

Una Experiencia Piloto de Red de Sensores Inalámbricos para Aplicaciones Agronómicas

Diego Baccino, Carolina Etchart, Miguel Tasende, Pablo Mazzara, Fernando Silveira, Leonardo Steinfeld

Resumen—Este trabajo presenta una primera experiencia realizada en nuestro país de desarrollo y prueba de una red de sensores inalámbrica, particularmente orientada a aplicaciones agropecuarias. Se describe el diseño utilizado, ejemplificando las posibilidades de esta tecnología, así como los primeros resultados de validación en campo. La aplicación piloto elegida para probar la red consiste en el monitoreo de temperatura en parideras de cerdos de campo en el centro Regional Sur de la Facultad Agronomía de la Universidad de la República. Concluimos que el sistema logra una precisión mejor que el sistema existente en el lugar, basado en “dataloggers”.

Palabras Clave—Redes de sensores inalámbricos.

I. NOMENCLATURA

Mote – Nodo de una red de sensores inalámbricos.

WSN – “Wireless Sensor Network” (red de sensores inalámbricos)

MultiHop – Protocolos de comunicación que permiten aumentar el alcance de los nodos mediante un sistema de reenvío de paquetes.

II. INTRODUCCIÓN

EN este artículo se presenta una primera experiencia en la implementación de una red de sensores inalámbricos para la medición de temperatura. Se expone brevemente un caso de aplicación, de qué forma fue posible combinar módulos de hardware y software existentes en la búsqueda de una solución, y los resultados efectivos del sistema, junto con las conclusiones obtenidas y proyecciones de trabajos futuros.

El sistema implementado se basa en una red de sensores inalámbricos (WSN). Una WSN es una red inalámbrica consistente de nodos (“motes”) autónomos, que utilizan sensores para monitorear de forma cooperativa ciertas características físicas de un objeto o del ambiente. Se utilizan para medir temperatura, vibraciones, sonido, presión,

Este trabajo fue financiado por los Proyectos PDT 17/17 “Sensores Inalámbricos Integrados de Bajo Consumo” y PDT 69/08 “WiseMAN: Redes de sensores inalámbricos para aplicaciones agropecuarias y médicas”. Parte del mismo se enmarca en un proyecto de fin de carrera de Ingeniería Eléctrica de 2007 (Instituto de Ingeniería Eléctrica, de la Universidad de la República; República Oriental del Uruguay), en el que participaron Diego Baccino, Carolina Etchart y Miguel Tasende como estudiantes de grado, Pablo Mazzara y Fernando Silveira como tutores. Leonardo Steinfeld, Pablo Mazzara y Fernando Silveira son integrantes del Instituto de Ingeniería Eléctrica de la Universidad de la República (ROU).

movimiento, agentes contaminantes, y más. Actualmente las WSN son utilizadas en el monitoreo de habitats y medio ambiente, aplicaciones médicas, domótica, y control de tráfico.

En general, y nuestro caso no es una excepción, es parte de la filosofía de las WSN, que las mismas sean de fácil instalación, que cuenten con protocolos de multisalto (MultiHop), que sean robustas frente a variaciones importantes en la calidad de enlace entre nodos (haciendo uso de caminos alternativos para el ruteo, etc.), que incorporen automáticamente nuevos nodos, y sean capaces de seguir funcionando ante la desaparición de nodos. Nuestro sistema no requiere ningún tipo de configuración de red, o de inicialización de nodos. Alcanza con encender los “motes” en una cierta área geográfica y los mismos se encargan de organizar la red, y hacer llegar los datos a la base. Incluso cuenta con un sistema de versiones de reconfiguración (basado en Trickle), que permite que los nuevos nodos agregados funcionen con las mismas especificaciones que toda la red (esto será expuesto en la sección III).

III. TRABAJO REALIZADO

A. Caso de Aplicación

En la producción de lechones la comodidad que se le puede ofrecer a la cerda durante el parto y la lactancia son parte fundamental para lograr un buen número de lechones al destete. Los lechones en los primeros días de vida son muy sensibles a las bajas temperaturas, así como los cerdos adultos lo son a las altas. La Facultad de Agronomía, de la Universidad de la República, a través de la Estación Experimental Centro Regional Sur, ha diseñado un tipo de construcción -paridera de campo móvil- considerada apta para las condiciones agrometeorológicas de nuestro país y al alcance de cualquier productor de cerdos.

La evaluación de las condiciones microclimáticas logradas dentro de la paridera es realizada luego de efectuar mediciones de temperatura permanentemente en diferentes parideras bajo distintas condiciones.

Típicamente, para realizar el relevamiento de las temperaturas, un usuario utilizando un termómetro mide la temperatura manualmente recorriendo cada una de las parideras. Es común también el uso de “data loggers” que son instalados y dejados en el sitio a medir, pero luego es

necesario llevar los mismos hasta un computador para la descarga de los datos.

Por esta razón es muy útil contar con un sistema de medición de temperatura que mida continuamente y de manera automática, pudiendo visualizar los datos en tiempo real y guardarlos en memoria para posterior análisis.

B. Implementación de una solución

1) Nodos

Para la elección de la plataforma de hardware, basándonos en el análisis que se realiza en [6] sobre los pro y los contra de las diversas opciones disponibles en el momento, utilizamos los nodos “Tmote Sky” de Moteiv (Fig. 1). Una de las ventajas que tienen es el hecho de que estos “motes” están pensados (y tienen aplicaciones desarrolladas) para ser usados con el sistema operativo TinyOS, el cual provee gran cantidad de abstracciones del hardware. Esto permite reutilizar el software creado en otra plataforma de hardware que implemente las mismas abstracciones.

Los nodos cuentan con un procesador de muy bajo consumo y un transceptor de radio como elementos básicos. En nuestro trabajo, adaptamos un sensor TMP175 (ver [4]) para conectarlo a la interfaz I2C del procesador. Como último componente de hardware, es necesario para el funcionamiento del sistema, contar con un PC en el que se pueda ejecutar una aplicación Java, que se ocupa de recibir los datos de la red y controlar su funcionamiento.

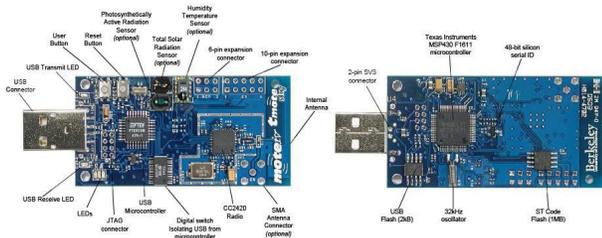


Fig. 1. Vista superior e inferior de un “mote” Tmote Sky de Moteiv.

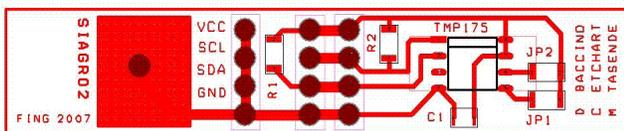


Fig. 2. Esquemático del circuito implementado para el sensor.

El firmware del sistema fue el resultado de seleccionar aplicaciones ya existentes con implementaciones propias de algunos módulos. Un esquema “en capas” se puede ver en la Fig. 2 y en lo que sigue se describen los distintos módulos.

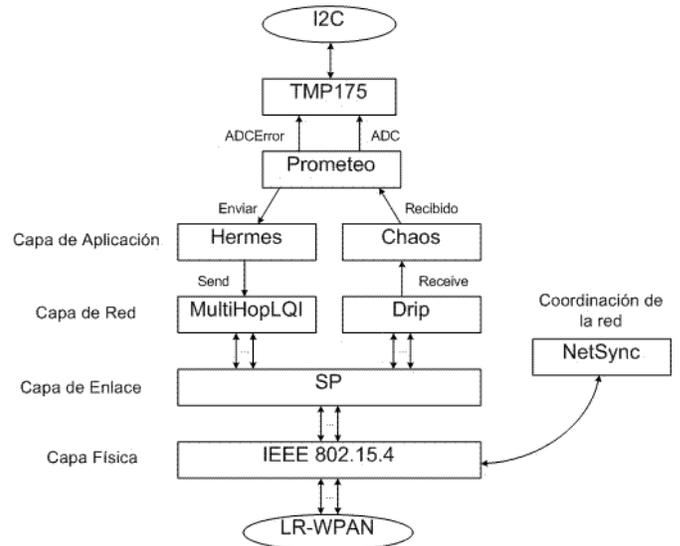


Fig. 3. Esquema en “capas de protocolos” del sistema implementado.

La programación fue realizada de forma modular, programando en NesC ([10]), sobre un sistema operativo TinyOS ([2],[3]). Los distintos módulos implementados, con sus respectivas funciones dentro del sistema, son los siguientes:

- Manejo de la interfaz I2C, comunicación con el sensor de temperatura: TMP175
- Pedido de datos, tiempos de sensado, seteo de alarmas, almacenamiento de datos, respuesta a consultas, envío de informes periódicos de datos y alarmas: PROMETEO.
- Configuración de los Motes inalámbrica: tiempo de muestreo, límites para las alarmas, modo de envío, pedido de consulta de datos almacenados: CHAOS.
- Envío de mensajes a la capa inferior, manejo diferenciado de los mensajes según sea alarma o datos: HERMES.

Todos estos módulos hacen uso de otros previamente existentes:

- MultiHopLQI : Protocolo de capa de Red que envía un dato desde un mote hasta la base. Se eligen las rutas de acuerdo con la calidad del enlace de los vecinos (Link Quality Indicator - proporcionado por la capa inferior SP). Fue modificado para esta aplicación para permitir el envío de alarmas: si el bit de alarma está prendido se setea una bandera urgente en el mensaje hacia la capa inferior (SP).
- SP (ver [8]): Este protocolo se ubica entre la capa de red y la de enlace para permitir independizar las implementaciones de capas superiores de las inferiores. Algunas de sus funciones son utilizadas por los

módulos directamente (Prioridades de mensajes, estado de enlace, calidad del enlace, etc.)

- Drip (ver [9]): Es un módulo que realiza una difusión desde la base, hacia toda la red. Es utilizado para las consultas, y reconfiguración. La diseminación usa el protocolo Trickle, y fue modificado para que utilice la capa SP.
- NetSync: Es el módulo que se encarga del apagado y sincronización de encendido de los motes para su mínimo consumo.

Tanto los componentes de hardware como la implementación y elección del software a utilizar, fueron pensados para lograr el menor consumo posible. El hardware utiliza un procesador de muy bajo consumo, con capacidad para pasar a un estado de "Standby", y la radio puede ser encendida y apagada desde el procesador. El software se ocupa de utilizar estos recursos para mantener la radio preñada sólo el tiempo necesario para el envío y recepción de paquetes, y mantener el procesador en "Standby" el mayor tiempo posible.

Como objetivo se pretende que los nodos puedan funcionar por un año con dos baterías AA de 2600mAh de rendimiento (es el rendimiento, por ejemplo, de las pilas Energizer E91 comerciales para corrientes menores a 25mA).

2) Interfaz con el usuario

El sistema requiere de un programa que se ejecute en una PC o similar, y que se encargue tanto de comunicarse de forma efectiva con los motes, como de brindar una interfaz de usuario amigable. Uno de los requisitos del proyecto era el de realizar una interfaz cuyos requerimientos en cuanto a hardware y sistema operativo para su ejecución no fueran muy restrictivos.

El programa creado es una aplicación java (Siagro2v5.jar). La pantalla principal posee un pequeño panel en el cual se registran los sucesos importantes como la inserción de nuevos motes a la red, las reconfiguraciones realizadas, las alarmas recibidas o información solicitada por el usuario. Desde esta pantalla se pueden abrir todas las otras subpantallas: "Panel de la Gráfica", "Panel de Reconfiguración", "Panel de consulta". También sirve para visualizar la topología de red de los motes y seleccionar algunos para graficar los datos que envían.

La aplicación permite ver en tiempo real y guardar los datos recibidos de los distintos motes, reconfigurar los niveles de alarma, la tasa de muestreo y el modo de envío, y realizar consultas de datos no recibidos. En las imágenes siguientes se puede ver el sistema en funcionamiento.



Fig. 4. Ventana Principal de Siagro2v5.jar.

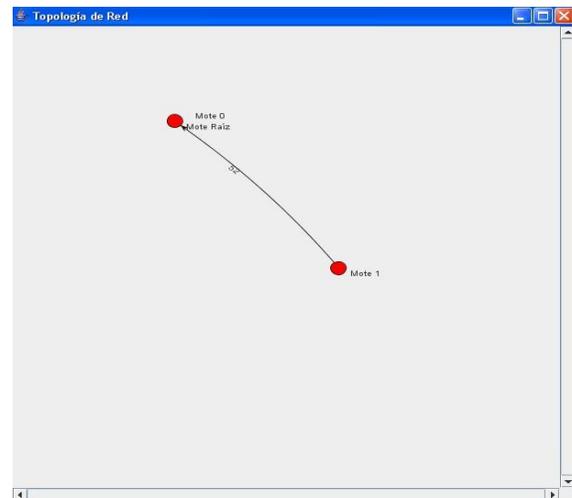


Fig. 5. Ventana que muestra la Topología de Red.

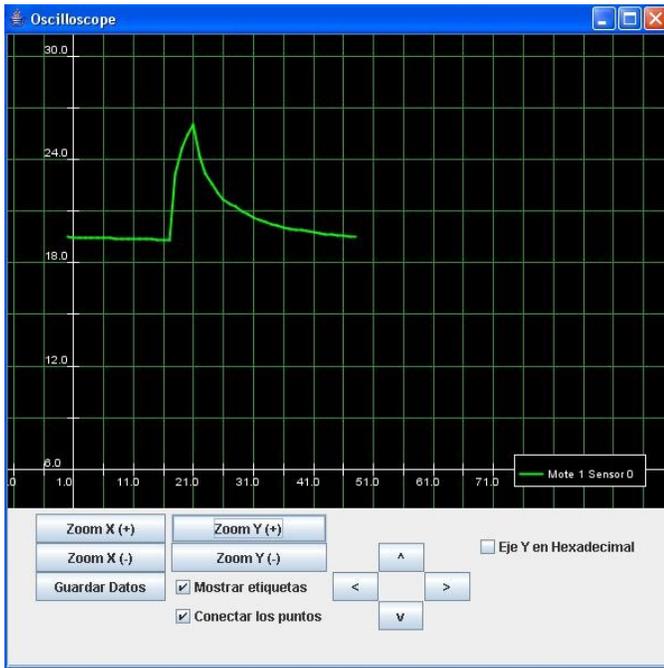


Fig. 6. Ventana que muestra la gráfica de los datos recibidos en tiempo real.

mote de forma individual durante un año. Para estimar la duración de la red primero realizamos las siguientes medidas de corriente en un mote en diferentes estados.

Las pruebas de consumo se realizaron con ayuda de una fuente externa continua de 3 volts, y una resistencia en serie de 11 Ohm, ubicada entre la entrada negativa de alimentación del mote y tierra. Los resultados fueron obtenidos con 60 segundos de periodo de muestreo, y los motes en modo lowpower (protocolo que mantiene el procesador y la radio prendidos lo mínimo indispensable) con un 5% de ciclo de trabajo:

TABLA 1
RESULTADOS DE LAS MEDIDAS DE CONSUMO

Periodo completo de la radio	2 segundos
Periodo de muestreo	1 minuto
Tiempo en que la radio está prendida	0,1 segundo
Consumo con la radio prendida	21,1 mA
Consumo de un led	4,7 mA

Quando la radio está apagada, se pueden distinguir 2 modos del microprocesador, y los consumos son los que siguen:

Con el procesador activo (5% del tiempo) = 1820 uA

Con el procesador en modo de bajo consumo (95% del tiempo) = 5,2 uA

Vemos que las mediciones de consumo coinciden con lo esperado, partiendo de la información de las hojas de datos (ver apéndices A y B). Con esos valores se estimó que la duración promedio de la red sería de 270 días (en el apéndice A se detallan los cálculos realizados).

2) Modelado del encapsulado:

Para estudiar el comportamiento del sensor dentro del tubo metálico del encapsulado se realizaron pruebas y posteriormente un ajuste de parámetros a un modelo de primer orden, con el fin de poder estimar el error introducido en la medida de la temperatura por el encapsulado. Los resultados obtenidos indican que la velocidad de respuesta es un poco menor cuando el sensor se encuentra protegido por el encapsulado, pero dentro de lo aceptable para nuestro caso de aplicación.

IV. MEDIDAS REALIZADAS

El sistema fue probado durante más de tres meses en el Centro Regional Sur, instalando nodos sensores en dos parideras -uno con dos sensores de temperatura-, un nodo base conectado a un PC con el programa de interfaz de usuario, además de dos nodos intermediarios para lograr conectividad entre la base y la zona de las parideras.

En las siguientes gráficas se observa la comparación de las medidas de temperatura de los motes, con un “datalogger”

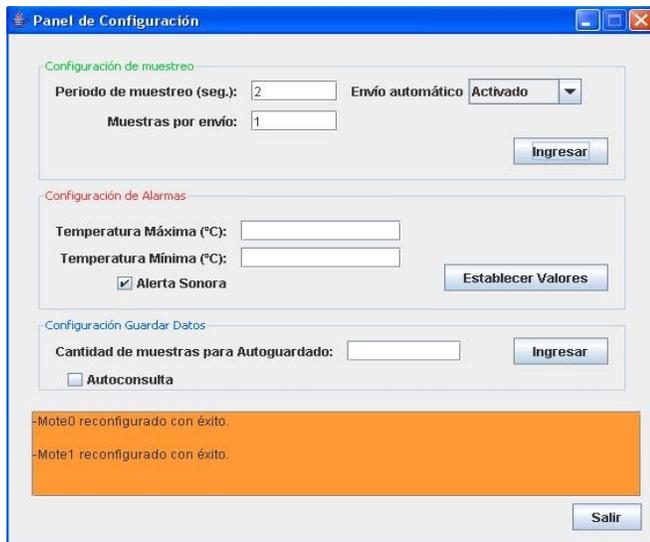


Fig. 7. Ventana de Reconfiguración.



Fig. 8. Ventana de pedido de Consultas.

C. Resultados sobre el consumo

El consumo del producto final era uno de nuestros objetivos principales. El objetivo era lograr un consumo tal que 2 pilas alcalinas AA fueran capaces de alimentar a cada

(Termocron) y usando como referencia de la zona los datos de una estación meteorológica.

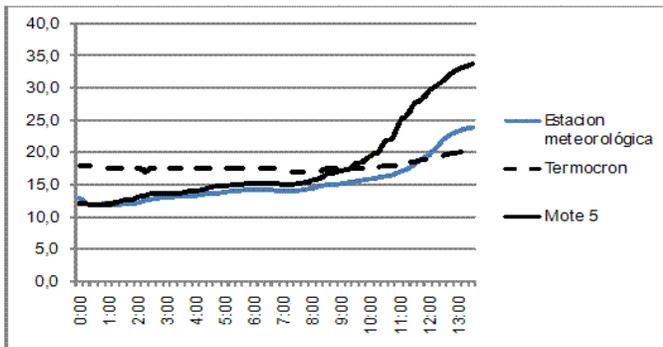


Fig. 9. Comparación de medidas de temperatura para el mote5.

Se observó que, a grandes rasgos, las medidas de los motes parecen concordar mejor con las de la estación meteorológica que las del datalogger (si bien no se han realizado estudios estadísticos).

El sensor del mote que se ve en la gráfica fue instalado en un encapsulado transparente, de material poco conductor del calor. Esto puede explicar el aumento excesivo de temperatura en las horas de mayor radiación solar, ya que se estaría generando un “efecto invernadero” dentro del encapsulado.

V. CONCLUSIONES

Se instaló una red de sensores inalámbricos para el monitoreo de temperatura en parideras de cerdos a campo abierto y se mantuvo operativa durante varios meses, alimentando cada nodo con dos pilas alcalinas tipo AA corrientes. Este tipo de tecnología permite implantación de sistemas de medida para la obtención de información en tiempo real y sin la necesidad de la intervención de un operario para ello. Esta experiencia abre nuevas oportunidades en la aplicación de las redes de sensores inalámbricos en áreas todavía inexploradas.

VI. APÉNDICES

A. Cálculo teórico del consumo

Netsync ofrece la capacidad de apagar y prender la radio, sincronizadamente con el resto de los motes. Para el cálculo del consumo, se requieren dos parámetros de Netsync:

- Periodo completo (ON-OFF) de la radio
- Ciclo de trabajo

El periodo de apagado y prendido es distinto al periodo de muestreo, y es necesariamente menor a éste ya que se envían además de los datos, otros paquetes que cumplen tareas de sincronización, confirmación de recepción, etc.

La vida útil de las baterías se verá muy afectada por el ciclo de trabajo que se elija (porcentaje del tiempo que está prendida la radio) pero no por el periodo completo de

funcionamiento de la radio. Otro factor a tener en cuenta es la topología de la red, de la que haremos suposiciones para considerar el peor caso.

Para hacer las cuentas, suponemos los siguientes valores:

- Factor de seguridad en la red de 0,9, dejando el resto para paquetes perdidos (FactorSeguridad)
- El tiempo entre beacons de envío a los vecinos es de 15 segundos (TBEACON)
- El tiempo de muestreo de datos es de 10 minutos (TMUESTREO)
- El número de vecinos es de 8 por mote (NV ECINOS)
- El número de motes total es de 100 (NMOTES)
- La corriente consumida en estado “despierto” es de 20mA (ION)
- La corriente consumida en estado “dormido” es de 10 μ A (IOFF)
- La capacidad de las pilas alcalinas es de 2600 mAh (CPILA)
- La capacidad del canal es de 100msg/s, lo que se deduce de la velocidad de transmisión (250 kbps) y un mensaje promedio de 31Bytes, teniendo en cuenta que aproximadamente un 90% del tráfico es debido a mensajes de control y sincronización de la red.

Para conocer el ciclo de trabajo mínimo, primero se debe proceder a calcular el flujo de paquetes que pasan por un mote de la red. Consideramos que el peor caso en la topología de la red es aquella en la que un nodo o una cadena de nodos sin ciclos es la única comunicación entre el resto de la red y el nodo base. Se considerarán solo los paquetes que contienen muestras y los beacons a los vecinos, tratando a otros posibles envíos y recepciones como eventos extraordinarios. Por lo tanto, calculamos el flujo de mensajes por el nodo más sobrecargado:

- El nodo en cuestión debe recibir las muestras de cada nodo de la red.
- Debe confirmar la recepción de dichos mensajes.
- Debe retransmitirlos a la base.
- Debe recibir confirmación de dichas retransmisiones.
- Debe recibir los beacons de sus vecinos.
- Debe enviar sus propios beacons (a la dirección de broadcast).

El flujo de mensajes por el nodo es de:

$$\text{FlujoMensajes} = 4 \frac{N_{\text{MOTES}}}{T_{\text{MUESTREO}}} + \frac{1 + N_{\text{VECINOS}}}{T_{\text{BEACON}}} =$$

$$4 \frac{100}{10 * 60s} + \frac{1 + 8}{15s} = 1,27 \text{msg/s}$$

Luego, el ciclo de trabajo necesario queda:

$$\text{CicloTrabajo} = \frac{\text{FlujoMensajes}}{\text{CapacidadCanal}} \frac{1}{\text{FactorSeguridad}}$$

$$= \frac{1,27\text{msgs/s}}{100\text{msgs/s}} \frac{1}{0,9} = 0,0141 = 1,4\%$$

Tomando un ciclo de trabajo de 2%, el consumo quedaría:

$$I_{\text{PROMEDIO}} = I_{\text{ON}} \cdot \text{CicloTrabajo} + I_{\text{OFF}} \cdot (1 - \text{CicloTrabajo})$$

$$= 20\text{mA}(0,02) + 0,01\text{mA}(1 - 0,02) = 0,4\text{mA}$$

Y el tiempo de vida:

$$t_{\text{vida}} = \frac{C_{\text{pila}}}{I_{\text{PROMEDIO}}} = \frac{2600\text{mAh}}{(0,4\text{mA})(24\text{h/día})} = \frac{2600\text{mAh}}{9,6\text{mAh/día}} = 270\text{días}$$

Recordamos que éste es el peor caso en que todos los paquetes pasan por un solo nodo. Este consumo se podría disminuir modificando la topología: por ejemplo, poner 2 nodos en vez de uno entre el nodo base y la red, para dividir el tráfico de mensajes.

Utilizando el mínimo ciclo de trabajo aceptado por NetSync (1%) el tiempo de vida sería de 516 días (disminuyendo el periodo de muestreo, esto sería posible).

B. Algunos Datos del Hardware Utilizado

Microprocesador MSP430

El bajo consumo del mote está dado principalmente por el bajo consumo del microprocesador. Este procesador tiene la principal característica de estar activo o durmiendo, consumiendo una corriente extremadamente baja. Posee 10kB de RAM y 48 kB de memoria flash.

El MSP430 tiene un oscilador interno controlado digitalmente (DCO), que opera por encima de 8 MHz. El DCO puede encenderse partiendo del estado dormido en tan sólo 6µs. Cuando el DCO está apagado, opera un oscilador externo a cristal de 32.768Hz. Cuenta con puertos SPI, UART, puertos digitales I/O, Timers y Watchdog timers, entre otros.

Algunas características más se pueden ver en la siguiente tabla:

TABLA 2
CARACTERÍSTICAS DEL MSP430

Condiciones de operación típicas	Min	Nom	Max	Unidades
Alimentación durante la ejecución de un programa	1.8	-	3.6	V
Alimentación durante la programación de memoria flash	2.7	-	3.6	V
Temperatura de operación al aire libre	-40	-	85	°C
Cristal de frecuencia		32.3768		kHz
Corriente activa a $V_{cc} = 3V$ y 1 MHz		500	600	µA
Corriente de sleep en LPM3, $V_{cc} = 3V$ y 32.768 kHz		2.6	3.0	µA
modo Low Power			6	µA

VII. AGRADECIMIENTOS

Los autores agradecen a los integrantes del Instituto de Ingeniería Eléctrica (Fing-UdelaR), que siempre están dispuestos a colaborar.

VIII. REFERENCIAS

Artículos Técnicos:

- [1] Tmote Sky – Moteiv Corporation. “Ultra low power IEEE 802.15.4 compliant wireless sensor module.” Junio de 2006.
- [2] Philip Levis. TinyOS Programming. (<http://csl.stanford.edu/~pal/pubs/tinyos-programming.pdf>)
- [3] TinyOS Community. TinyOS Documentation. (<http://www.tinyos.net/tinyos-1.x/doc>)
- [4] TMP175 Datasheet. Texas Instrument Digital Temperature Sensor with Two-Wire Interface.

Artículos Presentados en Conferencias (Publicados):

- [5] Antonio Vadell. Producción de cerdos a campo en un sistema de mínimos costos. Facultad de Agronomía – Universidad de la República (R.O. del Uruguay).
- [6] Nicolás Piriz, Sebastián Monzón, Diego Méndez. Sistema Registrador de Temperatura para Aplicaciones Agronómicas. Facultad de Ingeniería – Universidad de la República (R.O. del Uruguay)
- [7] J.Hill, D. Culler. A wireless Embedded Sensor Architecture for System-Level Optimization. 2001.
- [8] Joseph Polastre, Jonathan Hui, Philip Levis, Jerry Zhao, David Culler, Scott Shenker and Ion Stoica. A Unifying Link Abstraction for Wireless Sensor Networks. (<http://www.polastre.com/papers/sensys05-sp.pdf>)
- [9] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. (<http://csl.stanford.edu/~pal/pubs/trickle-nsdi04.pdf>)
- [10] David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, and David Culler. The NesC Language: A Holistic Approach to Network Embedded Systems. (<http://www.cs.berkeley.edu/~kwright/nest.papers/nesc.pdf>)

Standards:

- [11] IEEE 802.15.4. Introduction to IEEE 802.15.4 Technology. Marzo de 2004.