

Capítulo 6

Conclusiones y líneas futuras

En este trabajo se ha realizado un programa o aplicación para la monitorización de la energía eléctrica usada en las tomas de un edificio, así como la medición de los armónicos que pueda contener la red (distorsión). Para ello, ha sido necesario un aprendizaje de un lenguaje de programación para sistemas operativos de tiempo real (orientados a eventos). Estudio de técnicas y protocolos de comunicación en redes de sensores inalámbricas de bajo consumo (recolección y diseminación de datos, reprogramación de una red completa de forma inalámbrica...). Para este desarrollo, se han usado dispositivos comerciales, lo que añade un enfoque cercano al que existe en el mercado de estas tecnologías.

Un trabajo de este tipo, relacionado con redes inalámbricas, trata un tema de investigación actual y con muchas aplicaciones. El lenguaje empleado permite un gran potencial a la hora de desarrollar aplicaciones.

A continuación se describen, a modo de resumen, las tareas más importantes llevadas a cabo en este trabajo. Para finalizar, se dan algunas pautas para una posible continuación del trabajo, desde mejorar los algoritmos y cálculos implementados hasta el desarrollo de prototipos comerciales.

6.1 Tareas realizadas

Bloques de tareas llevadas a cabo durante la realización del trabajo, acompañadas de algunas indicaciones u observaciones.

- Aprendizaje del lenguaje de programación nesC, para el desarrollo de aplicaciones en TinyOS. Es uno de los primeros pasos a realizar, pues la mayor parte del trabajo consistirá en programar con el mencionado lenguaje. En paralelo, se han desarrollado algunos manuales o guías de referencia: instalación y puesta en marcha del entorno de desarrollo, tutorial de iniciación a TinyOS... así como presentaciones con ejemplos de uso. De hecho, el Capítulo 3 se puede utilizar como manual para principiantes en nesC.

Es importante tener presentes ciertos problemas que pueden aparecer en la instalación del entorno de desarrollo para TinyOS. Al utilizar un sistema Windows, es necesario utilizar el emulador Cygwin. Aunque en la web documentación de TinyOS se describen los pasos a seguir, cada PC es diferente y puede no tener determinados ficheros necesarios.

- Basándose en el trabajo previo de (Larios Marín, y otros, 2009), adaptar y ampliar objetivos alcanzados en este trabajo. Algunos de estos cambios han sido: modificación de la red de datos, usando sensores inalámbricos para realizar las mediciones; adición de funcionalidad de monitorización de armónicos y distorsión de las señales, pudiendo actuar en consecuencia; uso de MATLAB para realizar la gestión y control de los motes...

Aunque el trabajo utilizado como punto de partida utiliza C para programar, y nesC es parecido, las diferencias existentes entre ambos hace prácticamente imposible la reutilización del código, obligando por tanto a escribir un código fuente desde cero.

- Estudio más en profundidad de un *hardware* de sensores específico. Se trata de aprender y conocer las características de funcionamiento del mote Tmote Sky, así como del dispositivo que actúa como pasarela Tmote Connect. La fuente de información básica son los documentos elaborados por el fabricante (ahora Sentilla) y las hojas de características (*datasheet*) de los chips que componen el sensor.

También se hace una revisión de los protocolos de comunicación inalámbricos más usados, fundamentalmente a través de los estándares publicados al respecto.

- Desarrollo y programación de la aplicación. Se trata del punto más extenso del trabajo. Partiendo de las funciones requeridas para la red, hay que adaptarlas a los componentes disponibles en TinyOS.

En esta tarea es frecuente encontrarse con errores que alteran o imposibilitan el funcionamiento normal de la aplicación. Esto puede deberse a fallos en alguna instrucción, uso incorrecto de las interfaces, o incluso errores en los propios componentes⁶⁵.

- Realización de pruebas de carácter práctico. En primer lugar las pruebas se centran en validar partes el código fuente de forma individual (algoritmo FFT, medidas de potencia, sensores auxiliares...). Posteriormente, se simulan señales tal y como se recibirían desde la red eléctrica. Por último, se testea la comunicación inalámbrica con distintas configuraciones.

Para las pruebas reales se estudiaron principalmente a dos alternativas. La desventaja de la opción de los Laboratorios es que los generadores de onda no son capaces de añadir armónicos a las señales, y tampoco se puede ajustar el desfase entre señales. La otra opción planteada fue usar LabVIEW para generar las señales analógicas. Tuvo que ser descartada, pues la resolución que presentaba el convertidor digital-analógico de la tarjeta no era suficiente y las señales analógicas no tenían la calidad suficiente. Se solventó usando un *software* de edición de audio, aprovechando estas señales como entradas analógicas para el sensor. LabVIEW se utiliza por tanto para recibir y procesar estas señales y poder comparar con el mote.

⁶⁵ En (TinyOS) se recoge la modificación del archivo *Msp430Adc12.h*, que contenía algunos errores en la declaración de constantes.

6.2 Líneas futuras de desarrollo

Se detallan algunas propuestas que continúan con el trabajo realizado y pueden mejorar el sistema de medidas planteado.

- Adaptación del mote para la realización de un posible prototipo (llegando a una posible comercialización). Gracias al pequeño tamaño de estos módulos, y a que se dispone del diagrama electrónico del mismo se puede diseñar un dispositivo que realice las mediciones y operaciones oportunas ocupando el mínimo espacio posible. Una propuesta de diseño físico es adaptarlo a la forma de una toma de corriente de España (según Figura 6.1), de forma que su instalación en un edificio sea lo más sencilla posible.



Figura 6.1 Enchufes utilizados en España

A la izquierda, el tipo C o Europlug (tensiones de hasta 230 V e intensidad máxima 2.5 A). A la derecha, tipo F conocido como Schuko (230 V e intensidad 16 A).

Este prototipo deberá incluir otros elementos adicionales al sensor:

- Sistema de alimentación, puede usarse la misma tensión adaptada de las tomas eléctricas y una pequeña batería para los momentos de interrupción del suministro. Para la transformación de corriente alterna a continua podría diseñarse una fuente de tipo conmutado (ocupan poco espacio y son más eficientes que las tradicionales).
- Transductores de tensión e intensidad. Para la medida de tensión puede usarse un pequeño transformador (suele ser de gran tamaño aunque proporciona aislamiento galvánico) o usar un puente resistivo (más pequeño pero sin aislamiento y con pérdidas). En el caso de la intensidad existen más alternativas: *shunt* coaxial, transformador de corriente, sensores de efecto *Hall*, o una bobina de *Rogowski* (Using Rogowski coils for transient current measurements, 1993). Esta última supera casi en todos los aspectos al resto de

transductores, por lo que sería una buena opción. Estos componentes deberán de ir acompañados de la circuitería necesaria para adaptar las medidas al rango de medida del sensor (acondicionamiento de señal).

- Control de cargas externas. En el trabajo actual se utilizan dos salidas digitales de propósito general que permiten activar/desactivar una carga que se pudiera tener. Para adaptar la señal digital también existen varias alternativas: relé, triac...
- Mejoras en los algoritmos de cálculo utilizados en el sensor. Si bien el bloque de cálculo de la FFT está en cierta forma optimizado (reducción de operaciones, evitación de operaciones trigonométricas y decimales, adaptación a secuencias reales...), es posible ajustar este algoritmo para reducir el consumo de recursos en la CPU y un menor uso de la memoria RAM, con el consiguiente ahorro energético superior.

Seguramente pueda optimizarse también el bloque de cómputo en señales de tensión e intensidad (valores medios, eficaces, potencias...), procurando no usar tantos bucles FOR.

- Búsqueda y comparación con otros modelos de sensores inalámbricos (A Comparative Review of Wireless Sensor Network Mote Technologies, 2009). Aunque el modelo utilizado para desarrollar el trabajo es capaz de implementar la mayoría de las funciones propuestas, puede pensarse en utilizar otra plataforma (con más memoria de programa, o más resolución en el CAD del microcontrolador⁶⁶). Aunque la programación en nesC es similar en todos los casos, lógicamente habrá que hacer cambios en el código.
- Desarrollo de una plataforma para la recepción/envío y gestión de datos de la red inalámbrica, por ejemplo con un sistema SCADA. El utilizado para este trabajo (MATLAB) no sería óptimo para estos fines, si bien desde el punto de vista académico sí es útil.

⁶⁶ Respecto al convertidor A-D, el microcontrolador de los modelos Tmote Sky (MSP430F1611) tiene una resolución de 12 bit, mientras que la mayoría de otros dispositivos solo cuentan con 10 bit.

