de código en comparación con la escritura manual de código es de apenas un 5%. La ventaja de usar esta estructura, comparada con la escritura manual de código, para aplicaciones de WSN es que sólo son necesarias unas pocas horas para tener la nueva aplicación funcionando. En otras palabras, uno puede concentrarse realmente en utilizar el tiempo para modelar y simular a nivel funcional y, luego, la generación de código, la compilación y la ejecución en dos plataformas muy diferentes se llevan a cabo automáticamente y son extremadamente rápidas.

### Conclusión

Una infraestructura completa para el modelado, simulación y generación de código multiplataforma basada en las herramientas de The MathWorks permite a los ingenieros de sistemas diseñar aplicaciones de WSN de forma efectiva y utilizarlas en una red inalámbrica.

Estas herramientas ponen a disposición de los desarrolladores de aplicaciones, bibliotecas completas para el procesamiento de señales digitales y la simulación del comportamiento de los algoritmos de control, junto con una amplia variedad de herramientas de depuración y análisis tales como diagramas de estado animados y visualizaciones de datos y gráficos. Además, con esta infraestructura los usuarios pueden generar automáticamente el código completo de la aplicación para varios sistemas operativos de destino a partir del mismo modelo simulado y depurado, sin tener que pensar en los detalles de la implementación en la plataforma de destino.

Finalmente, la reimplementación de una nueva aplicación (por ejemplo, la generación de código, compilación y ejecución) en una plataforma específica es automática y muy rápida, con sólo un pequeño aumento en el tamaño del código si se lo compara con código escrito a mano. <<

Stefano Olivieri es ingeniero de aplicaciones senior en The MathWorks. Mostafizur Mozumdar es estudiante del programa de Doctorado en el Politecnico di Torino, Italia. Luciano Lavagno es profesor asociado del Politecnico di Torino, Italia. Laura Vanzago trabaja en STMicroelectronics, Agrate Brianza, Italia.

Web: [www.themathworks.es](http://www.themathworks.es)

# Más allá de la excelencia

- > Miles de productos adicionales añadidos de Epcos, Molex, Panasonic, ST, Tlm Tyco Electronics y Vishay
- > Más de 100 nuevas tecnologías de marca añadidas cada día incluyendo: Intel Atom Processor, Jennic ZigBee-Pro y RFMD RF205x

Diseñe con el mejor



[www.farnell.com/es](http://www.farnell.com/es)

- La gama de productos más amplia de la distribución europea, con más de 480.000 productos de 3.500 fabricantes
- i-Buy - La solución de eProcurement gratuita
- eCotiza - Reciba ya sus cotizaciones online
- Sin cantidad mínima de pedido, y envío en 24/48h
- La más reciente información en legislación
- element14 - El primer portal tecnológico y comunidad online para ingenieros de diseño

A Premier Farnell Company



70  
1939-2009  
FARNELL

Diseñe con el mejor