

UNIVERSIDAD POLITÉCNICA DE VALENCIA

DEPARTAMENTO DE ELECTRÓNICA



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# Modelado y optimización de energía en redes de sensores inalámbricas para la medida de parámetros medioambientales

---

**TESIS DOCTORAL**

**Autor:**

José Rafael Lajara Vizcaíno

**Directores:**

José Pelegrí Sebastia  
Juan José Pérez Solano

Junio, 2014



# ABSTRACT

---

Wireless sensor networks are a recent topic of research. They are formed by a set of nodes that perform a particular task. The nodes often are small electronic devices, autonomous, battery powered and they are able to communicate each with others wirelessly.

Due to the characteristics of size and battery powered, energy consumption is a key factor in the design. The need of optimizing power consumption originates new research topics as energy harvesting and consumption optimization. This thesis is content within these fields, studying, proposing and implementing solutions.

In the first part, the behavior, the most commonly used architectures devices and operating systems in the field of sensor networks are studied. The analysis will focus on the TinyOS, Contiki and MantisOS operating systems and the Tmote Sky and MICAz devices.

In the next part of the art of theoretical models for energy consumption in sensor networks will be studied from different perspectives: the transceiver, an entire node, a network, etc. Then, it is proposed a methodology to obtain models that improve knowledge on state of charge of a sensor node, taking into account factors such as temperature or battery wear. Applying this method, it will be proposed several models based on linear regressions and neural networks which can be executed by an end node. The results are validated with experimental measurements and compare with other hardware devices.

A power supply architecture based on solar energy harvesting will be proposed. Besides, this source will reduce wear on the rechargeable batteries by using supercapacitors. This power supply automatically switches between the two sources and it prioritizes the supercapacitor respect to the battery. The design will operate at a typical node exclusively from a supercapacitor for several days; the battery operation is only required when weather conditions prevent getting enough energy from the sun. After that, the possibility of using other sources of energy harvesting will be studied: energy from commercial radio waves. Several circuits will be studied, and their results will be compared. This method provides a little current, but it may be enough for a node with an extraordinarily low consumption, or to support other energy source, especially considering that it is always available and is not dependent on weather conditions.

In the last part of the thesis, several applications are implemented. Firstly, a wireless node will be implemented to control irrigation systems by using electrovalves. The node will have a power supply and trigger system combining capacitors and supercapacitors. Furthermore, a medium access protocol is implemented, this protocol maintains synchronism between adjacent nodes by a hardware system that reduces power consumption without losing the timing. The second application is a system that measures environmental parameters. This

system uses the solar power source designed previously. Besides the consumption of this node energy is very low, this consumption approaches to the energy that the radio energy harvesting system can provide. The environmental parameters can be accessed on Internet.

# RESUMEN

---

Las redes de sensores inalámbricos forman un reciente campo de investigación. Están formadas por una serie de nodos que realizan una determinada tarea. Los nodos suelen ser pequeños dispositivos electrónicos, autónomos, alimentados por baterías y con capacidad para comunicarse entre ellos inalámbricamente.

Las características del tamaño y la alimentación con batería hacen que el consumo de energía sea un factor clave en su diseño. A partir de la necesidad de optimizar el consumo de energía aparecen nuevos tópicos de investigación como la recolección de energía y la optimización del consumo. Esta tesis se enmarca dentro de estos campos, tratando de estudiar, proponer soluciones e implementarlas.

En la primera parte se estudiará el comportamiento y arquitecturas de los dispositivos y sistemas operativos más utilizados en el ámbito de las redes de sensores. El análisis se enfocará en los sistemas operativos TinyOS, MantisOS y Contiki y en los dispositivos Tmote Sky y MICAz.

En la siguiente parte se estudiará el estado del arte de los modelos teóricos sobre el consumo de energía en redes de sensores desde diferentes perspectivas: el *transceiver*, un nodo completo, toda una red, etc. Después se propondrá una metodología para obtener modelos para mejorar el conocimiento sobre estado de carga de un nodo sensor, teniendo en cuenta factores tales como la temperatura o el desgaste de las baterías. Aplicando este método se propondrán varios modelos basados en regresiones lineales y redes neuronales que puedan ser ejecutados por un nodo final. Los resultados se validarán con medidas experimentales y comparativas con otros dispositivos hardware.

Se propondrá una arquitectura de fuente de alimentación basada en recolección de energía solar. Además esta fuente permitirá reducir el desgaste de las baterías recargables mediante el empleo de supercondensadores. Para ello la fuente de alimentación cuenta con un sistema que automáticamente conmuta entre ambas fuentes y prioriza la del supercondensador respecto a la de la batería. El diseño permitirá operar a un nodo típico exclusivamente desde un supercondensador durante varios días, entrando la batería en funcionamiento únicamente cuando las condiciones climatológicas impiden obtener la suficiente energía del sol. Después se estudiará la posibilidad de utilizar otra fuente de recolección de energía: la energía proveniente de ondas de radio comerciales. Para ello se estudiarán diversos circuitos y se compararán sus resultados. Este método de obtención de energía, si bien proporciona poca corriente, puede ser suficiente para un nodo con un consumo extraordinariamente reducido, o como apoyo a otra fuente de energía, sobre todo teniendo en cuenta que su disponibilidad no depende de condiciones climatológicas.

En la última parte de la tesis se realizarán varias aplicaciones. En primer lugar se implementará un nodo inalámbrico para controlar sistemas de regadíos mediante electroválvulas. El nodo tendrá un sistema de alimentación y disparo de las electroválvulas combinando condensadores y supercondensadores. Además se implementará un protocolo de acceso al medio que mantiene el sincronismo entre nodos adyacentes mediante un sistema hardware que permite reducir el consumo del nodo sin perder la temporización. La segunda aplicación será un sistema medidor de parámetros medioambientales que utilizará la fuente de alimentación diseñada anteriormente. Además el consumo de este nodo se aproxima a la energía que podría proporcionar un sistema de recolección mediante ondas de radio. Se podrá acceder a los parámetros medioambientales recogidos a través de internet.

# RESUM

---

Les xarxes de sensors sense fils formen un recent camp de recerca. Estan formades per una sèrie de nodes que realitzen una determinada tasca. Els nodes solen ser petits dispositius electrònics, autònoms, alimentats per bateries i amb capacitat per a comunicar-se entre ells sense fils.

Les característiques de la grandària i l'alimentació amb bateria fan que el consum d'energia siga un factor clau en el seu disseny. A partir de la necessitat d'optimitzar el consum d'energia apareixen nous tòpics de recerca com la recol·lecció d'energia i l'optimització del consum. Aquesta tesi s'emmarca dins d'aquests camps, tractant d'estudiar, proposar solucions i implementar-les.

En la primera part s'estudiarà el comportament i arquitectures dels dispositius i sistemes operatius més utilitzats en l'àmbit de les xarxes de sensors. En anàlisi s'enfocarà en els sistemes operatius TinyOS, MantisOS i Contiki i en els dispositius Tmote Sky i MICAz.

En la següent part s'estudiarà l'estat de l'art dels models teòrics sobre el consum d'energia en xarxes de sensors des de diferents perspectives: el *transceiver*, un node complet, tota una xarxa, etc. Després es proposarà una metodologia per a obtenir models per a millorar el coneixement sobre estat de càrrega d'un node sensor, tenint en compte factors tals com la temperatura o el desgast de les bateries. Aplicant aquest mètode es proposaran diversos models basats en regressions lineals i xarxes neuronals que puguen ser executats per un node final. Els resultats es validaran amb mesures experimentals i comparatives amb altres dispositius maquinari.

Es proposarà una arquitectura de font d'alimentació basada en recol·lecció d'energia solar. A més aquesta font permetrà reduir el desgast de les bateries recarregables mitjançant l'ocupació de supercondensadors. Per a açò la font d'alimentació compta amb un sistema que automàticament commuta entre ambdues fonts i prioritza la del supercondensadors respecte a la de la bateria. El disseny permetrà operar a un node típic exclusivament des d'un supercondensador durant diversos dies, entrant la bateria en funcionament únicament quan les condicions climatològiques impedeixen obtenir la suficient energia del sol. Després s'estudiarà la possibilitat d'utilitzar una altra font de recol·lecció d'energia: l'energia provinent d'ones de ràdio comercials. Per a açò s'estudiaran diversos circuits i es compararan els seus resultats. Aquest mètode d'obtenció d'energia, si bé proporciona poc corrent, pot ser suficient per a un node amb un consum extraordinàriament reduït, o com a suport a una altra font d'energia, sobretot tenint en compte que la seua disponibilitat no depèn de condicions climatològiques.

En l'última part de la tesi es realitzaran diverses aplicacions. En primer lloc s'implementarà un node sense fil per a controlar sistemes de regadius mitjançant electrovàlvules. El node tindrà un sistema d'alimentació i tret de les electrovàlvules combinant condensadors i supercondensadors. A més s'implementarà un protocol d'accés al medi que manté el sincronisme entre nodes adjacents mitjançant un sistema maquinari que permet reduir el consum del node sense perdre la temporització. La segona aplicació serà un sistema mesurador de paràmetres mediambientals que utilitzarà la font d'alimentació dissenyada anteriorment. A més el consum d'aquest node s'aproxima a l'energia que podria proporcionar un sistema de recollida mitjançant ones de ràdio. Es podrà accedir als paràmetres mediambientals arreplegats a través d'internet.

# AGRADECIMIENTOS

---

En primer lugar quiero agradecer a los tutores de esta tesis porque sin éste, no hubiera sido posible. En primer lugar por la insistencia en que la continuara después de haber dejado la universidad y durante tanto tiempo; y en segundo lugar por las facilidades y apoyo para realizarla a pesar de las condiciones en las que se ha hecho.

También tengo que reconocer las aportaciones y agradecer la ayuda a los miembros del grupo de investigación GSYM que han participado en trabajos que han derivado en esta tesis, y por las consejos y comentarios sobre la misma.

Además también debo agradecer a los revisores de esta tesis por sus aportaciones y sugerencias siempre constructivas que han ayudado a mejorar el resultado final.

Finalmente, a mi familia y amigos por estar siempre ahí.



# ACRÓNIMOS

---

ACK:	Acknowledgment	MCFA:	Minimum Cost Forwarding Algorithm
ADC:	Analog-to-Digital Converter	MLP:	MultiLayer Perceptron
AM:	Amplitude Modulation	MOS:	Mantis Operating System
AMI:	Alternate Mark Inversion	MOSFET:	Metal-Oxide-Semiconductor Field-Effect Transistor
AODV:	Ad-Hoc On Demand Vector	MPP:	Maximum Power Point
CSMA:	Carrier Sense Multiple Access	NRZ:	Non-Return Zero
CTP:	Collection Tree Protocol	P2P:	Peer to Peer
CTS:	Cleat To Send	PMU:	Power Management Unit
DAQ:	Data Acquisition	QPSK:	Quadrature Phase-Shift Keying
DMA:	Direct Memory Access	RFID:	Radio-Frequency Identification
DoD:	Deep of Discharge	RH:	Relative Humidity
DSSS:	Direct-Sequence Spread Spectrum	RMS:	Root Mean Square
FTDI:	Future Technology Devices International	RTS:	Request To Send
I <sup>2</sup> C:	Inter-Integrated Circuit	SoC:	State of Charge
IEEE:	Institute of Electrical and Electronics Engineers	SPI:	Serial Peripheral Interface bus
IIR:	Infinite Impulse Response	SPICE:	Simulation Program with Integrated Circuit Emphasis
ISM:	Industrial, Scientific and Medical band	TDMA:	Time Division Multiple Access
JTAG:	Joint Test Action Group	UART:	Universal Asynchronous Receiver/Transmitter
LED:	Light-Emitting Diode	UHF:	Ultra High Frequency
LiPo:	Lithium-Polymer	WSN:	Wireless Sensor Network



# ÍNDICE DE FIGURAS

FIGURA 1-1: DIAGRAMA DE BLOQUES GENÉRICO DE UN NODO FINAL DE REDES DE SENSORES INALÁMBRICAS. ....	1
FIGURA 2-1: DISTINTOS TIPOS DE TOPOLOGÍAS UTILIZADOS EN REDES DE SENSORES.....	9
FIGURA 2-2: PILA DE PROTOCOLOS DEL MODELO OSI (IZQUIERDA) Y DIFERENTES IMPLEMENTACIONES EN SISTEMAS DE COMUNICACIÓN INALÁMBRICOS DE ALTA FRECUENCIA. ....	10
FIGURA 2-3: ESQUEMA DE TIEMPOS EN LOS QUE SE ENVIARÁN LAS DISTINTAS TRAMAS EN EL PROTOCOLO SMAC. ....	18
FIGURA 2-4: ESQUEMA DE TIEMPOS DEL PROTOCOLO BMAC. ....	18
FIGURA 3-1: CONSUMO INSTANTÁNEO DE CORRIENTE PARA CADA APLICACIÓN, SISTEMA OPERATIVO Y MOTE. ....	39
FIGURA 3-2: CONSUMO DE CORRIENTE MEDIO. PRIMERA FILA BLANK; SEGUNDA FILA BLANK2 (OPTIMIZACIÓN DE CONSUMO); TERCERA FILA BLINK; CUARTA FILA SENS; QUINTA FILA SENS2 (OPTIMIZACIÓN DE CONSUMO).....	40
FIGURA 3-3: MEDIDA DE RUIDO EN LA LÍNEA DE ALIMENTACIÓN DE TMOTE SKY Y MICAZ CON MOS. ....	42
FIGURA 4-1: ESQUEMA DE MEDICIÓN DE LA TENSIÓN Y CORRIENTE DE LA BATERÍA.....	52
FIGURA 4-2: CONSUMO DE CORRIENTE DE LA TMOTE SKY. RESISTENCIA SHUNT DE 32 Ω; ABAJO: DETALLE DE LA GRÁFICA. ...	53
FIGURA 4-3: MÁQUINA DE ESTADOS PARA MONITORIZAR EL ESTADO DE LA BATERÍA. ....	54
FIGURA 4-4: SISTEMA DE DESGASTE AUTOMÁTICO DE BATERÍAS. ....	55
FIGURA 4-5: EVOLUCIÓN DE LA TENSIÓN DE UNA BATERÍA EN EL BANCO DE STRESS.....	56
FIGURA 4-6: EVOLUCIÓN DE LAS HORAS DE AUTONOMÍA RESPECTO AL NÚMERO DE CICLOS EN LAS BATERÍAS. ....	57
FIGURA 4-7: CURVAS DE LA TENSIÓN DE DESCARGA DE LAS MEDIDAS. ....	58
FIGURA 4-8: CURVA DE TENSIÓN VS SOC IDEAL. ....	59
FIGURA 4-9: EFECTO DEL SOH DE LA BATERÍA SOBRE LOS RESULTADOS DE LOS MODELOS.....	63
FIGURA 4-10: COMPARATIVA DEL AJUSTE ENTRE LA ESTIMACIÓN DE LOS MODELOS RESPECTO AL VALOR REAL DURANTE UNA SESIÓN COMPLETA DE FUNCIONAMIENTO. BATERÍA CON 1 CICLO DE DESGASTE Y TEMPERATURA A 25 °C. ....	64
FIGURA 5-1: DIAGRAMA DE BLOQUES DE LA ARQUITECTURA DEL SISTEMA. ....	74
FIGURA 5-2: MODELOS DE CONSUMO DE UN NODO SENSOR TÍPICO A DIFERENTES CICLOS DE TRABAJO. ....	76
FIGURA 5-3: ESQUEMA ELECTRÓNICO DE LA FUENTE DE ALIMENTACIÓN Y SUS NIVELES DE TENSIÓN. ....	79
FIGURA 5-4: FOTOGRAFÍA DEL PROTOTIPO DE LA FUENTE DE ALIMENTACIÓN CON RESISTENCIAS DE CARGA PARA SIMULAR EL CONSUMO DE UN NODO SENSOR. ....	79
FIGURA 5-5: CURVAS DE CORRIENTE Y POTENCIA PARA EL PANEL SOLAR SELECCIONADO. CURVA ROJA: CORRIENTE Y TENSIÓN CON BUENAS CONDICIONES AMBIENTALES; CURVA AZUL: CORRIENTE Y TENSIÓN UN DÍA NUBLADO; CURVA VERDE: POTENCIA ESCALADA POR 10 CON BUENAS CONDICIONES AMBIENTALES; CURVA MORADA: POTENCIA ESCALADA POR 10 UN DÍA NUBLADO. ....	80
FIGURA 5-6: EVOLUCIÓN DE LOS NIVELES DE TENSIÓN EN LOS PUNTOS MUESTREADOS. ....	84
FIGURA 5-7: RECUPERACIÓN DE LA BATERÍA DE LITIO DESPUÉS DE UNA DESCARGA PROFUNDA INTENCIONADA. ....	85
FIGURA 5-8: DESCARGA DE LOS SUPERCONDENSADORES DURANTE 4 NOCHES SUPERPUESTAS A DIFERENTES CARGAS DE TRABAJO. ....	85
FIGURA 6-1: ESQUEMA CONCEPTUAL DEL SISTEMA DE RECOLECCIÓN DE ENERGÍA SOBRE UN NODO SENSOR.....	92
FIGURA 6-2: DIAGRAMA DE BLOQUES DE UNA RECTENA.....	92
FIGURA 6-3: SIMULACIÓN DE UNA ANTENA EN L DE LAS DIMENSIONES DADAS PARA 1,584 MHZ. ARRIBA: GEOMETRÍA DE LA ANTENA; ABAJO: DIAGRAMA DE RADIACIÓN.....	93
FIGURA 6-4: DOBLADORES VILLARD (IZQUIERDA) Y DICKSON (DERECHA); Y COMPARATIVA DE SUS SALIDAS (GRÁFICA INFERIOR), DONDE EN VERDE SE VE LA ENTRADA, EN AZUL LA SALIDA DEL CIRCUITO VILLARD Y EN ROJO LA DEL DICKSON. ....	94

FIGURA 6-5: ESQUEMÁTICOS DE LAS DIFERENTES TOPOLOGÍAS DE CIRCUITOS. DESDE ARRIBA HACIA ABAJO: (1) VILLARD; (2) DICKSON; (3) VILLARD RESONANTE; (4) DICKSON RESONANTE. ....	97
FIGURA 6-6: RESPUESTA DE LAS CUATRO TOPOLOGÍAS DE CIRCUITOS MOSTRANDO LA TENSIÓN EN EL CONDENSADOR DE SALIDA. ....	98
FIGURA 6-7: RESULTADO EXPERIMENTAL. TENSIÓN DE SALIDA VS FRECUENCIA PARA EL CIRCUITO RESONANTE VILLARD. ....	99
FIGURA 6-8: RESULTADO EXPERIMENTAL. TENSIÓN DE SALIDA VS FRECUENCIA PARA EL CIRCUITO RESONANTE DICKSON. ....	99
FIGURA 6-9: RESULTADOS EXPERIMENTALES. POTENCIA VS FRECUENCIA EN DIFERENTES PUNTOS DEL SISTEMA. DESDE ARRIBA HACIA ABAJO: (1) POTENCIA SIN ANTENA; (2) POTENCIA CON ANTENA DE TIPO L; (3) POTENCIA CON CIRCUITO DE SINTONIZACIÓN. ....	100
FIGURA 7-1: DIAGRAMA DE BLOQUES DEL NODO TERMINAL.....	105
FIGURA 7-2: CRISTAL Y REGISTRO COMO DIVISOR DE FRECUENCIA PARA GENERAR SEÑAL QUE DESPIERTA AL MICROCONTROLADOR. ....	108
FIGURA 7-3: ESQUEMA <i>LISTEN/SLEEP</i> CON DOS NODOS UTILIZANDO SMAC. ....	109
FIGURA 7-4: ARQUITECTURA DE LA RED. ....	110
FIGURA 7-5: CONSUMO DE CORRIENTE DE UN NODO TERMINAL. LOS PICOS DE 27 mA SON PRODUCIDOS CADA 4 S DEBIDO A LA RESINCRONIZACIÓN. EL CONSUMO MEDIO ES DE 836,76 $\mu$ A. ....	110
FIGURA 7-6: CURVA DE DESCARGA DEL SUPERCONDENSADOR PRINCIPAL. LA AUTONOMÍA ES DE APROXIMADAMENTE 26 HORAS SIN APORTE DEL PANEL SOLAR CON UN TIEMPO DE SINCRONIZACIÓN DE 4 SEGUNDOS. CON UN TIEMPO DE 8 SEGUNDOS LA AUTONOMÍA SE EXTIENDE A 50 HORAS.....	111
FIGURA 7-7: DESCARGA DEL SUPERCONDENSADOR PRINCIPAL, EN VERDE, DURANTE UNA SECUENCIA ABRIR-CERRAR-ABRIR-CERRAR LA ELECTROVÁLVULA DE UN NODO. SEÑAL DE DISPARO DE LA ELECTROVÁLVULA EN AZUL Y TENSIÓN DEL PANEL SOLAR EN LOS EXTREMOS DE UNA RESISTENCIA SHUNT (GRIS Y ROJO). LA TENSIÓN EN EL SUPERCONDENSADOR NO SE VE AFECTADA POR LOS DISPAROS. ....	111
FIGURA 7-8: ABERTURA DE LA ELECTROVÁLVULA (SUPERIOR): UN PULSO DE CORTA DURACIÓN (MICROSEGUNDOS) INICIA LA CARGA DEL SUPERCONDENSADOR DE 3,3 MF HASTA ALCANZAR EL UMBRAL DE DISPARO, CUANDO SE ALCANZA, SE DISPARA INTERRUPTOR DE ABERTURA DURANTE 15 MS. CIERRE DE LA ELECTROVÁLVULA (INFERIOR): SE CARGA EL SUPERCONDENSADOR HASTA EL UMBRAL Y SE PROVOCA UN PULSO DE DISPARO DE 20 MS. ....	112
FIGURA 7-9: CANAL 1: CORRIENTE PARA CONMUTAR LA ELECTROVÁLVULA. CANAL 2: CAÍDA DE VOLTAJE EN EL SUPERCONDENSADOR DE 3,3 MF. GRÁFICA SUPERIOR: ABERTURA, GRÁFICA INFERIOR: CIERRE. ....	112
FIGURA 8-1: ESQUEMA GENERAL DEL SISTEMA.....	116
FIGURA 8-2: DIAGRAMA DE BLOQUES DEL HARDWARE DEL NODO QUE IMPLEMENTA LA ESTACIÓN METEOROLÓGICA INALÁMBRICA.....	116
FIGURA 8-3: PLANIFICACIÓN DE LA LECTURA DE LOS SENSORES EN EL NODO SENSOR. EL ORDEN OPTIMIZA EL CONSUMO ENERGÉTICO. ....	119
FIGURA 8-4: INTERFAZ GRÁFICO QUE MUESTRA LA VARIACIÓN DE LAS VARIABLES METEOROLÓGICAS. ES EL PRIMERO DE LOS MÉTODOS DE ACCESO A LOS DATOS.....	121
FIGURA 8-5: CONSUMO DE CORRIENTE EN UN NODO SENSOR DE LA ESTACIÓN METEOROLÓGICA INALÁMBRICA ALIMENTADO A 3,3 V. EL MICROCONTROLADOR SE DESPIERTA CADA SEGUNDO Y OBTIENE UNA MUESTRA DE CADA SENSOR. DESPUÉS DE 10 MUESTRAS SE ENVÍA UNA TRAMA DE DATOS.....	122
FIGURA 8-6: FOTOGRAFÍA DEL NODO SENSOR. SE MUESTRA LA LOCALIZACIÓN DE LOS CUATRO SENSORES, EL MICROCONTROLADOR Y EL <i>TRANSCEIVER</i> . EL CONECTOR DE PROGRAMACIÓN ESTÁ EN LA PARTE INFERIOR.....	123
FIGURA 8-7: FOTOGRAFÍA DEL NODO SENSOR (IZQUIERDA) CONECTADO A LA FUENTE DE ALIMENTACIÓN DEL CAPÍTULO 5 (DERECHA), AL PANEL SOLAR (ARRIBA) Y DENTRO DE UNA CARCASA PROTECTORA.....	123

# ÍNDICE DE TABLAS

---

TABLA 2-1: COMPARATIVA ENTRE MOTES. ....	12
TABLA 3-1: SISTEMAS OPERATIVOS PARA REDES DE SENSORES INALÁMBRICAS. ....	27
TABLA 3-2: CARACTERÍSTICAS PRINCIPALES DE TMOTE SKY Y MICAZ. ....	32
TABLA 3-3: TAMAÑO EN MEMORIA DE LOS PROGRAMAS EN BYTES. ....	36
TABLA 3-4: VARIANZA DE LAS MUESTRAS DE LAS MEDIDAS DE CONSUMO INSTANTÁNEO ( $A^2$ ). ....	41
TABLA 4-1: RESUMEN DE AUTONOMÍA (HORAS) DE LA MEDIDAS PARA CADA DESGASTE Y TEMPERATURA. MEDIA $\pm$ DESVIACIÓN ESTÁNDAR (ERROR TÍPICO). ....	57
TABLA 4-2: RESULTADOS DE LOS MODELOS PROPUESTOS SOBRE LAS MEDIDAS ORIGINALES. ....	62
TABLA 4-3: RESULTADOS DE LOS MODELOS PROPUESTOS SOBRE LAS MEDIDAS NO CONSIDERADAS EN LA GENERACIÓN. ....	65
TABLA 4-4: COSTE COMPUTACIONAL Y ENERGÉTICO DE LOS MODELOS PROPUESTOS. ....	67
TABLA 6-1: RESUMEN DE LOS RESULTADOS EXPERIMENTALES EN LOS CIRCUITOS RESONANTES. ....	101
TABLA 8-1: CARACTERÍSTICAS PRINCIPALES DE LOS SENSORES UTILIZADOS. ....	118



# ÍNDICE DE CONTENIDO

---

Abstract .....	i
Resumen.....	iii
Resum.....	v
Agradecimientos .....	vii
Acrónimos .....	ix
Índice de figuras .....	xi
Índice de tablas .....	xiii
Índice de contenido.....	xv
1. Introducción .....	1
1.1 Problemática a tratar .....	2
1.2 Objetivos .....	2
1.3 Estructura de la tesis .....	3
1.4 Bibliografía .....	4
2. Estado del arte .....	5
2.1 Redes de sensores.....	5
2.2 Aplicaciones.....	6
2.2.1 Medio ambiente.....	6
2.2.2 Domótica y salud .....	6
2.2.3 Industriales y vehículos .....	7
2.2.4 Eficiencia energética .....	8
2.2.5 Aplicaciones militares y entornos de alta seguridad.....	8
2.3 Características de las redes de sensores.....	8
2.3.1 Topología de la red y comunicaciones .....	8
2.3.2 Motes .....	11
2.3.3 Sistemas operativos .....	13
2.4 Gestión de energía en redes de sensores .....	14
2.4.1 Recolección de energía .....	14
2.4.2 Hardware.....	15

2.4.3	Firmware .....	16
2.4.4	Protocolos .....	17
2.5	Conclusión .....	19
2.6	Bibliografía .....	20
3.	Análisis de consumo de energía en WSN .....	25
3.1	Introducción .....	25
3.2	Estado del arte .....	26
3.3	Sistemas operativos .....	28
3.3.1	TinyOS version 1.0 (T1).....	28
3.3.2	TinyOS version 2.0 (T2).....	30
3.3.3	Contiki .....	30
3.3.4	Mantis OS (MOS) .....	30
3.4	Plataformas .....	31
3.4.1	Tmote Sky .....	33
3.4.2	MICAz .....	33
3.5	Aplicaciones.....	33
3.6	Metodología y materiales .....	35
3.7	Resultados .....	35
3.7.1	Resultados y discusión de las medidas de consumo .....	36
3.7.2	Resultados y discusión de las medidas de ruido .....	41
3.8	Conclusiones.....	43
3.9	Bibliografía .....	44
4.	Modelado .....	47
4.1	Introducción .....	47
4.2	Estado del arte .....	48
4.3	Metodología .....	50
4.4	Medidas.....	51
4.4.1	Setup .....	51
4.4.2	Resultados de las medidas .....	56
4.5	Modelado .....	59
4.5.1	Regresiones .....	60
4.5.2	Redes neuronales.....	61
4.6	Validación y resultados .....	62
4.7	Conclusiones.....	67

4.8	Bibliografía .....	68
5.	Fuente de energía solar perpetua para nodos sensores.....	73
5.1	Introducción .....	73
5.2	Estado del arte .....	73
5.3	Diseño.....	75
5.3.1	Main path .....	76
5.3.2	Backup path.....	78
5.4	Implementación .....	78
5.4.1	Hardware.....	79
5.4.2	Carga.....	82
5.4.3	Sistema de adquisición.....	83
5.5	Resultados y discusión .....	83
5.6	Conclusiones.....	87
5.7	Bibliografía .....	88
6.	Viabilidad de recolección de energía RF para WSN .....	91
6.1	Introducción .....	91
6.2	Implementación del sistema .....	92
6.2.1	Antena .....	92
6.2.2	Etapas de sintonización .....	93
6.2.3	Doblador de tensión.....	94
6.3	Simulación y resultados.....	98
6.4	Conclusiones.....	101
6.5	Bibliografía .....	102
7.	Nodo alimentado por energía solar para sistemas de irrigación .....	103
7.1	Introducción .....	103
7.2	Desarrollo hardware wEcoValve .....	104
7.3	Desarrollo de los protocolos de sincronización .....	106
7.3.1	Protocolos de división de tiempo.....	106
7.3.2	Propuesta .....	107
7.3.3	Desarrollo del firmware .....	109
7.4	Resultados y discusión .....	110
7.5	Conclusiones.....	113
7.6	Bibliografía .....	113
8.	Estación meteorológica de ultra-bajo consumo .....	115

8.1	Introducción .....	115
8.2	Descripción del sistema.....	116
8.2.1	Hardware.....	116
8.2.2	Software .....	119
8.3	Resultados.....	121
8.4	Conclusiones.....	123
8.5	Bibliografía .....	124
9.	Conclusiones.....	127
9.1	Conclusiones generales.....	127
9.2	Trabajo futuro .....	129
9.3	Contribuciones y publicaciones.....	130
10.	Anexos.....	133
10.1	Diagrama de operaciones para implementar la metodología de modelado .....	133
10.2	Circuitos.....	134
10.2.1	Circuito del banco de desgaste de baterías .....	134
10.2.2	Circuito de la fuente de alimentación .....	135
10.2.3	Circuito del nodo de irrigación .....	136
10.2.4	Circuito de la estación meteorológica.....	138
10.3	Tablas de resultados de los modelos sobre trazas.....	139
10.4	Otras publicaciones no relacionadas con la tesis.....	142

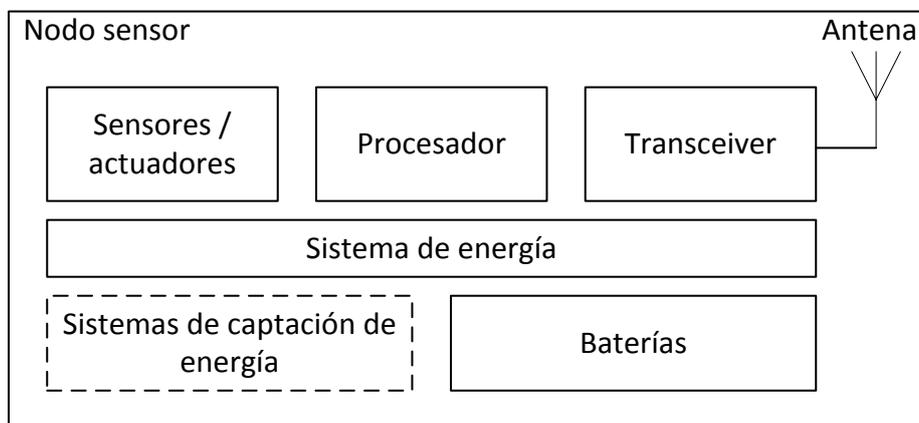
# 1. INTRODUCCIÓN

---

Las redes de sensores inalámbricas (WSN, *Wireless Sensor Networks*) son un nuevo y muy activo campo de investigación. Ha sido señalado como la primera de una lista de las diez tecnologías que cambiarán el mundo por la revista *Technology Review* del MIT [1].

Las redes de sensores están formadas por una serie de pequeños nodos que típicamente sensan algún parámetro y se comunican entre sí de forma inalámbrica [2]. Estos nodos son pequeños dispositivos electrónicos autónomos y potencialmente reubicables. Todos los nodos de la red trabajan de una forma coordinada para llevar a cabo el trabajo de una determinada aplicación. Para mantener la autonomía necesitan una fuente de alimentación y reducir en lo posible su consumo.

Los nodos sensores están formados generalmente por un mecanismo de transmisión inalámbrico, una unidad de procesamiento —normalmente un microcontrolador—, sensores o actuadores y un elemento para almacenar la energía —típicamente una batería—. En la Figura 1-1 se representa esquemáticamente esta estructura.



**Figura 1-1: Diagrama de bloques genérico de un nodo final de redes de sensores inalámbricas.**

Cada vez hay más bibliografía y se desarrollan más aplicaciones con redes de sensores inalámbricas, en el siguiente capítulo se comentarán algunas de las más habituales y representativas. Uno de los campos donde más se ha trabajado es en la monitorización de parámetros ambientales, aplicado a la agricultura.

## 1.1 Problemática a tratar

Los nodos sensores tienen una serie de limitaciones: necesitan ser pequeños para poder ser colocados en cualquier lugar, necesitan poder comunicarse para enviar y recibir información y, además eso, lo tienen que combinar con el hecho de ser muy eficientes energéticamente para lograr ser autónomos.

Las baterías que incorporan los nodos de las redes de sensores pueden ser recargables o no recargables; en el caso de las no recargables, cuando la energía almacenada en ella se agote, un operario necesitará reemplazarla; en el caso de las baterías recargables, o bien será posible diseñar un mecanismo de recarga automática, o bien se necesitaría igualmente un operario para recargarlas manualmente. La eficiencia energética permitirá que los nodos puedan funcionar durante más tiempo sin intervención de un operario, o que las baterías que se utilicen sean más pequeñas y baratas.

Además, la mejora de la eficiencia energética posibilitará el que se puedan utilizar técnicas de recolección de energía (*energy harvesting*) [3], las cuales pueden proporcionar nueva energía a los nodos. Estas técnicas extraen energía del medio, por ejemplo de radiación solar, del viento, de diferencias de temperatura, del movimiento, etc. Estas técnicas generalmente proporcionan cantidades de energía relativamente baja y muchas de ellas no son constantes, pero estos inconvenientes podrán solventarse si los requisitos de energía son menores a la energía media generada.

Por lo tanto, cuanto menos energía consuma un nodo sensor, éste será capaz de trabajar por más tiempo para la misma energía almacenada. Además, si los requisitos energéticos son bajos, el espectro de técnicas de recolección de energía que se podrían utilizar será más amplio, permitiendo así técnicas como extracción de energía del ambiente, por ejemplo de RF, temperatura, movimiento...

Por tanto, el estudio de la eficiencia energética en las redes de sensores es un problema de vital importancia en su área de investigación y cuya mejora ampliará los límites de las aplicaciones que se pueden cubrir con esta tecnología.

## 1.2 Objetivos

El objetivo principal de esta tesis es el estudio del consumo en los nodos finales de las redes de sensores inalámbricas. El mejor conocimiento derivado de este estudio permitirá realizar optimizaciones sobre estos nodos y, a su vez, estas optimizaciones harán posible implementar aplicaciones con consumos mínimos.

De este objetivo principal de la tesis se derivan otros objetivos específicos, los cuales se pueden resumir en los siguientes puntos:

- Estudiar las herramientas más utilizadas en redes de sensores. Para construir una red de sensores es habitual partir de un hardware y parte de un software prediseñado. El hardware son los nodos sensores, también llamados *nodes*. El software son *frameworks* o sistemas operativos específicos para redes de sensores.

- Estudiar el comportamiento en lo relativo al consumo de energía de las herramientas más utilizadas en redes de sensores. Tanto los *motes* como los sistemas operativos han desarrollado varias estrategias para gestionar su consumo. Conocer estas estrategias permitirá determinar cuáles son más adecuadas para cada ocasión.
- Modelar el consumo energético en nodos finales típicos de las redes de sensores inalámbricos. Desarrollar modelos de consumo permitirá por una parte conocer mejor el estado actual de cada uno de los nodos, lo que redundará en determinar qué acciones consumen más, cuales tienen más margen de mejora y cuales aportan un mayor beneficio.
- Realizar aplicaciones de redes de sensores inalámbricas con un consumo muy reducido destinadas a la monitorización ambiental y agrícola.

### 1.3 Estructura de la tesis

Esta tesis se ha estructurado como se describe a continuación:

En el capítulo 2 se estudiará el estado del arte de las redes de sensores en general y del consumo de energía en ellas y en los nodos sensores en particular. Además se comentarán varias aplicaciones que se han implementado con redes de sensores.

En el capítulo 3, se analizarán las plataformas y sistemas operativos más utilizados en redes de sensores inalámbricos. En esta parte, además se estudiará el comportamiento que presentan estas plataformas respecto al consumo de energía con algunos programas típicos y representativos.

El siguiente capítulo se dedica al modelado del consumo de energía en nodos sensores. En este capítulo se presentará una metodología para obtener modelos del estado energético de un nodo sensor. Además, se propondrán varios modelos que presentan mejoras respecto a los tradicionales y se validarán en nuevas condiciones.

Los capítulos 5 y 6 se dedican a estudiar el sistema de alimentación de estos nodos. En primer lugar, en el capítulo 5 se propondrá una arquitectura de fuente de alimentación perpetua basada en radiación solar con un doble sistema de almacenamiento de energía. En el siguiente capítulo se estudiará la viabilidad de disponer de un sistema de recolección de energía secundario basado en la captación de energía RF del ambiente y sin emisor dedicado que sirva de complemento al sistema primario.

Los últimos capítulos contienen varias aplicaciones de redes de sensores en los que se ha buscado minimizar el consumo. Estas aplicaciones están enfocadas al ámbito agrícola y medioambiental. En el capítulo 7 se presenta un nodo final que puede regular regadíos mediante una electroválvula, las órdenes de abrir y cerrar el sistema de riego viene del *gateway* de la red y se dispone de comunicación bidireccional con un protocolo de sincronización. En el capítulo 8 se muestra un nodo sensor que implementa una estación meteorológica de muy bajo consumo; esta estación recoge diversos parámetros medioambientales y puede presentarlos y ofrecer servicios a terceros a través de un interfaz web.

Finalmente, el último capítulo se dedica a presentar las conclusiones globales de esta tesis y las posibles líneas de trabajo futuro.

Estos capítulos son autocontenidos, por lo que, aunque el hilo conductor de la tesis es el consumo energético en nodos sensores, cada capítulo se puede tratar independientemente de los demás, con su propia introducción, conclusiones y bibliografía.

### 1.4 Bibliografía

- [1] Technology Review. *10 Emerging Technologies That Will Change the World*. Editado por MIT. febrero de 2003. <http://www.technologyreview.com/Infotech/13060/page2/> (último acceso: 2 de mayo de 2008).
- [2] Ilyas, M., & Mahgoub, I. (Eds.). (2004). *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press.
- [3] Seah, W. K., Eu, Z. A., & Tan, H. P. (2009, May). Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on* (pp. 1-5). IEEE.

# 2. ESTADO DEL ARTE

---

En este capítulo se tratarán los conceptos teóricos principales y las aplicaciones de las redes de sensores mediante varios ejemplos representativos. Además, se hará especial énfasis en la gestión y optimización del consumo de energía de las redes y de los nodos sensores en particular.

## 2.1 Redes de sensores

Las redes de sensores es un tópico relativamente nuevo y emergente en el campo de la computación [1]. Se trata de construir pequeños nodos autónomos que se comunican entre sí. Cada nodo tendrá un procesador y una cantidad de memoria que le otorgan cierta capacidad de procesamiento.

Los nodos de las redes de sensores acostumbran a comunicarse entre sí de forma inalámbrica, en este caso se habla de redes de sensores inalámbricas. La comunicación suele establecerse por radiofrecuencia (aunque hay otras, desde ópticas a acústicas) y corto alcance. Por este motivo, otro de los componentes que tienen los nodos de las redes de sensores son los *transceivers* de comunicaciones.

Estas dos características (procesamiento y comunicaciones) permiten a las redes de sensores realizar tareas complejas debido a la colaboración entre los nodos que las componen.

Los nodos de las redes de sensores se pueden distribuir de muchas formas, por ejemplo con comunicación punto a punto, por grupos con un nodo coordinador, con comunicación con todos los vecinos, redes híbridas, etc. Para que los datos lleguen al destino, las redes suelen utilizar una pila de protocolos que incluya un tipo de enrutamiento multisalto (*multihop*). En la actualidad los protocolos de comunicaciones en las WSN es uno de los temas de investigación más activos.

El alto grado de adaptabilidad hace que las redes de sensores sean idóneas para entornos cambiantes. También son muy utilizadas en el ambiente académico para probar nuevos algoritmos y protocolos que, de otra forma, harían más costosas las pruebas con hardware real [2].

Las áreas de aplicación de las WSN son muy variadas, por ejemplo se utiliza la tecnología RFID para gestionar los equipajes en aeropuertos y se está estudiando su uso en supermercados;

sensores ambientales, monitorizando terrenos, variables climáticas o especies animales; automoción, domótica, medicina, defensa, etc. [3].

## 2.2 Aplicaciones

Las redes de sensores, como se ha comentado antes, se han usado en el ámbito académico para investigar sobre protocolos de redes de comunicaciones. Además de en el terreno de la investigación, ha habido numerosos proyectos para utilizar los conceptos de las redes de sensores en aplicaciones específicas y, en algunos casos, comerciales.

En los siguientes apartados se mostrarán algunos ejemplos representativos de algunas aplicaciones prácticas en las que se han utilizado las redes de sensores.

### 2.2.1 Medio ambiente

El ámbito del medio ambiente quizás haya sido uno de los primeros beneficiados por las redes de sensores. Por ejemplo, se han utilizado redes de sensores para monitorizar la actividad sísmica de volcanes activos [4]. En esta aplicación se utilizaron 16 nodos Tmote Sky a lo largo de la cordillera del volcán Reventador en Ecuador; dichos nodos iban retransmitiendo los datos mediante un protocolo *multihop* hasta una estación base que retransmitía los datos a gran distancia.

También se han desarrollado varios proyectos para la detección temprana de incendios en bosques [5][6]. Estas aplicaciones detectan incendios antes que otros métodos basados en satélites. En [5] los nodos sensan parámetros como la temperatura y humedad, los cuales retransmiten y son utilizados en los nodos centrales mediante algoritmos basados en redes neuronales artificiales para detectar la presencia de incendios. En [6] añaden detectores de humo, utilizan una suma ponderada de los parámetros medidos para discernir si hay fuego o no y presentan los datos y alarmas vía web.

En aplicaciones agrícolas se puede comentar la monitorización de viñedos que hacen en [7]. En esta aplicación se dispone de un nodo central y varios nodos terminales que recogen datos ambientales y agrícolas mediante sensores de humedad de hoja y humedad del suelo.

En [8] desarrollan un sistema eficiente de control de regadíos en Australia basado en una matriz de sensores con nodos Tmote Sky. La información de estos sensores corresponde a la humedad en diferentes profundidades del suelo, la cual es transmitida a través de múltiples saltos en la red a un nodo central con conexión 3G. La información recopilada ayudará a un operario a tomar la decisión de cuándo regar y cuándo no.

También hay aplicaciones genéricas de monitorización meteorológica y del hábitat como [9] y [10]. En la primera de ellas realizan una monitorización de una pequeña isla mediante una red jerárquica y en la segunda se experimenta con nuevos sensores para crear una estación meteorológica, ambas con acceso a internet a través de sus respectivos *gateways*.

### 2.2.2 Domótica y salud

Una de las primeras aplicaciones en las que se pensó aplicar las redes de sensores fue a la domótica mediante el concepto de «polvo inteligente». Este concepto se deriva del proyecto

Smartdust de Berkeley [11] y hacía referencia a pequeños dispositivos, virtualmente imperceptibles, que realizan tareas y se comunican entre sí. De hecho, de este concepto viene el nombre de «motas» o «*motes*» para referirse a los nodos, haciendo un juego de palabras entre «*remote*» y una analogía con las motas de polvo.

En domótica el principal interés ha sido introducir una tecnología inalámbrica y ampliamente usada en redes de sensores en aplicaciones comerciales [12]. Una de las aplicaciones más sencillas es realizar conexiones punto a punto sustituyendo a otras tecnologías, por ejemplo en el caso de RF4CA para mandos a distancia y, en general, todos los productos comerciales listados en [13].

El proyecto Eaton's Home Heartbeat System iba un paso más allá buscando conectar todos los dispositivos de una casa. Estaba especialmente enfocado a seguridad, pero también permitía monitorizar y controlar diversos objetos. Por desgracia este proyecto no ha tenido continuidad.

Volviendo al terreno de la investigación, y en un campo a medio camino entre el hogar y la salud se sitúa el proyecto [14], donde se pretende aprovechar la infraestructura típica de un hogar para mejorar la calidad de vida de pacientes, la detección temprana de algunas dolencias y mejorar la comunicación entre el médico y el paciente. Por otro lado, en [15] se despliega una red que combina el sensado del ambiente con el de la salud de personas para determinar si es necesario activar un protocolo de asistencia.

### **2.2.3 Industriales y vehículos**

En entornos industriales ocurre algo parecido al ámbito doméstico: la primera aproximación consiste en sustituir dispositivos que ya existen, aportando nuevas características. Así, hay en el mercado numerosos dispositivos para sustituir distintos buses de comunicaciones con tecnologías inalámbricas, típicamente sustitución de RS232 con zigbee, bluetooth o wifi.

En [16] se identifican varias áreas donde las redes de sensores pueden aportar un valor añadido, como son la detección de eventos, recolección de información, diagnóstico de máquinas y sistemas de seguridad. Por ejemplo se cita que la empresa BP [17] utiliza redes de sensores para monitorizar el nivel de gas licuado de petróleo en tanques de almacenaje. O la NASA [18], que emplea nodos para monitorizar remotamente pruebas de propulsión de cohetes.

En automoción ha habido varios fabricantes y consorcios que han presentado diversos proyectos para utilizar los conceptos de las redes de sensores en coches. Por ejemplo el programa V2V del proyecto Car2Car [19] pretende que los coches se comuniquen inalámbricamente entre sí para alertarse de peligros, obstáculos que puedan encontrarse en su camino y de la presencia de vehículos de emergencias. El estado actual de este proyecto es prometedor, ya que se encuentra haciendo pruebas reales con prototipos.

En otro proyecto híbrido entre salud y vehículos se sitúa [20], donde el objetivo es monitorizar a cierto tipo de personal en trabajos que requieren mucha concentración y asistirles en caso de necesidad. Una parte de este proyecto consta de redes de sensores para recolectar información de diversos sensores dentro de un vehículo, como los que se utilizan para

averiguar el estado del conductor, y transmiten la información a una centralita para realizar un procesado y tomar decisiones.

### 2.2.4 Eficiencia energética

Un campo donde se está explotando comercialmente a las redes de sensores es el de la eficiencia energética en el hogar y, sobre todo, en industrias. De hecho, una de las empresas pioneras en redes de sensores, Moteiv —luego Crossbow— se reconvirtió en Sentilla [21], una empresa enfocada en productos de monitorización de energía. Por ejemplo, mediante el empleo de redes de sensores —algo mínimamente invasivo—, esta empresa puede detectar patrones que permiten planificar mejor las cargas de trabajo en *data centers*.

El camino de Sentilla ha sido recorrido también por otras empresas, por ejemplo la española AdvanticSys [22] también dispone de herramientas para desplegar una red de sensores en un edificio y realizar una auditoría energética leyendo datos de contadores, sensores, detectores, etc.

Otro proyecto interesante ha sido EQUOS Lighting System de Philips Lighting, en el cual se utilizan balastos de lámparas fluorescentes equipadas con comunicación zigbee para realizar tareas de mantenimiento, control y estudiar mecanismos de ahorro. Este proyecto ha culminado en un producto comercial llamado Hue [23], a medio camino entre las redes de sensores y el *Internet of Things* (IoT).

### 2.2.5 Aplicaciones militares y entornos de alta seguridad

También ha habido propuestas para utilizar las redes de sensores en el campo de batalla. Por ejemplo, en [24] utilizan una red agrupada en clústers para sensor señales acústicas y magnéticas que puedan dar lugar a detectar y localizar vehículos y movimiento de tropas en tiempo real.

En labores defensivas, en [25] proponen una red de sensores robusta y de baja latencia para la detección de posibles ataques contra instalaciones estratégicas o civiles. Para ello se plantean el escenario de una infraestructura de un metro de superficie y monitorizan vibraciones, temperaturas, ruido, olor, etc. para detectar infracciones en la seguridad.

Por otro lado, otro ámbito de aplicación es en lugares con un alto nivel de seguridad, tales como centrales nucleares. En [26] se hace un análisis de los requisitos y la aplicación de las redes de sensores inalámbricas aplicadas a la monitorización de centrales de generación de energía nuclear. Por ejemplo, comentan el interés en sensor temperatura, vibraciones, sonido y radiación en los reactores para enviar los datos a la sala de control.

## 2.3 Características de las redes de sensores

### 2.3.1 Topología de la red y comunicaciones

Una de las principales características de las redes de sensores es que se comunican entre sí de forma inalámbrica. La forma de comunicación puede utilizarse para clasificar las redes atendiendo a varios parámetros. Como se ha comentado en la introducción, en redes de sensores se trabaja con topologías de redes muy diversas. Los tipos de redes pueden ser

centralizadas, malladas, en árbol con todos los nodos iguales, con nodos distintos, con nodos móviles, estáticos, etc. [27].

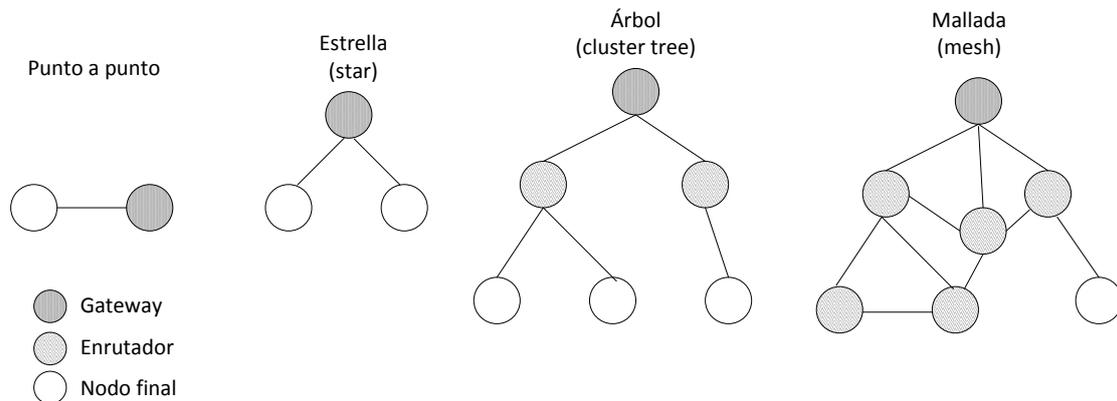


Figura 2-1: Distintos tipos de topologías utilizados en redes de sensores

En la Figura 2-1 se muestran las arquitecturas de redes de sensores más utilizadas. En primer lugar está la más trivial: comunicación punto a punto; en segundo lugar una red en estrella, donde hay un nodo central, que es el único con el que se comunican el resto de nodos; el tercer tipo es una red en árbol, la cual puede verse como la unión de varias redes en estrella a través de sus respectivos nodos centrales; y finalmente se muestra una red mallada, donde los nodos pueden comunicarse entre sí sin tener que pasar por un nodo central.

La comunicación entre máquinas es un proceso que puede hacerse tremendamente complicado, de ahí que haya que dividir este problema en varios subproblemas y resolverlos individualmente. Esta es la filosofía de la arquitectura por capas que se utiliza en las redes de computadores y redes de sensores. Un protocolo es un conjunto de reglas que gobiernan cómo dos o más entidades interactúan entre sí. Típicamente las tareas de los niveles más bajos las desarrollan los circuitos integrados encargados de actuar como interfaz de red (transceptor o *transceiver*); mientras que los niveles superiores se implementan por software.

Al transmitir información, los datos se generan en el nivel superior del emisor, los datos irán pasando de un nivel al inmediatamente inferior a través de los interfaces que tienen definidos los protocolos. Estos interfaces básicamente son estructuras de datos y funciones que deben reconocer un nivel y el contiguo. Cada nivel realizará las operaciones oportunas con los datos, comúnmente se añadirán cabeceras de control y códigos de detección o corrección de errores. Además, algunos niveles pueden llevar a cabo tareas adicionales como un control del flujo, retransmisiones, etc.

En recepción el proceso es el inverso. Los datos llegan al nivel más bajo y, de ahí, se va pasando a un nivel superior donde se vuelven a realizar ciertas tareas y se desglosan las estructuras de datos.

En la Figura 2-2 se pueden ver varias pilas de protocolos típicas comenzando por el más teórico modelo OSI, y a continuación tres sistemas de comunicación inalámbricos: wifi, bluetooth y zigbee.

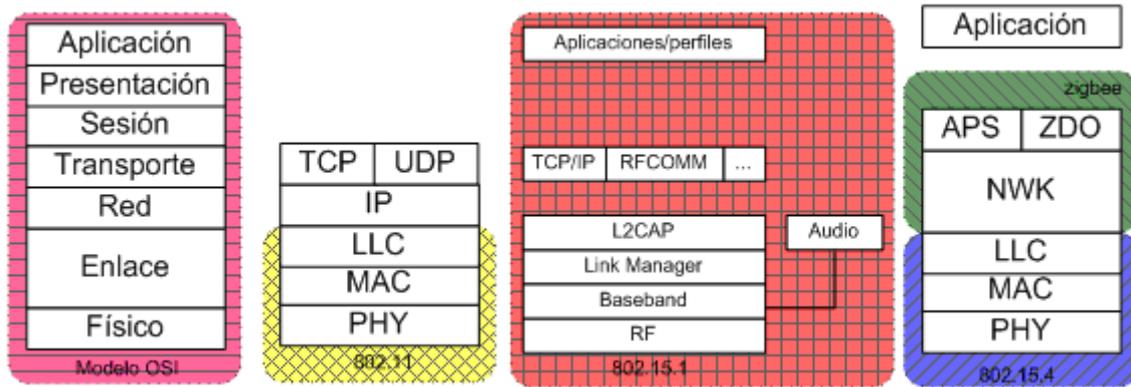


Figura 2-2: Pila de protocolos del modelo OSI (izquierda) y diferentes implementaciones en sistemas de comunicación inalámbricos de alta frecuencia.

Las funciones principales de cada nivel se pueden resumir brevemente en:

- Nivel físico: se encarga del envío de información entre los nodos a través del medio. Entre otras cosas, establece el tipo de modulación y codificación. En este nivel se establece el código que se utilizará para transmitir la señal, como la codificación Manchester, AMI, NRZ, etc. Una codificación muy utilizada en redes de sensores es la DSSS (*Direct-Sequence Spread Spectrum*) que consiste en dividir cada byte en dos símbolos y codificar cada uno en una secuencia preestablecida de 32 bits (*chips*), con lo cual se transforman 4 bits en 32; en el dominio de la frecuencia se consigue que la amplitud se reduzca y se ensanche el ancho de banda; así, aunque no se consigue un gran rendimiento en términos de velocidad, se puede mejorar la coexistencia con otras señales de banda estrecha que utilicen la misma frecuencia porque para éstas, la señal DSSS será simplemente un leve aumento del ruido de fondo. En este nivel también se establece el tipo de modulación empleado, el cual tiene un impacto en la inmunidad al ruido, la posibilidad de enviar varias señales, etc.
- Nivel de enlace de datos: se encarga de controlar el acceso al medio si éste es compartido, ahorrar energía, minimizar la latencia, proporcionar escalabilidad y servicios de transmisión, recepción, confirmación, control de errores, etc. Los protocolos de este nivel se pueden dividir en:
  - Ranurados: basados en slots o cuantos de tiempo. Algunos ejemplos son: TDMA, 802.15.4, SMAC, DSMAC, TMAC, TRAMA, etc.
  - De muestreo: buscan periódicamente actividad en el medio. Algunos ejemplos son: ALOHA, BMAC, WiseMAC, CSMA, etc.
  - Otros, como STEM, híbridos como ZMAC, SCP-MAC, etc.
- Nivel de red: se encarga de encaminar los datos para que lleguen desde un emisor a un receptor a través de una ruta que puede estar compuesta por otros nodos. En redes de sensores hay diversos protocolos de nivel de red, como zigbee, *Collection Tree Protocol* (CTP), AODV, SPIN, MCFA, COUGAR, etc. [28].
- Nivel de transporte: mantiene el flujo y controla los errores que se puedan producir en la comunicación a un nivel más alto que el de enlace de datos (por ejemplo detectaría información duplicada, en orden incorrecto, etc.).
- Nivel de sesión: si hay una comunicación bidireccional este nivel se encarga de regularlo, además de la recuperación de errores.

- Nivel de presentación: uniforma la representación de la información (ASCII, Unicode...).
- Nivel de aplicación: define las normas para intercambiar los datos provenientes de la aplicación del usuario.

En las redes de sensores tienen especial importancia los tres primeros, siendo determinantes en la eficiencia energética de la propia red. Existen varias implementaciones de estos tipos de protocolos. Por ejemplo el más típico es 802.15.4/zigbee, que será el utilizado en esta tesis. Otras pilas de protocolos son 6LoWPAN, Ant, Wibree, MiWii, etc.

802.15.4 implementa los niveles físico y de enlace de datos. En el nivel físico utiliza una modulación de ensanchado de espectro y O-QPSK a 2,4 GHz, 250 kb/s: Proporciona unos servicios a la capa de enlace de datos tales como indicación si hay datos, leer y establecer ciertas propiedades, como la potencia, activación y desactivación del *transceiver*, seleccionar el canal, etc. En el nivel de enlace de datos establece el formato del paquete y el tipo de tramas, proporciona varios servicios como la petición y respuesta de una asociación, petición de datos, etc. También se delimitan unos tiempos llamados *slots* donde ciertos nodos podrán transmitir, además de establecer un mecanismo opcional de tramas señalizadoras o *beacon* para establecer un mecanismo de sincronización.

Zigbee se sitúa por encima de 802.15.4, en el nivel de red, y establece diferentes topologías de red: en estrella, árbol, malladas y varios servicios: unirse o abandonar una red, seguridad, rutado de los paquetes, etc.

### 2.3.2 Motes

Los nodos de las redes de sensores se suelen llamar *motes* o *motas*. En la sección 2.2.2 se mencionaba la etimología de este nombre, el cual proviene de un juego de palabras entre una mota de polvo –aludiendo a su pequeño tamaño y ubicuidad– y la palabra «*remote*».

Una red puede tener diferentes tipos de nodos: nodos finales, nodos coordinadores y *gateways*, tal como se muestra en la Figura 2-1. Éstos pueden implementarse con el mismo hardware o distinto. De acuerdo a los requisitos de cada aplicación es posible que los coordinadores y *gateways* necesiten más capacidad de procesamiento y memoria, otros sistemas de comunicación, no necesiten sensores, etc. En cualquier caso, en el resto de la sección se asumirán nodos finales salvo indicación de lo contrario.

En la Tabla 2-1 se listan algunos de los nodos para redes de sensores más utilizados, junto con sus principales características. Durante esta tesis se hará uso extensivo de los dos primeros, que se estudiarán en detalle en el siguiente capítulo; estos dos son los más utilizados en la bibliografía y tienen un amplio soporte. MICAz está fabricado por Crossbow y actualmente puede adquirirse en Memsic. Por otra parte Tmote Sky estaba fabricada por Moteiv y luego Sentilla, y su clon llamado telosb por Crossbow; actualmente también puede ser adquirida en Memsic o AdvanticSys. Además, sus ficheros de diseño están publicados, por lo que cualquiera puede fabricar sus propios nodos o hacer modificaciones de los mismos. Waspote es una plataforma diseñada por la empresa Libelium que cuenta con un hardware bastante flexible, basado en módulos intercambiables. Eyes cuenta con varias versiones, desarrolladas por Infineon, todas ellas basadas en el micro MSP430, aunque en la actualidad el producto está

## 2. Estado del arte

descatalogado. IMote es una de las plataformas más poderosas en cuanto a capacidad de cómputo, ya que incorpora procesadores ARM de 32 bits, por lo que un uso típico es como *gateway* de una red; está diseñada por Intel. Iris mote es un desarrollo derivado de MICAz pero con una actualización de sus componentes. La XYZ es otro nodo sensor de alta potencia desarrollado en la universidad de Yale. Finalmente, la Z1 es un sistema relativamente nuevo comercializado por Zolertia.

Por otra parte, en cuanto a microcontroladores utilizados se puede ver que tres familias claramente diferenciadas: TI MSP430, ARM Atmel ATmega. Siendo el primero de ellos muy conocido por su bajo consumo, el segundo por su mayor potencia de cómputo y el tercero por el amplio uso y soporte.

En cuanto a *transceivers* destaca sobre todo el CC2420 de Chipcon, luego adquirido por Texas Instruments, que permite comunicaciones sobre 802.15.4. Otros en menor medida o bien se decantan por bluetooth o bien permiten varios sistemas.

Tabla 2-1: Comparativa entre motes.

	Procesador	Transceiver	Sensores
<b>MICAz</b>	Atmel AVR ATmega128L 128 kB Flash de programa, 4 kB RAM, 4 kB EEPROM	Chipcon CC2420 250 kbps 2.4 GHz	Módulos externos
<b>Tmote Sky</b>	TI MSP430F1611 8 MHz 10 kB SRAM, 48 kB Flash + 1024 kB	Chipcon CC2420 250 kbps 2.4 GHz	Humedad, temperatura y luz integrados
<b>Waspnote</b>	ATmega1281 14 MHz 8 kB SRAM, 128 kB Flash, 4 kB EEPROM	Modular (802.15.4, Bluetooth, wifi, 3G...)	Módulos externos
<b>EyesIFX 1</b>	TI MSP430F149 2 kB SRAM, 60 kB Flash	TDA5250 64 kbps 868 MHz	Temperatura, luz Módulos externos
<b>IMote 1</b>	ARM 7 TDMI 12 MHz 64 kB SRAM, 512 kB Flash	Bluetooth 2.4 GHz	
<b>Iris mote</b>	Atmel AVR ATmega1281 128 kB Flash de, 8 kB RAM	AT86RF230 2.4 GHz	Módulos externos
<b>XYZ</b>	ARM OKI ML67Q500x 57,6- 1,8 MHz 256 kB Flash, 32 kB RAM, 4 kB EEPROM, 2 Mb RAM externa	CC2420 250 kbps 2.4 GHz	Luz, temperatura, aceleración
<b>Z1</b>	TI MSP430F2617 16 MHz 8 kB RAM, 92 kB Flash	Chipcon CC2420 250 kbps 2.4 GHz	Temperatura, aceleración

La reducción del tamaño, coste y consumo de las distintas tecnologías de procesadores y *transceivers* ha permitido la posibilidad de hacer viables los *motes* [27]. Internamente, los *motes* constan de un procesador, un *transceiver* de comunicaciones y uno o varios sensores o actuadores. El procesador suele ser un microcontrolador, el cual incorpora memoria y otros periféricos.

Existe la posibilidad de hacer *motes ad-hoc* para cada aplicación. Sin embargo si no se persigue un objetivo muy específico o comercial, no suele ser rentable invertir tiempo y dinero en ello porque se pueden adquirir estos *motes* genéricos.

### 2.3.3 Sistemas operativos

Un sistema operativo no es más que un programa. Este programa controla la ejecución de otros programas y proporciona un interfaz entre éstos y el hardware [29]. Los motivos para usar un sistema operativo son:

- Hacer que el equipo sea más cómodo de usar, tanto para el usuario como para el programador. El sistema operativo crea una o más capas jerárquicas que permiten abstraer el hardware; además se proporciona una serie de programas que los desarrolladores pueden utilizar en sus aplicaciones, por ejemplo funciones de entrada/salida (I/O), de creación de procesos, etc.
- Mejorar la flexibilidad del sistema. La abstracción mencionada en el punto anterior permite que los cambios en el hardware sean transparentes a los programas de usuario.
- Aprovechar de la forma más eficiente posible los recursos disponibles. El sistema operativo controla quién y cuándo usa cada recurso. Se consideran recursos el tiempo de uso del procesador, periféricos, una señal software, etc. [2].

Algunos de los aspectos de los sistemas operativos mencionados con anterioridad no son aplicables a las redes de sensores, un ejemplo es el caso de la gestión de múltiples usuarios. A pesar de que ha habido algunos de los sistemas operativos relativamente genéricos que se han aplicado a las redes de sensores, como es el caso de Contiki [30], se hace evidente que es necesario crear nuevas implementaciones de sistemas operativos específicamente para las WSN.

Los principales objetivos de los sistemas operativos en las redes de sensores son:

- Facilitar el desarrollo de aplicaciones.
- Modularidad: si algún componente del sistema operativo es innecesario no debería incluirse.
- Configuración para una aplicación específica: los nodos sensores tienen una tarea determinada, a diferencia de un ordenador personal que puede ejecutar muchos programas distintos. El sistema sería más eficiente si se puede adaptar a la aplicación del nodo.
- Optimizar las comunicaciones, ya que es uno de los aspectos clave de la red y además es de los que más energía consume.

Las particularidades de las WSN hacen que en el diseño de estos sistemas operativos haya que tomar algunas decisiones con el fin de favorecer algunos de los objetivos anteriores.

El paradigma de programación tradicional es el imperativo o procedural, el comportamiento del programa se basa en ejecutar una tras otra todas las instrucciones de una forma secuencial, algunas de estas instrucciones son llamadas a funciones; por otra parte los datos están almacenados en variables, ambos –datos e instrucciones– son independientes entre sí. Existen otros paradigmas diferentes, por ejemplo la programación orientada a objetos, orientada a aspectos, flujo de datos, etc. Cada aplicación se adaptará mejor a uno u otro paradigma. En el caso de las aplicaciones habituales de las WSN los mejores modelos de programación son los que están conducidos por eventos y los concurrentes expulsivos [1].

Entre los sistemas operativos para redes de sensores se pueden comentar los siguientes:

- TinyOS: uno de los primeros sistemas operativos para redes de sensores y el que más popularidad ha conseguido. En último término no es más que una colección de componentes utilizados y un conjunto de herramientas para automatizar ciertas tareas en un lenguaje específico, nesC. Este lenguaje utiliza un paradigma basado en componentes y una ejecución basada en eventos.
- SOS: otro de los sistemas operativos más antiguos, aunque en la actualidad discontinuado. Al igual que el anterior, utilizaba un paradigma basado en componentes y un sistema de ejecución basado en eventos.
- Mantis OS: un sistema operativo para redes de sensores con capacidades derivadas de sistemas operativos más potentes, con planificador de tareas, compatibilidad con POSIX, multitarea y ejecución basada en hilos.
- Contiki: sistema operativo originalmente para sistemas embebidos que se ha portado también a redes de sensores. Se suele utilizar en combinación con uIP y protothread ya que son obra de su mismo autor. uIP es una flexible pila de comunicaciones y los protothreads son un mecanismo de programación para implementar concurrencia en sistemas de muy bajos recursos. Utiliza un paradigma basado en procesos e hilos y un sistema de ejecución que combina los eventos con la multitarea.

Otros sistemas operativos que se pueden enumerar son: Nano-RK, t-kernel, Bertha, CORMOS, SensOS, etc. En el próximo capítulo se ampliarán más estos conceptos y la descripción de estos sistemas operativos.

## 2.4 Gestión de energía en redes de sensores

Una de las grandes limitaciones de las redes de sensores es la disponibilidad de energía, por ello es necesario establecer estrategias para gestionar adecuadamente el consumo de la energía disponible y para maximizar la consecución de más energía.

### 2.4.1 Recolección de energía

Las redes de sensores típicamente almacenan su energía en baterías, las cuales tienen una capacidad finita. A veces es muy dificultoso reemplazar las baterías si se agotan, por lo que en

aplicaciones pensadas para funcionar de forma ininterrumpida durante un largo periodo de tiempo es necesario que los nodos consigan por ellos mismos energía con la que autoabastecerse. Esta necesidad ha dado lugar a una línea de investigación llamada *energy harvesting*, o *energy scavenging*.

Una de las fuentes de energía más utilizadas es la solar. Esto se debe a que tanto la fuente de energía como la tecnología están muy extendidas y se obtienen rendimientos muy altos comparados con otras fuentes. Por ejemplo en [31] se propone un seguidor del punto de máxima potencia de paneles solares aplicado a nodos de redes de sensores para obtener la máxima energía con la que recargar una batería; para ello emplean un convertidor DC/DC variable controlado por un microcontrolador.

Otra fuente que se ha aplicado ampliamente es la piezoeléctrica, donde se extrae energía de movimientos o vibraciones. Se ha aplicado en suelos para extraer energía de las pisadas, en ropa para extraer energía de los movimientos de personas o por ejemplo en [32], donde se utilizan celdas de material piezoeléctrico para aprovechar las corrientes de agua en mares y ríos para generar unos pocos milivatios, pero suficientes para cargar una batería.

Además, también ha habido intentos de extraer energía de las señales de RF en el ambiente. En [33] se ideó un sistema analógico para aprovechar la energía de emisiones de radio y cargar un condensador con el que poder alimentar un pequeño nodo.

Ha habido muchas más propuestas de fuentes de energía ciertamente imaginativas, tales como de la salinidad de plantas, del flujo de la sangre dentro del cuerpo humano, de diferencias de temperatura, etc.

Este tema ha sido aplicado en la industria y existen varias soluciones comerciales que pueden utilizarse fácilmente. En [34] se listan y comparan varios de ellos, como los solares EnOcean ST100 y Solio, los basados en vibraciones Perpetuum PMG7 y Cedrat APA400M-MD o los sistemas basados en temperatura de Tellurex.

### **2.4.2 Hardware**

Tanto en el hardware como en el firmware hay muchas pequeñas cosas que se pueden hacer para ahorrar el consumo. Normalmente no se repara en ellas bien porque son aspectos más propios del desarrollo que de la investigación, bien porque se utilizan dispositivos y sistemas operativos realizados por terceros y se supone que ya han tenido en cuenta estas consideraciones. Más allá de la consideración de si es adecuado realizar dicha suposición, conviene conocer estos mecanismos para cuando haya que realizar desarrollos *ad-hoc*.

En el terreno del hardware se pueden comentar algunos puntos evidentes, como apagar todos los dispositivos que no se utilicen, tanto externos, como periféricos internos del microcontrolador. Este hecho tan trivial impone que sea conveniente utilizar dispositivos que permitan manejar modos de bajo consumo y tengan unos tiempos de activación cortos. En el caso de dispositivos que no tienen esos modos de bajo consumo, por ejemplo sensores analógicos como una PTC, conviene desarrollarlos, bien mediante interruptores en su alimentación, alimentándolos desde líneas digitales del microcontrolador (si aportan corriente suficiente) u otros [35].

También es recomendable ajustar los valores de las típicas resistencias de *pull-up* de los buses de comunicación lo más alto posible para la velocidad de transmisión deseada, dado que presentan un consumo cuando la línea está a nivel bajo [36]. Evidentemente lo mismo ocurrirá con los *pull-down* cuando la línea esté a nivel alto.

En general las tensiones de trabajo menores presentan también unos consumos menores, por lo que es conveniente alimentar el circuito a la tensión más baja posible, o si está disponible la opción, utilizar una tensión baja únicamente para el núcleo del procesador. En [37] proponen un procesador con PMU (*Power Management Unit*), el cual responde a estados y va cambiando de modo, además de variar la tensión de alimentación a través de un convertidor DC/DC variable.

De igual manera, se puede utilizar una frecuencia de reloj lo más baja posible (más adelante se matizará) y, en los casos que no sea necesario una buena resolución temporal, utilizar osciladores RC en lugar de cristales.

La mayor parte de las técnicas anteriores están dirigidas a los microcontroladores, ya que son los que controlan este tipo de nodos. Entre los microcontroladores que mejor manejan el consumo destacan Texas Instruments con los MSP430, los ARM de Energy Micro y los nanoWatt de Microchip.

El otro gran componente común a todos los *motes* que tiene mayor efecto en el consumo es el *transceiver*. Además de tener diferentes modos de funcionamiento como los microcontroladores, los *transceivers* generalmente permiten regular su potencia de transmisión, por lo que ésta puede ajustarse dinámicamente para reducir el consumo. También se han realizado estudios sobre los mejores esquemas de modulación, etapas de potencia más eficientes, etc.

La forma más efectiva de ahorrar energía en los *transceivers* es llevarlos a modos de bajo consumo cuando no son necesario, si los protocolos de comunicaciones lo permiten [38]. No obstante, para mantener la comunicación, los diferentes nodos deben establecer un mecanismo de sincronización para que unos envíen datos cuando otros escuchan. Así, desde el firmware se establecerán mecanismos como los esquemas de ciclos de trabajo dormido/desperto o equivalentes para reducir su consumo.

Otras consideraciones pueden tenerse en cuenta en el resto de dispositivos, como los sensores, actuadores, memorias, etc. No obstante, como la presencia de estos componentes no es homogénea, es complicado establecer criterios genéricos respecto a ellos.

### 2.4.3 Firmware

No se debe descuidar el estado de las líneas digitales del microcontrolador. Dado que la mayoría son bidireccionales y algunas tienen multiplexadas funciones analógicas, puede darse el caso que se esté derivando una corriente por uno de estos pines. Por ello como norma general se recomienda tener una alta impedancia en las líneas conectadas al circuito y no utilizadas en un momento dado —lo que puede conseguirse configurándolas como entrada— y, si se utiliza un bus de comunicación con resistencias *pull-up* o *pull-down*, dejarlas en el nivel que no haya diferencia de potencial en dicha resistencia.

Una técnica que a veces se utiliza es tener un manejo variable de la frecuencia de reloj. Dado que los nodos de las redes de sensores suelen tener ciclos de trabajo periódicos, sería posible utilizar un reloj de una determinada frecuencia en algunos estados (por ejemplo en estado dormido un reloj RC de baja frecuencia) y de otra frecuencia diferente en otros estados (por ejemplo en estado *listen* un reloj de cristal de mayor frecuencia).

Por otro lado, también se ha medido que energéticamente cuesta más transmitir un byte que ejecutar cien instrucciones [38][39], por tanto puede ser conveniente establecer estrategias para reducir la cantidad de bytes a enviar, tanto de la propia información de los sensores como del sobrecoste que supone la gestión de las comunicaciones y la retransmisión en caso de errores.

Cuando se habla de extender la vida de los nodos sensores, normalmente se habla de reducir su consumo medio para aumentar el tiempo que una carga inicial de la batería lo puede mantener alimentado. No obstante, en nodos que puedan autoabastecerse y, por tanto, tengan mecanismos de recarga de la batería como, por ejemplo, paneles solares, aparece otro factor que puede ser limitante: el desgaste de las baterías. Si hay un gran desgaste en las baterías, será necesario un mantenimiento más frecuente para comprobarlas y/o cambiarlas. Relacionado con este aspecto, en [40] apuntan a técnicas para reducir este desgaste y, por tanto, extender la vida de las baterías en las redes de sensores como por ejemplo que la demanda de corriente por parte de la carga conectada a la batería sea una corriente continua y baja, para lo que proponen aumentar el ciclo de trabajo a cambio de bajar la tensión de alimentación.

Todos los dispositivos, incluido el microcontrolador, se deben poner en el modo de más bajo consumo posible el mayor periodo de tiempo posible. El caso del microcontrolador es especialmente crítico porque es quien controla las operaciones y debe tener algún mecanismo para despertarse, típicamente interrupciones, bien de periféricos internos o externos. También debe tenerse en cuenta el tiempo que tarda el microcontrolador en despertarse. Por ejemplo se puede comentar el caso de TinyOS, el cual siempre pone el microcontrolador en modo dormido cuando se ejecuta sobre un MSP430, pero no cuando lo hace sobre un ATmega. El motivo es que el MSP430 tarda 6  $\mu$ s en activarse (tiempo de *wakeup*) y el ATmega varios milisegundos. Un tiempo alto como es el segundo caso haría que, si el paso a modo dormido es frecuente y de corta duración, no se consiga un ahorro efectivo [41].

Del hecho de cambiar de estados de consumo y de trabajar de forma periódica se derivan los ciclos de trabajo con esquema despierto/dormido. Además de este esquema, típico de las redes de sensores, también se pueden establecer estrategias como la transmisión conducida por datos, donde sólo se transmiten datos cuando es necesario, por ejemplo ajustando dinámicamente la frecuencia de muestreo de los sensores, comprimiendo, enviando información sólo cuando ésta cambie de valor o estableciendo mecanismos de predicción [42].

#### **2.4.4 Protocolos**

Como se ha comentado antes, los tres primeros niveles de la pila de protocolos tienen un impacto fundamental en el consumo de energía. De esta forma, se han propuesto diferentes soluciones en cada capa para lograr optimizar el consumo.

En cuanto al nivel físico y de acceso al medio se puede comentar una de las primeras propuestas: SMAC [43]. Este protocolo es sencillo: utiliza un esquema despierto/dormido donde el periodo despierto se subdivide en tres ranuras temporales (*slots*): SYNC, RTS y CTS, como se representa en la Figura 2-3. En el primer *slot* se envía un paquete de sincronización, el cual es utilizado por los nodos que lo reciben para ajustar su base de tiempos respecto al emisor; en RTS los nodos hacen una petición de envío de datos en caso que necesiten transmitir, e indican a quién enviarán esos datos; en CTS el nodo coordinador confirma el envío al nodo que lo solicitó. En el siguiente periodo los nodos, en caso de no tener que enviar o recibir datos, podrán pasar al modo dormido; mientras que si están involucrados en la transmisión, permanecerán en el estado adecuado hasta que finalice el envío y éste se confirme con un ACK (paquete de confirmación de llegada de datos).

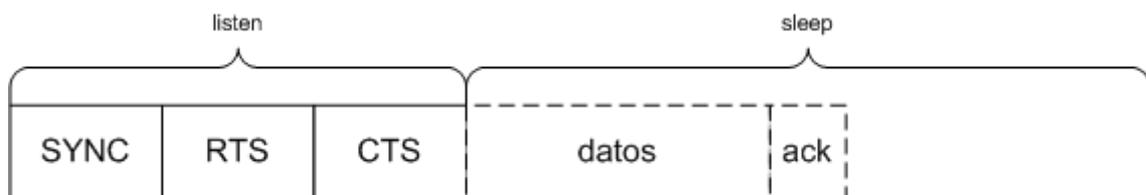


Figura 2-3: Esquema de tiempos en los que se enviarán las distintas tramas en el protocolo SMAC.

Sobre este protocolo se han propuesto algunas modificaciones, como DSMAC, el cual presenta un ciclo de trabajo variable o TMAC, que juega con las ventanas de RTS y CTS.

Otra propuesta es BMAC [44], representado en la Figura 2-4. Este protocolo supone que hay un nodo central con alimentación perpetua. Los nodos finales se despertarán periódicamente (tiempo llamado *check interval*) para comprobar brevemente si hay actividad en el canal; si la hay permanecerán a la escucha; si no, pasarán al estado dormido. El nodo central emitirá un preámbulo de duración mayor al *check interval* para que se sincronicen con él los nodos con los que haya que establecer comunicación y, al finalizar, se enviarán los datos.

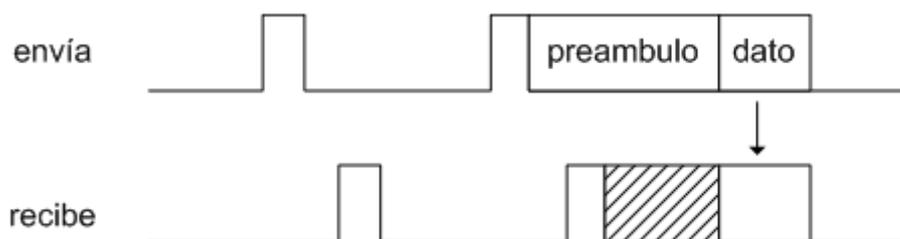


Figura 2-4: Esquema de tiempos del protocolo BMAC.

Una modificación de este protocolo es WiseMAC [45] que combina el mecanismo tradicional de CSMA/CA con el BMAC, pero éste último ajustará la duración de su preámbulo dinámicamente para reducir los tiempos despiertos.

La topología afecta directamente a la forma en que se transmiten los datos, principalmente en el direccionamiento. La forma de comunicación más directa es una punto a punto, en la que hay alcance directo entre el nodo emisor y el receptor; no obstante, esta circunstancia no puede darse en todos los casos porque el rango de cobertura limita la dispersión geográfica, porque la aplicación requiere segmentar la red, porque los nodos son móviles y reubicables,

etc. Por tanto, en muchos casos será necesario encaminar la información de un nodo a otro de forma que ésta puede pasar por diversos nodos intermedios; así se puede hablar de redes multisalto (*multihop*).

El criterio para seleccionar o diseñar un determinado tipo de topología de red puede pretender responder a una o varias cuestiones, tales como la eficiencia en el direccionamiento de los datos, la respuesta ante caídas de algunos nodos, maximizar la cobertura, minimizar la latencia, etc. [46].

Los nodos pueden caer por diversas causas, siendo una de ellas el agotamiento de la energía. Para detectar y notificar tales eventos se han propuestos varios algoritmos que básicamente consisten en monitorizar parámetros a nivel de red como los paquetes perdidos, retransmisión de tramas balizas (*beacon*), etc.

Además de detectar problemas en los nodos, también es de sumo interés evitar que estos problemas lleguen a producirse, o por lo menos retrasarlos lo máximo posible. Una solución en redes estáticas y con cierta redundancia puede ser el algoritmo PEAS, propuesto en [47], donde la red mantiene únicamente el subconjunto mínimo de nodos necesarios para que la aplicación funcione.

Por otro lado, hay diferentes aproximaciones dinámicas para minimizar el consumo de energía. Por ejemplo una de ellas es la de ajustar la potencia de transmisión de acuerdo con los requisitos en cada instante, propuesta en [48] para redes P2P. La filosofía de los agentes móviles, donde el procesado se distribuye entre los nodos, puede ser de utilidad en las redes de sensores para asignar más carga de trabajo a unos nodos respecto a otros o reducir la cantidad de información a enviar respecto a otros paradigmas como el cliente/servidor [49].

Por último cabe mencionar propuestas como [50] que no buscan optimizar el consumo de energía de un nodo o de un camino de direccionamiento en concreto, sino el de la red en su conjunto, de tal forma que la decisión tomada en un momento puntual no pueda ser la óptima, pero sí aporte ventajas a escala global y a largo plazo. Todos estos mecanismos hacen que los nodos cooperan de una forma más profunda para optimizar sus recursos.

## 2.5 Conclusión

Como se puede comprobar en este capítulo, las redes de sensores son un campo muy activo de investigación y con una gran aplicabilidad. Hasta ahora, normalmente se han desarrollado aplicaciones específicas a raíz de proyectos de investigación, aunque todavía no han logrado una importante penetración en mercados más generales. Es de esperar que con la compatibilidad con internet que proporcionaría IoT este hecho cambie y se popularice más su uso.

El control y gestión de la energía en las redes de sensores es un factor muy importante y muy estudiado. Se ha afrontado desde diferentes perspectivas: desde el hardware, software, protocolos, etc. Aunque se echa en falta un estudio que englobe todos los aspectos anteriores para ofrecer una visión global y cuantifique cuál de ellas ofrecería más beneficio.

Un campo de investigación muy relevante para el futuro de la aplicabilidad de las redes de sensores es el de *energy harvesting*. Esta línea de investigación es muy activa y podría permitir reducir una de las mayores limitaciones de estas redes: la disponibilidad de energía.

### 2.6 Bibliografía

- [1] Kuorilehto, Mauri, Mikko Kohvakka, Jukka Suhonen, Panu Hämäläinen, Marko Hännikäinen, y Timo Hämäläinen. *Ultra-Low Energy Wireless Sensor Networks in Practice. Theory, realization and deployment*. West Sussex: John Wiley & Sons, 2007
- [2] Lajara, R. (2008). Revisión y análisis de los sistemas operativos existentes para redes de sensores basadas en dispositivos embebidos. Trabajo de investigación, Diploma de Estudios Avanzados. Universidad Politécnica de Valencia.
- [3] Ilyas, M., & Mahgoub, I. (Eds.). (2004). *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press.
- [4] Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., & Welsh, M. (2006). Deploying a wireless sensor network on an active volcano. *Internet Computing, IEEE*, 10(2), 18-25.
- [5] Yu, L., Wang, N., & Meng, X. (2005, September). Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on* (Vol. 2, pp. 1214-1217). IEEE.
- [6] Son, B., Her, Y. S., & Kim, J. (2006). A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. *International Journal of Computer Science and Network Security (IJCSNS)*, 6(9), 124-130.
- [7] Matese, A., Di Gennaro, S. F., Zaldej, A., Genesio, L., & Vaccari, F. P. (2009). A wireless sensor network for precision viticulture: The NAV system. *Computers and Electronics in Agriculture*, 69(1), 51-58.
- [8] McCulloch, J., McCarthy, P., Guru, S. M., Peng, W., Hugo, D., & Terhorst, A. (2008, April). Wireless sensor network deployment for water use efficiency in irrigation. In *Proceedings of the workshop on Real-world wireless sensor networks* (pp. 46-50). ACM.
- [9] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002, September). Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 88-97). ACM.
- [10] Lajara, R., Alberola, J., Pelegrí, J., Sogorb, T., & Vicente Llario, J. (2007, October). Ultra low power wireless weather station. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on* (pp. 469-474). IEEE.
- [11] Pister, K., Kahn, J., Boser, B. SmartDust Project. DARPA/MTO MEMS program. <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>
- [12] Wheeler, A. (2007). Commercial applications of wireless sensor networks using ZigBee. *Communications Magazine, IEEE*, 45(4), 70-77.
- [13] <http://www.zigbee.org/Standards/ZigBeeHomeAutomation/Overview.aspx>
- [14] Baker, C. R., Armijo, K., Belka, S., Benhabib, M., Bhargava, V., Burkhart, N., ... & Wright, P. K. (2007, May). Wireless sensor networks for home health care. In *Advanced*

- Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on* (Vol. 2, pp. 832-837). IEEE.
- [15] Wood, A., Virone, G., Doan, T., Cao, Q., Selavo, L., Wu, Y., ... & Stankovic, J. (2006). ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring. *University of Virginia Computer Science Department Technical Report, 2*.
- [16] Low, K. S., Win, W. N. N., & Er, M. J. (2005, November). Wireless sensor networks for industrial environments. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on* (Vol. 2, pp. 271-276). IEEE
- [17] King, J., BP pioneers large-scale use of wireless sensor. In *Computer World, mar 2005*.
- [18] Solano, W. M., Junell, J., Schmalzel, J. L., & Shumard, K. C. (2004). Implementation of wireless and intelligent sensor technologies in the propulsion test environment. In *Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE* (pp. 135-138). IEEE.
- [19] Car 2 Car Communication Consortium. (2007). *Braunschweig, November*. Available online <http://www.car-to-car.org/> (accessed on 17 2 2014)
- [20] Astute project (2011). Available online <http://www.astute-project.eu/> (accessed on 17 2 2014)
- [21] Sentilla. Available online <http://www.sentilla.com/> (accessed on 18 2 2014)
- [22] Advanticsys. Available online <http://www.advanticsys.com> (accessed on 18 2 2014)
- [23] Hue, personal wireless lighting. Available online <http://www.meethue.com/es-ES> (accessed on 17 2 2014)
- [24] Bokareva, T., Hu, W., Kanhere, S., Ristic, B., Gordon, N., Bessell, T., ... & Jha, S. (2006, October). Wireless sensor networks for battlefield surveillance. In *Proceedings of the land warfare conference*.
- [25] Puri, S. (2012). Intelligent Wireless Sensor Network System to shrink Suspected Terror from Militants. *International Journal of Computer Network & Information Security*, 4(3).
- [26] Lin, R., Wang, Z., & Sun, Y. (2004, June). Wireless sensor networks solutions for real time monitoring of nuclear power plant. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on* (Vol. 4, pp. 3663-3667). IEEE.
- [27] Pérez, E. (2009) Implementación de redes de sensores mediante LabVIEW. Proyecto final de carrera, Universidad Politécnica de Valencia.
- [28] Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *Wireless communications, IEEE*, 11(6), 6-28.
- [29] Stallings, W. (1997). *Sistemas operativos* (Vol. 732). Prentice Hall.
- [30] Dunkels, A., Gronvall, B., & Voigt, T. (2004, November). Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on* (pp. 455-462). IEEE.
- [31] Seah, W. K., Eu, Z. A., & Tan, H. P. (2009, May). Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on* (pp. 1-5). IEEE.

- [32] Taylor, G. W., Burns, J. R., Kammann, S. A., Powers, W. B., & Welsh, T. R. (2001). The energy harvesting eel: a small subsurface ocean/river power generator. *Oceanic Engineering, IEEE Journal of*, 26(4), 539-547.
- [33] Sogorb, T., Vicente Llario, J., Pelegri, J., Lajara, R., & Alberola, J. (2008, May). Studying the feasibility of energy harvesting from broadcast RF station for WSN. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE* (pp. 1360-1363). IEEE.
- [34] Penella, M. T., & Gasulla, M. (2007, May). A review of commercial energy harvesters for autonomous sensors. In *Proceedings of IEEE Instrumentation and Measurement Technology Conference Proceedings (IMTC 2007), Warsaw, Poland* (pp. 1-5).
- [35] Ivey, B. (2011). Low-Power Design Guide. Microchip application note 1416. Available online <http://ww1.microchip.com/downloads/en/AppNotes/01416a.pdf> (accessed on 25 2 2014)
- [36] Microchip Technology. Compiled Tips 'N Tricks Guide. Available online <http://ww1.microchip.com/downloads/en/DeviceDoc/01146B.pdf> (accessed on 26 2 2014)
- [37] Zhao, L., Zhang, W., Xu, C., Xu, Y., & Li, X. (2006). Energy-aware system design for wireless sensor network. *ACTA Automatica Sinica*, 32(6), 892.
- [38] Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2), 40-50.
- [39] Pottie, G. J., & Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5), 51-58.
- [40] Reddy, B. V., & Reddy, D. R. K. Maximizing the usable battery capacity in WSN with load harmonization. *International Journal of Instrumentation Electrical Electronics Engineering*, 2013.
- [41] Handziski, V., Polastre, J., Hauer, J., Sharp, C., Wolisz, A., & Culler, D. (2005, February). Flexible hardware abstraction for wireless sensor networks. In *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on* (pp. 145-157). IEEE.
- [42] Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537-568.
- [43] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1567-1576). IEEE.
- [44] Polastre, J., Hill, J., & Culler, D. (2004, November). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 95-107). ACM.
- [45] El-Hoiydi, A., Decotignie, J. D., Enz, C., & Le Roux, E. (2003, November). Wisemac, an ultra low power mac protocol for the wisenet wireless sensor network. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 302-303). ACM.
- [46] Li, M., & Yang, B. (2006, June). A Survey on Topology issues in Wireless Sensor Network. In *ICWN* (p. 503).

- [47]Ye, F., Zhong, G., Cheng, J., Lu, S., & Zhang, L. (2003, May). PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on* (pp. 28-37). IEEE.
- [48]Rodoplu, V., & Meng, T. H. (1999). Minimum energy mobile wireless networks. *Selected Areas in Communications, IEEE Journal on*, 17(8), 1333-1344.
- [49]Chen, M., Kwon, T., Yuan, Y., & Leung, V. (2006). Mobile agent based wireless sensor networks. *Journal of computers*, 1(1), 14-21.
- [50]Shah, R. C., & Rabaey, J. M. (2002, March). Energy aware routing for low energy ad hoc sensor networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE* (Vol. 1, pp. 350-355). IEEE.



# 3. ANÁLISIS DE CONSUMO DE ENERGÍA EN WSN

---

## 3.1 Introducción

Las redes de sensores inalámbricas (WSN) constan fundamentalmente de una serie de sensores/actuadores, un sistema procesador basado en microcontrolador, una fuente de alimentación y un sistema de comunicación inalámbrico. Usualmente se utilizan sistemas electrónicos ya fabricados llamados *motes*. Los *motes* consisten en pequeñas placas electrónicas autónomas que incluyen todas las partes de un nodo sensor, excepto —en algunos casos— algunos sensores o actuadores. El motivo de utilizar *motes* en lugar de diseños ad-hoc es la velocidad de desarrollo y la reutilización de código de terceros.

El software que ejecutan los *motes* tiene un rol fundamental en el desarrollo de las WSN, ya que controla la operación del nodo, implementa los protocolos de red y maneja los dispositivos controlando sus consumos de energía. Al igual que con los *motes*, también se suelen utilizar sistemas operativos como el software base sobre el cual desarrollar una aplicación de un nodo. Los motivos son los mismos que en el párrafo anterior: la velocidad de desarrollo y la reutilización de código de terceros.

Actualmente los sistemas operativos para redes de sensores más utilizados son TinyOS, Contiki y Mantis OS. El objetivo principal de estos sistemas operativos es proporcionar un mecanismo robusto y confiable de operación, manteniendo a la vez el menor consumo posible en cada parte del nodo y en cada momento. Reducir la potencia de cada parte del nodo consiste fundamentalmente en apagar o llevar a un modo de bajo consumo (*sleep*) al dispositivo cuando no se utiliza, mientras que cuando se utiliza se activa o despierta. Al reducir el consumo de cada parte del nodo, se reduce el consumo global y, por tanto, se extiende el tiempo de vida de la batería de los *motes*.

Una red de sensores puede considerarse como un sistema embebido con severas restricciones de memoria, capacidad de cómputo y consumo de energía. Tradicionalmente, el desarrollo de software para sistemas embebidos con recursos limitados ha sido el modelo de programación dirigida por eventos. TinyOS sigue el modelo de programación dirigida por eventos, consigue que el manejo de la energía del nodo sea muy eficiente y además una baja ocupación de la memoria de programa (*footprint*), finalmente su modelo de programación es similar al funcionamiento del hardware. Contiki es el otro gran sistema operativo para redes de sensores en la actualidad, ambos soportan IPv6 en sus pilas de protocolos en vistas al incremento de aplicaciones con conectividad a internet. Contiki también puede ser considerado un sistema

operativo conducido por eventos, pero incorpora abstracciones de programación para manejar la sincronización de tareas concurrentes y herramientas para facilitar la programación de alto nivel. Finalmente, a diferencia de los dos anteriores, Mantis OS es un sistema operativo multihilo. La principal característica de Mantis es la integración de un planificador multihilo y de abstracciones para crear, sincronizar y manejar los hilos concurrentes.

El objetivo de este capítulo es analizar y comparar el nivel de consumo de varios *motes* durante la ejecución de varias aplicaciones sobre cada uno de los sistemas operativos anteriores. Estas medidas revelan el estado general de consumo del *mote* en cada momento de la ejecución del programa a través de su consumo de corriente (como la alimentación de los nodos suele ser a tensión constante, se puede equiparar la potencia con el consumo de corriente). Otro efecto evaluado es el ruido que las operaciones del sistema operativo pueden generar sobre la fuente de alimentación durante el cambio continuo del estado energético de las diferentes partes del *mote*. Este ruido podría afectar a partes sensibles del *mote* como los sensores analógicos.

Para realizar la evaluación de la corriente consumida se ha utilizado un conjunto de ensayos compuesto por cuatro aplicaciones para cubrir las tareas típicas de un nodo de una WSN: planificación de eventos temporizados, muestreo de datos de sensores, procesado de datos y comunicaciones inalámbricas.

Este capítulo se divide en las siguientes secciones: en la sección 2 se presentan algunos artículos relacionados y sus resultados. La sección 3 está dedicada a presentar las características más importantes de los sistemas operativos comparados en este análisis. Los *motes* utilizados en los ensayos se analizan en la sección 4. La sección 5 se describe las aplicaciones que formarán parte de los ensayos y que serán programadas en cada sistema operativo y para cada *mote*. Finalmente la sección 6 ilustra las medidas, tanto los valores medios como los instantáneos, del consumo de corriente de cada *mote* durante la ejecución de cada tarea.

## 3.2 Estado del arte

En la bibliografía hay una gran cantidad de artículos concernientes a nuevos protocolos, algoritmos y sistemas operativos para redes de sensores inalámbricos. En la Tabla 3-1 se listan algunos de los sistemas operativos para redes de sensores. Esta lista es extensa y variada, cada sistema operativo ha sido desarrollado persiguiendo objetivos diferentes, por lo que consecuentemente presentan características también muy diferentes. Así, elegir el sistema operativo más apropiado para una aplicación en concreto es una tarea importante y no es sencilla, ya que hay una gran cantidad de opciones y muy pocas revisiones y comparaciones de los aspectos más prácticos de ellos. En particular, el manejo de potencia es una de las principales restricciones que tienen los desarrolladores de aplicaciones de WSN y será en la que se hará foco en este análisis. Por tanto, los resultados proporcionados por este artículo intentarán comparar el consumo de energía en aplicaciones reales en los principales sistemas operativos y en los *motes* más utilizados para poder elegir con criterio la opción más adecuada para una aplicación dada.

Tabla 3-1: Sistemas operativos para redes de sensores inalámbricas.

SO	Modelo	ROM (kB)	RAM (B)	Tipo de procesos
TinyOS v1	Eventos	3,4	336	Tareas, comandos y manejadores de eventos
TinyOS v2	Eventos	3,4	336	Tareas, comandos y manejadores de eventos
Contiki	Eventos	3,8	230	Protohilos
Mantis OS	Multihilo	14	500	Hilos
Nano-RK	Multihilo	10	2000	Tareas con prioridad
t-kernel	Multihilo	28,2	2000	Hilos
Bertha	Agentes móviles	10	1500	Fragmentos de procesos
CORMO	Eventos	5,5	130	Tareas y manejadores de eventos
SOS	Eventos	20	1163	Tareas definidas como módulos
SenOS	Máquinas de estados	No especificado		Procesos

Los sistemas guiados por eventos son aquellos en los que el paradigma de programación facilita el establecer acciones que respondan ante determinados sucesos. En el caso de microcontroladores, estos sucesos suelen provenir en último término de interrupciones. Este mecanismo tiene la ventaja de ejecutar únicamente aquel código necesario, el de respuesta al evento, por lo que el sobre coste en cuanto a cómputo, memoria y energía puede verse reducido. Por el contrario, presenta un paradigma de programación menos frecuente y, en algunos casos, una cierta complejidad de implementación. No obstante, las ventajas que ofrece los hacen bastante adecuados para redes de sensores, lo que explicaría el que sean mayoría en la tabla anterior.

Los sistemas multihilo heredan su concepto directamente de sistemas operativos para sistemas más potentes, tales como los de tiempo real o genéricos. Estos sistemas basan su ejecución en dividir una tarea en varias sub tareas, llamadas hilos, que se ejecutan concurrente. Sus ventajas son que se trata de un paradigma más extendido y estudiado, y el mejorar la eficiencia aprovechando los momentos en que el procesador pueda estar esperando datos de sistemas externos. Su principal desventaja es que su implementación también puede ser más compleja que otros sistemas, especialmente en sistemas expulsivos. Además, esa expulsión supone un trabajo extra que tiene que realizarse.

Los sistemas de agentes móviles son un paradigma que consiste en que el código del agente puede trasladarse de unos nodos a otros para ejecutarse. Entre las ventajas cabe destacar la robustez inherente al hecho de poder cambiar de nodo de ejecución si éste cae (si hay redundancia en la red). La principal desventaja en el ámbito de las redes de sensores es su necesidad de comunicación, siendo éste un aspecto especialmente consumidor de energía.

Las máquinas de estado son un paradigma para implementar tareas de forma metódica muy utilizado en sistemas embebidos y sistemas de control.

Hasta ahora ha habido pocos artículos publicados que incluyan una comparación de sistemas operativos en términos de consumo energético. El primero fue un artículo que comparaba

TinyOS y MantisOS, que puede verse en [1]. La principal contribución de este artículo es la evaluación del rendimiento del modelo conducido por eventos respecto a los sistemas multihilo en términos de consumo de energía y tiempo de ejecución, pero la evaluación no tiene en cuenta la interacción entre el hardware y el software y cómo el sistema operativo maneja las diferentes partes del mote usando las opciones de manejo de energía. Por otro lado, el artículo [2] propone una evaluación del tiempo de vida de la batería de un mote ejecutando algunas aplicaciones sobre diferentes sistemas operativos, pero no da información sobre el consumo instantáneo y su relación con el estado de consumo del mote. Finalmente, en el artículo [4] los autores identifican y miden el coste de operaciones básicas respecto al consumo total, pero no relacionan esta información con el contexto de una aplicación dentro de un sistema operativo real.

Además de medir el consumo de corriente instantáneo y medio, en este capítulo se tendrá en cuenta el ruido que el sistema operativo puede introducir en las líneas de alimentación de los motes durante su ejecución. Esta particularidad es especialmente importante porque puede afectar a la adquisición de sensores analógicos del mote si no ha sido tenido en cuenta, que es al fin y al cabo, la aplicación final del nodo. Otros investigadores han confirmado este riesgo [5] y sus efectos sobre los sensores son estudiados en [6].

## 3.3 Sistemas operativos

Esta sección proporciona un resumen de los sistemas operativos más importantes para redes de sensores. La atención se ha enfocado en cuatro de ellos, basándose la selección en parámetros tales como el número de publicaciones sobre ellos o la actividad de la comunidad de soporte. Respecto al número de publicaciones, se ha calculado el porcentaje de artículos relacionados con cada sistema operativo incluidos en las principales bases de datos online de temáticas relacionadas con la ciencia y la ingeniería hasta el año 2008. Las bases de datos han sido: IEEE Xplore, ACM Digital Library y Science Direct. Los porcentajes han sido: 81% TinyOS, 9% Contiki, 8% Mantis OS y 1% el resto. El soporte más activo por parte de la comunidad son los grupos de desarrolladores de TinyOS, con más de 10 nuevas liberaciones en una década, soporte para 12 plataformas diferentes y una conferencia anual sobre la tecnología de TinyOS. El grupo de Contiki ha realizado 7 publicaciones de código y ha formado un equipo formado por personas de compañías prestigiosas e instituciones de investigación. Como resultado del análisis anterior, se han seleccionado como los sistemas operativos más activos y aceptados: TinyOS Version 1.0 [8][9], TinyOS Version 2.0 [10], Contiki [14] y Mantis OS [11][12]. En principio también se incluyó en el análisis el sistema operativo SOS [13], pero finalmente se descartó debido a varios problemas para hacer que todos sus módulos fueran totalmente funcionales y al anuncio de los desarrolladores de que no se iba a dar más soporte en el futuro.

### 3.3.1 TinyOS version 1.0 (T1)

TinyOS fue el primer sistema operativo conducido por eventos específico para redes de sensores. Fue creado en la universidad de California (Berkeley) como una colección de componentes que implementan operaciones básicas y fue escrito en una variante del lenguaje C llamada NesC. Por tanto, se considera que TinyOS es un sistema operativo que utiliza la abstracción de componentes, los cuales son conectados con otros componentes por medio de interfaces. Se pueden programar rápidamente nuevas aplicaciones combinando componentes

y conectándolos entre sí por medio de los interfaces. Hay componentes de alto nivel que implementan protocolos, abstracciones de datos, estructuras de datos, servicios, etc. Como TinyOS es código abierto, los programadores pueden combinar y adaptar los componentes básicos proporcionados por el sistema operativo, personalizándolos para sus aplicaciones particulares. Únicamente los componentes utilizados son compilados e incluidos en el fichero final ejecutable, lo que supone una reducción significativa de la cantidad de memoria requerida.

TinyOS proporciona una funcionalidad robusta y confiable haciendo uso de reserva estática de memoria y un planificador FIFO no expulsivo. Los mecanismos de concurrencia implementados son los manejadores de interrupciones asociados al hardware. Cuando una interrupción ocurre, el microcontrolador 'salta' inmediatamente al manejador del evento correspondiente, parando la ejecución de la tarea activa en el instante previo. En TinyOS hay básicamente tres tipos de procedimientos:

1. Comandos, que son ejecutados inmediatamente después de su invocación. Están concebidos para realizar alguna acción sobre los elementos hardware del mote.
2. Manejadores de eventos, que interrumpen la ejecución de los comandos y las tareas después de ser activados por el hardware.
3. Tareas, que son funciones ejecutadas de forma normal, posiblemente de forma diferida.

Los comandos y manejadores de eventos constituyen los elementos que implementan la concurrencia de las tareas en el modelo en que TinyOS estructura los programas. Cuando el sistema intenta realizar una acción sobre componentes hardware, en primer lugar llama a un comando que envía la orden y, cuando acaba, devuelve el flujo de ejecución al sistema. Después de la configuración del hardware, realizada por medio de comandos, el mote puede ser llevado a un estado de bajo consumo mientras espera que el hardware emita alguna respuesta. La siguiente fase en la ejecución del programa ocurre cuando el hardware provoca una interrupción y el manejador de eventos recibe la señalización pertinente y ejecuta el código asociado. Este modelo de programación conducida por eventos proporciona concurrencia con un bajo sobrecoste en términos energéticos debido a que se adapta a la forma en que el hardware trabaja. Además, los componentes del mote pueden ser llevados a modos de bajo consumo durante sus periodos de inactividad.

El principal problema de esta metodología es la ausencia de mecanismos más complejos propios de los sistemas multihilos, lo cual hace difícil la implementación de secciones de exclusión mutua o el acceso a recursos compartidos. Por esta razón, el código normalmente se escribe utilizando máquinas de estados, a través de las que se implementa el flujo de ejecución y la sincronización para el acceso a los recursos compartidos. Sin embargo, este modelo de programación basado en máquinas de estados no favorece el desarrollo rápido de aplicaciones complejas. Además, la ausencia de estas herramientas utilizadas normalmente en aplicaciones concurrentes hace que sea más complicada establecer una secuencia adecuada de operaciones de alto nivel y propicia la aparición de problemas de condiciones de carrera o bloqueos entre tareas.

#### 3.3.2 TinyOS version 2.0 (T2)

La principal diferencia entre TinyOS v1.0 y TinyOS v2.0 desde el punto de vista del programador es la aparición en éste último de una nueva clase de abstracciones, llamadas componentes genéricos. En estos nuevos componentes pueden ser incluidos otros componentes, pero cada instancia es una nueva copia independiente de las demás. TinyOS v2.0 también mejora algunos aspectos relacionados con el soporte a las diferentes plataformas, fiabilidad de componentes básicos y las estructuras de datos. La secuencia de arranque también ha sido cambiada y puede ser bloqueada durante un tiempo para evitar problemas de condición de carrera durante la ejecución de tareas concurrentes.

#### 3.3.3 Contiki

Contiki se desarrolló en el Swedish Institute of Computer Science. Al igual que en el caso de TinyOS, Contiki también puede considerarse un sistema operativo conducido por eventos, pero con algunas particularidades que facilitan el desarrollo de nuevas aplicaciones con varias tareas concurrentes.

Una de las principales contribuciones de Contiki es el concepto de protohilos. Este mecanismo permite a los programadores realizar operaciones de bloqueo para que unos hilos esperen la activación de eventos desde otros hilos concurrentes. Los protohilos simplifican y reducen el número de máquinas de estados que serían necesarias para implementar una secuencia de operaciones de alto nivel. El sobrecoste de memoria introducido por los protohilos es muy bajo porque todos ellos comparten la misma pila, y además la conmutación entre hilos sólo supone un cambio en un puntero a dicha pila, lo que supone también un bajo coste en términos de cómputo. Por tanto, los protohilos combinan una buena eficiencia energética y una baja utilización de memoria dentro de un modelo de programación conducido por eventos con herramientas para realizar bloqueos y la facilidad de programación propia de los modelos basados en hilos.

Los programas en Contiki pueden ser distribuidos y ejecutados dinámicamente, además las últimas versiones incluyen una gran variedad de pilas de protocolos como uIP, SICSlowpan, Rime, etc.

#### 3.3.4 Mantis OS (MOS)

El último sistema operativo que es tenido en cuenta en este análisis es Mantis OS, desarrollado en la universidad de Colorado. Éste es un sistema operativo para redes de sensores que facilita la programación de nuevas aplicaciones con una aproximación completamente diferente. Mantis utiliza un planificador multihilo que permite conmutar entre tareas, por ejemplo entre tareas cortas con unos requisitos de tiempo muy estrictos, con otras tareas más complejas y de menos prioridad.

El planificador implementa un sistema *round robin* que incluye una cola de tareas listas para pasar a ejecución. El planificador utiliza un temporizador hardware que divide el tiempo de cómputo del procesador en segmentos (*slices*) y los asigna a las tareas en la cola. Durante cada segmento de tiempo se selecciona una tarea para ser ejecutada, mientras las demás permanecen en la cola esperando a otros segmentos. El planificador también tiene la capacidad de expulsión si ocurren eventos que deban ser atendidos inmediatamente. Este

modelo basado en hilos es utilizado en sistemas operativos modernos y de propósito general. Además proporciona herramientas para prevenir el bloqueo, condiciones de carrera y condiciones de inanición entre las tareas, entre otras incluye semáforos para implementar estructuras que permitan bloqueos en el acceso a recursos compartidos.

La posibilidad de tener diferentes tareas conmutando incrementa la cantidad de memoria RAM utilizada, además hace necesario realizar más operaciones, por ejemplo cambios de contexto, que suponen más consumo energético.

Mantis OS está programado en lenguaje C estándar, lo que implica una curva de aprendizaje menor y facilita la portabilidad de aplicaciones de terceros. En este análisis se ha utilizado la versión 1.0 beta.

### 3.4 Plataformas

En esta sección se presentan las principales características de las dos plataformas utilizadas en este análisis: Tmote Sky [15] y MICAz [16][17]. Estas plataformas son placas creadas con propósitos de investigación y experimentación, no tanto para aplicaciones profesionales; no obstante se han convertido en muy populares entre la comunidad de investigadores por la gran cantidad de software —principalmente de código abierto— desarrollado para ellas y la adaptabilidad a diferentes escenarios y casos de uso. Por ello, son los motes más utilizados en ensayos y en el desarrollo y validación de nuevos protocolos. La acogida de estas placas en entornos académicos e investigadores hace idóneo emplearlas para este análisis.

Ambos motes presentan una arquitectura muy similar (Tabla 3-2). Están basadas en un microcontrolador junto con un *transceiver* inalámbrico y algunos sensores. La principal diferencia es el tipo de microcontrolador: la Tmote Sky utiliza el TI MSP430F1611 [21] y la MICAz utiliza el Atmel ATmega128 [23]. El *transceiver* para las comunicaciones inalámbricas es el mismo en ambas plataformas: CC2420 [26] de Texas Instruments.

Estas dos plataformas eran suministradas por Moteiv y Crossbow, surgidas de Berkeley, pero ambas empresas reorientaron sus líneas de negocio y dejaron de proporcionar estos productos. En la actualidad, ambas plataformas se pueden adquirir en empresas como Memsic o AdvanticSys. No obstante, al tener diseños abiertos, existe documentación. Esta documentación puede encontrarse, por ejemplo, en la página web de TinyOS, y consiste sobre todo en sus circuitos esquemáticos y selección de componentes; esto permitiría realizar desarrollos propios o personalizados basados en los originales.

En la actualidad, además de estas plataformas, existen otras como Waspote -de la empresa Libelium-, z1 -de la empresa Zolertia- y distintos derivados de Arduino. De hecho hay motes estudiados en el capítulo anterior, como IRIS, que está basado en una modificación de MICAz y existen otros, como Shimmer, que consisten en una Tmote Sky adaptada a medidas biológicas corporales.

Tabla 3-2: Características principales de Tmote Sky y MICAz.

	Tmote Sky (Telosb)	MICAz
<b>Microcontrolador</b>	<b>Texas MSP430 F1611</b>	<b>Atmel ATmega128(L)</b>
Vcc	1,8 .. 3,6 V	2,7 .. 5,5 V
Consumo de corriente instantáneo	Activo 500 $\mu$ A @ 1 MHz, 3V Standby 2,6 $\mu$ A Off 0,2 $\mu$ A	Activo 5,5 mA @ 4MHz, 3V Power down 5 $\mu$ A
Tiempo en despertar	6 $\mu$ s	4,1 ms
Arquitectura	RISC 16 bits	RISC 8 bits
Flash	48 kB	128 kB
RAM	10 kB	4 kB
EPROM		4 kB
A/D	12 bits, 8 canales	10 bits, 8 canales
D/A	12 bits, 2 canales	
Comunicaciones	JTAG, 2xUART, 2xSPI, I2C, 3xDMA	JTAG, 2xUART, SPI, I2C
<b>Transceiver</b>	<b>CC2420</b>	
Vcc	2,1 ... 3,6 V	
Potencia de transmisión	0, -5, -10, -15, -25 dBm	
Sensitividad	-95 dBm	
Consumo de corriente instantáneo	RX 18,8 mA TX 17,4 mA (@ 0 dBm) sleep 426 $\mu$ A Power down 20 $\mu$ A off 0,02 $\mu$ A	
Tiempo en arrancar	1 ms (xtal oscillator)	
Radio de cobertura	Sobre 120 m en exterior con 0 dBm	
<b>Memoria Externa</b>	<b>ST M25P80</b>	<b>AT45DB041B</b>
Capacidad	Flash 1 MB	Flash 512 kB
Vcc	2,7 ... 3,6 V	2,5 ... 3,6 V
Consumo de corriente instantáneo	Read >4 mA Standby >50 $\mu$ A Power-down 1 $\mu$ A	Read 4 mA Standby 2 $\mu$ A
Interfaz	SPI	SPI
Sensores	Sensores integrados en la placa: humedad, temperatura y luz.	Conector de expansión con I/O digitales y analóg., I <sup>2</sup> C, SPI y UART. Placas de expansión con sensores de luz, temp., humedad, presión barométrica, acelerómetro, sensores acústicos y magnéticos.

### 3.4.1 Tmote Sky

La plataforma Tmote Sky también es conocida como Telosb. Esta dualidad de nombres es debido a que son fabricadas por dos empresas: Moteiv Corporation y Crossbow, que comparten el mismo diseño pero lo comercializan independientemente. Actualmente Moteiv Corporation ha cambiado de nombre por Sentilla y se ha enfocado en proyectos de manejo de energía para data centers, por lo que ha discontinuado su producto.

Los principales componentes de la plataforma Tmote son el microcontrolador Texas Instruments MSP430F1611 y el *transceiver* Texas Instruments CC2420. El MSP430 es un microcontrolador de muy bajo consumo, con 10 kB de memoria RAM y 48 kB de memoria de programa. Es un procesador de 16 bits con varios modos de bajo consumo y un modo más de *sleep* profundo. También cuenta con un oscilador digital (DCO) que implementa un reloj interno de 8 MHz. El microcontrolador puede despertarse desde el modo *sleep* en sólo 6  $\mu$ s, lo que permite un tiempo de reacción muy corto. El convertidor analógico a digital (ADC) puede tener una referencia interna y una velocidad de muestreo máxima de 200 ksamples por segundo.

El resto de periféricos disponibles incluyen un interfaz SPI, UARTs, temporizadores con función de comparación, dos convertidores digital a analógico (DAC) de 12 bits, supervisor de alimentación y tres controladores de DMA. El CC2420 está conectado al MSP430 a través del bus SPI. Otros periféricos integrados en la placa son una conexión USB mediante un *transceiver* FTDI, una memoria flash de 1 MB y un sensor digital Sensirion SHT15 para medir temperatura y humedad. La lista de los sistemas operativos soportados se encuentra al final de la Tabla 3-2.

### 3.4.2 MICAz

La segunda plataforma utilizada en este artículo es la MICAz. Este *mote* es un producto de Crossbow y la principal diferencia respecto a Tmote Sky es el microcontrolador, en este caso es un Atmel ATmega 128. Éste es un microcontrolador de 8 bits que es capaz de ejecutar cada instrucción en un único ciclo de reloj. El ATmega 128 tiene 128 kB de memoria de programa, 4 kB de EEPROM y 4 kB de SRAM, además de 53 pines de entrada/salida genéricos, cuatro temporizadores con función de comparación y modos PWM, dos UARTs, un bus I<sup>2</sup>C y otro SPI, un ADC de 10 bits con 8 canales multiplexados y un oscilador RC interno.

Esta plataforma admite placas de expansión conectadas a través de un conector, estas placas incluyen una gran variedad de sensores: luz, temperatura, presión barométrica, aceleración, sensores acústicos, magnéticos, etc. Los sistemas operativos soportados también se encuentra al final de la Tabla 3-2.

## 3.5 Aplicaciones

El objetivo de este capítulo es proporcionar un análisis y comparativa de las características de los sistemas operativos mencionados antes, proporcionando además resultados que puedan ser fácilmente reproducibles por otros investigadores. Como primera opción está la posibilidad de utilizar las aplicaciones que incluyen los propios sistemas operativos a modo de ejemplos. Sin embargo no ha sido posible encontrar un mismo conjunto de aplicaciones en todos los sistemas y plataformas. Por consiguiente, se ha tomado la decisión de realizar una serie de

aplicaciones específicamente para este ensayo y programarlas para cada sistema operativo y plataforma para realizar una comparación lo más equitativamente posible.

La primera aplicación es un programa trivial que no hace tarea alguna. Se ha llamado *Blank*. Con este programa se comprobará la forma en que el sistema operativo maneja el consumo cuando no hay tareas para ser ejecutadas.

Como hay sistemas operativos que no realizan un control automático del estado de funcionamiento respecto al consumo de potencia, se ha creado otra aplicación llamada *Blank2*, donde el código se optimiza para declarar explícitamente el paso a un modo de bajo consumo, por ejemplo en TinyOS se han utilizado los componentes `HPLPowerManagementM` y `McusSleepC`, mientras que en MOS se ha eliminado el efecto que provoca el interfaz USB en la placa Tmote Sky debido a un drenaje de corriente residual.

El siguiente programa es la típica aplicación *Blink*. Esta tarea consiste en la conmutación de un LED de forma periódica. Con este programa se evaluará el comportamiento del sistema operativo con tareas simples.

Otro programa evaluado es el llamado *XTEA*, el cual representa el caso opuesto al anterior, ya que su objetivo es evaluar una aplicación que involucre tareas con una importante carga computacional. El programa consiste en la implementación del algoritmo XTEA [22] para codificar palabras de 32 bits en un bucle que se repite 150.000 veces. Además se ha añadido una segunda tarea a esta aplicación para evaluar las capacidades multihilo del sistema operativo, que consiste en conmutar un LED.

Finalmente el último programa se ha llamado *Sens*. Este programa consiste en una aplicación típica de un nodo sensor. Consiste en la lectura de un sensor de temperatura cada segundo y la transmisión del valor leído inalámbricamente.

Al igual que en el primer caso, se ha realizado una variante del programa anterior, *Sens2*, que realiza un manejo explícito de los modos de bajo consumo. En estos programas hay que advertir que los sensores integrados en cada placa son diferentes, lo que tiene un impacto que se refleja en las medidas: MICAz utiliza un termistor de 10 k $\Omega$  mientras que Tmote Sky incluye un sensor SHT15.

Los núcleos de Contiki y MOS han integrado manejadores para controlar automáticamente los modos de bajo consumo de los motes, evitando así que el programador tenga que incluir explícitamente llamadas al sistema para pasar a estos modos. Por tanto, en estos dos sistemas operativos las aplicaciones *Blank2* y *Sens2*, en principio, no son necesarias. Sin embargo también hay que comentar que MOS no es capaz de controlar de forma totalmente automática el *transceiver* inalámbrico ni controla adecuadamente las líneas del *transceiver* USB en Tmote Sky. Por este motivo se han creado las aplicaciones *Blank2* y *Sens2* para Tmote Sky únicamente, en este caso *Blank2* configura adecuadamente las líneas del USB y *Sens2* además cambia explícitamente el estado del *transceiver* inalámbrico en los periodos en que está inactivo.

### 3.6 Metodología y materiales

En primer lugar se debe tener en cuenta que la tensión de los *motes* se mantiene relativamente constante a lo largo de todo su funcionamiento, por tanto el consumo de energía está directamente relacionado con el consumo de corriente. Por tanto se hablará de ambos como equivalentes. La consecuencia de esta suposición es que las medidas de energía consisten en medidas de corriente, la cual puede medirse más fácilmente.

Para medir la corriente hay varios métodos, por ejemplo se puede emplear una resistencia *shunt* conectada en serie con la fuente de alimentación del *mote*. También hay artículos donde se evalúa el tiempo de vida de las baterías; para ello el *mote* es alimentado inicialmente con baterías con un nivel máximo de carga y se mide el periodo que permanece en funcionamiento [2].

En este artículo se utilizará otro método para evaluar el consumo de energía del *mote*. Entre las premisas iniciales estaba poder realizar medidas instantáneas, lo que inhabilita el método de las baterías (además este método introduce la indeterminación de la carga inicial y otros factores). También se pretende realizar medidas con un alto grado de precisión.

El montaje experimental utiliza un SourceMeter que genera una tensión constante de 3 V y puede medir de forma muy precisa la corriente entregada a la carga, en este caso, el *mote*. El modelo es un Keithley 2400, que ofrece una resolución de hasta 1  $\mu$ A en el rango de 100 mA y una frecuencia de muestreo configurable dependiente de la línea eléctrica (50 Hz) de hasta 1200 sps [3]. Además, un programa realizado con LabVIEW se comunica con este equipo a través de GPIB para configurar la tensión de alimentación, la frecuencia de muestreo, automatizar y representar las medidas. Una vez que el SourceMeter comienza el proceso de medida, almacenará las muestras en una memoria interna cuya capacidad es de 2500 muestras. Cuando la memoria esté llena, el SourceMeter enviará los datos a un PC para representar y almacenar en un fichero los datos. También se calculará el valor medio y la varianza de la medida.

La ventaja de este método es la gran precisión obtenida debido a que es una medida bastante directa de un equipo también muy preciso. Por otro lado, la mayor limitación es la baja frecuencia de muestreo que permite el equipo; no obstante ésta es suficiente para casi todos los propósitos de este test. Sólo hay un caso en que se necesitará más frecuencia de muestreo, en este caso se utilizará una resistencia *shunt* seguida por un amplificador [20] conectado a un osciloscopio para medir la caída de tensión en la resistencia y, por tanto, la corriente que pasa a través de ella.

### 3.7 Resultados

Los resultados están enfocados a medir el consumo de corriente instantáneo y medio de los diferentes *motes* con los programas presentados en la sección 3.5. Los programas han sido compilados para MICAz y Tmote Sky, dando como resultados los ficheros binarios cuyas principales características se presentan en la Tabla 3-3. Respecto a esta tabla hay que indicar que Contiki no tenía soporte para MICAz en el momento de hacer la comparativa; también indicar que, como se comentó anteriormente, en MOS no es necesario programar la aplicación

*Blank2* para MICAz porque este sistema operativo gestiona automáticamente los modos de bajo consumo y no hace falta indicarlo explícitamente en el código del programa. Lo mismo ocurre en Contiki sobre Tmote Sky con la aplicación *Blank2*.

Tabla 3-3: Tamaño en memoria de los programas en bytes.

		Blank	Blank 2	Blink	XTEA	Sens	Sens 2
<b>MICAz</b>							
TOS	ROM	476	620	1674	1790	11402	11594
	RAM	19	21	48	64	441	443
T2	ROM	680	686	2218	2104	11890	13906
	RAM	4	4	51	49	278	331
MOS	ROM	26 k	-	26 k	27 k	30 k	30 k
	RAM	1 k	-	1 k	1 k	1,1 k	1,1 k
<b>Tmote Sky</b>							
TOS	ROM	1586	1586	2722	2858	13040	13203
	RAM	27	27	45	45	405	407
T2	ROM	1418	1430	2654	2656	12198	14068
	RAM	4	4	55	35	328	384
MOS	ROM	14 k	15 k	14 k	14 k	16 k	16 k
	RAM	1,6 k	1,6 k	1,6 k	1,6 k	1,7 k	1,7 k
Contiki	ROM	20,8k	-	20,9 k	21 k	-	21 k
	RAM	2,3 k	-	2.3 k	2,3 k	-	2,3 k

Como se aprecia en la tabla, las distintas versiones de TinyOS son las que menos memoria ocupan, tanto de datos (RAM) como de programa (ROM). La diferencia respecto a MOS y Contiki es de un orden de magnitud. Este resultado es el esperable ya que debido a su arquitectura interna y su paradigma de programación, TinyOS añade un sobrecoste mínimo respecto a la aplicación del usuario. Por otro lado, MOS y Contiki necesitan las funciones de soporte de los hilos, librerías del *kernel*, etc. Y esto se refleja, como se puede ver, en el *footprint* final de la aplicación. Este sobrecoste se puede extrapolar a través del tamaño de la aplicación *blank*, ya que es la que no incluye aplicación del usuario.

### 3.7.1 Resultados y discusión de las medidas de consumo

El consumo instantáneo de corriente de los *motes* para cada aplicación se muestra en la Figura 3-1, mientras que el consumo medio está representado en la Figura 3-2. A partir de estas gráficas se puede interpretar el estado en el que está el mote en cada momento.

Los resultados del programa *Blank* muestran que T2 tiene un consumo muy similar en ambas plataformas. El consumo de T2 es el mejor en el caso de MICAz y muy similar al de T1 en el caso de Tmote Sky. Lo más llamativo es que T1 para MICAz presenta un consumo de corriente mucho mayor en MICAz respecto a Tmote Sky. Esto es debido a la diferente forma que tiene el sistema operativo de manejar los modos de bajo consumo del microcontrolador. En el caso del ATmega128 sólo lo llevará a un modo de bajo consumo cuando se indique explícitamente utilizando la función `adjustPower()`. Mientras que en el MSP430 el planificador constantemente evalúa el estado para cambiar el modo de consumo y llevarlo al más bajo

posible que sea compatible con la aplicación [18]. El motivo de esta diferencia fundamentalmente es que el MSP430 es muy rápido despertando desde el modo *sleep*, mientras que el ATmega necesita varios ciclos de reloj; por ello la penalización por llevar a *sleep* el ATmega cuando no se debe puede ser alta, mientras que en el MSP430 sería insignificante. Por otro lado, MOS estima el modo de consumo cada vez que el planificador no tiene tareas a ejecutar. El problema en este caso es que el microcontrolador no puede entrar nunca en el estado más profundo de *sleep* porque siempre necesita dejar, al menos, un temporizador funcionando para lanzar las interrupciones que determinan los *slices* de tiempo y lanzar el planificador. Contiki presenta un comportamiento muy similar y la eficiencia de ambos es, en general, peor que las versiones de TinyOS.

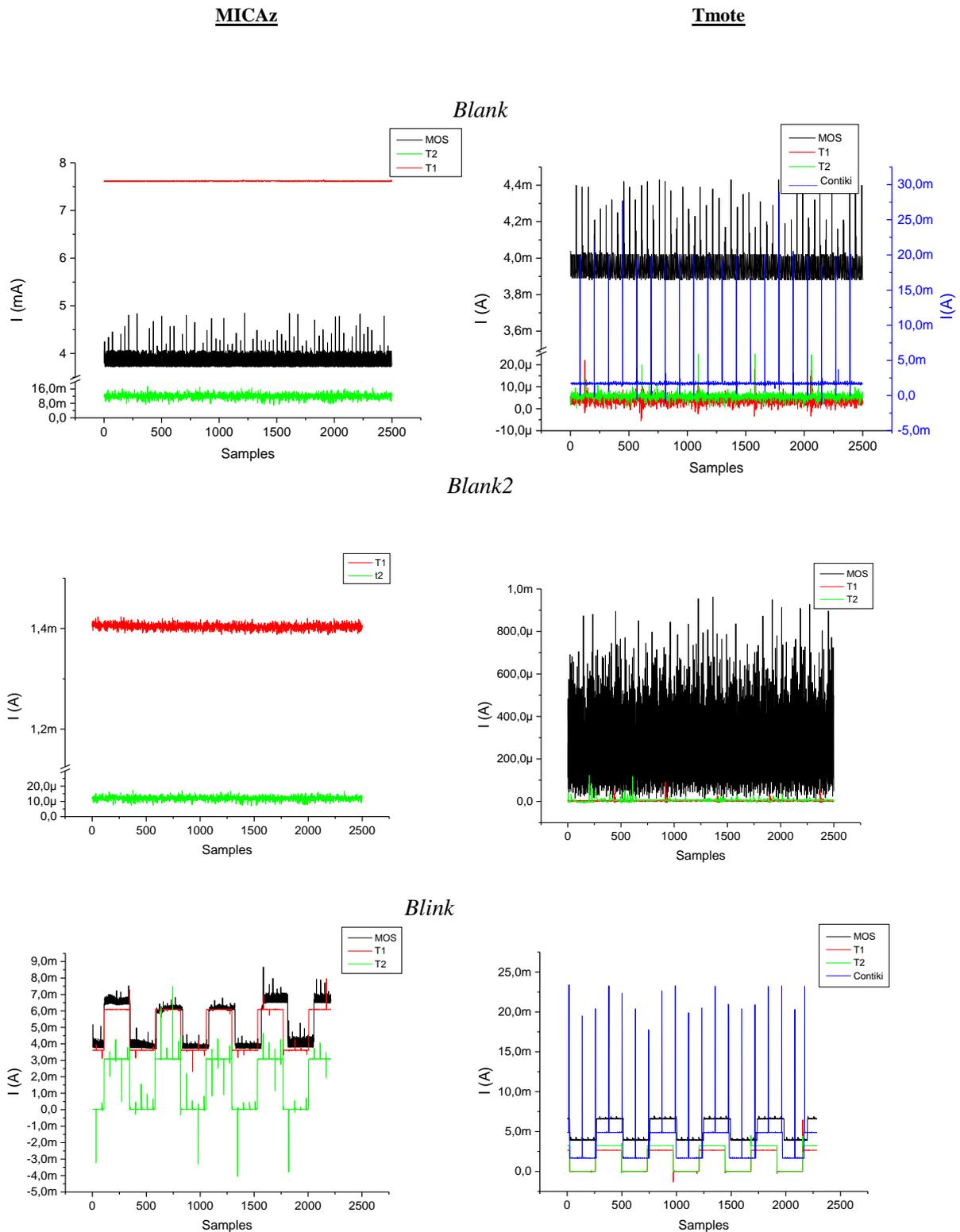
En el programa *Blank2* el consumo ha sido controlado explícitamente desde el código del programa llamando a los recursos propios de cada sistema operativo para manejar el modo de operación. Curiosamente, el sobrecoste que introduce T2 hace que el consumo de corriente se eleve ligeramente, de 10,45 a 11,99  $\mu\text{A}$  en MICAz y de 5,7 a 6,3  $\mu\text{A}$  en Tmote, en cualquier caso la variación no es muy significativa. El mismo comportamiento puede verse en T1 sobre Tmote Sky; no obstante, se aprecia una reducción apreciable del consumo de T1 sobre MICAz, de 7,62 a 1,4 mA porque el microcontrolador ya no permanece en modo activo durante el periodo de inactividad. En el caso de MOS también se ha apreciado una notable mejora, de 3,95 mA a 279  $\mu\text{A}$ , al controlar adecuadamente las líneas de la conexión USB de Tmote Sky [19].

Como se ve en la Figura 3-1, el programa *Blink* tiene básicamente dos niveles de consumo. Uno de ellos es equivalente al nivel de la aplicación *Blank*, mientras que el otro es igual a este nivel sumando el consumo de un LED a través de una resistencia serie de 470  $\Omega$ .

Los resultados del programa *XTEA* revelan que MOS es el único sistema operativo que realmente realiza una ejecución concurrente de las dos tareas de forma implícita. El planificador multihilo de MOS puede ejecutar efectivamente varias tareas 'en paralelo' desde el punto de vista del programador. En el resto de sistemas operativos la ejecución de las dos tareas es secuencial: primero el mote ejecuta el algoritmo *XTEA* como se ha comentado en la sección 3.5 y cuando acaba, se ejecuta *Blink*. Esto no quiere decir que la concurrencia no se pueda conseguir en los otros sistemas operativos, pero para conseguirla la tendría que realizar el programador, no es proporcionada por el sistema operativo. Para T1, T2 y Contiki hay dos partes que pueden ser distinguidas fácilmente en las gráficas de consumo instantáneo: la primera parte es una línea plana que representa el consumo durante la ejecución del algoritmo *XTEA* y la segunda parte tiene una forma de onda cuadrada que corresponde al proceso de parpadeo. Además este programa puede revelar otro parámetro: cuánto tiempo de procesado se necesita para realizar una determinada tarea; para ello simplemente hay que fijarse en el instante en que el consumo instantáneo tiene una abrupta caída. Se puede comprobar cómo el MSP430 necesita mucho más tiempo que el ATmega en finalizar la tarea. Ambos microcontroladores operan a la misma frecuencia de reloj, una hipótesis para explicar esta diferencia es el tipo de arquitectura interna de ambos núcleos. Atmel implementa una arquitectura Harvard de 8 bits mientras que Texas utiliza una von Newman de 16 bits; además el ATmega puede ejecutar una instrucción por ciclo de reloj, mientras que el MSP430 tiene un tiempo por instrucción que varía entre 1 y 6 ciclos de reloj. En la Figura 3-2 no se han mostrado los resultados del programa *XTEA* porque éste tiene dos fases: inicialización con procesado y

### 3. Análisis de consumo de energía en WSN

régimen estacionario; dado que el tiempo de inicialización es diferente, el periodo de tiempo de adquisición desvirtuaría una comparativa haciendo que los consumos sean más parecidos cuanto más tiempo se adquiriera ya que el inicio perdería relevancia. Los principales puntos de este programa simplemente sería el tiempo de cómputo y la concurrencia de aplicaciones, no tanto su consumo medio.



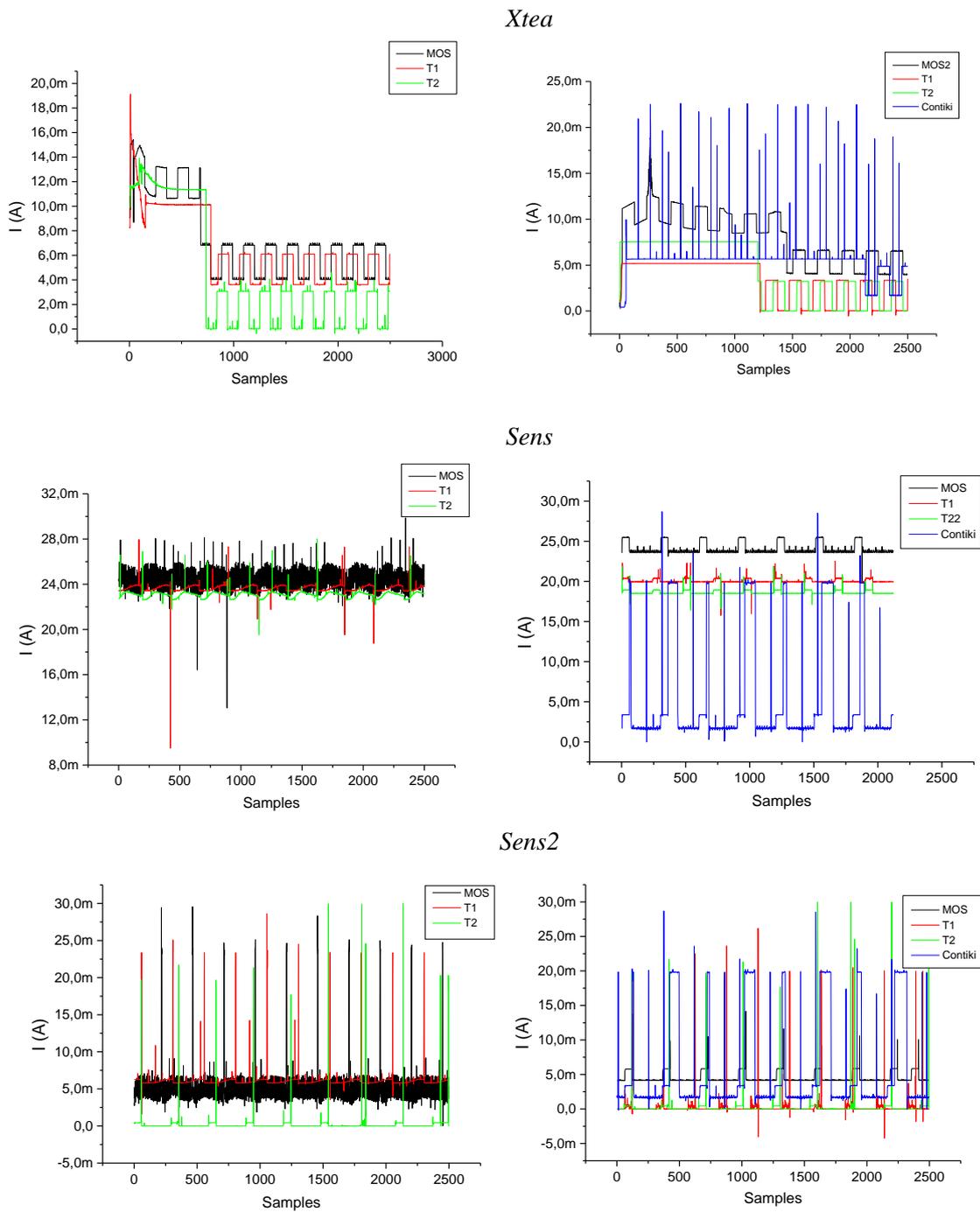


Figura 3-1: Consumo instantáneo de corriente para cada aplicación, sistema operativo y mote.

### 3. Análisis de consumo de energía en WSN

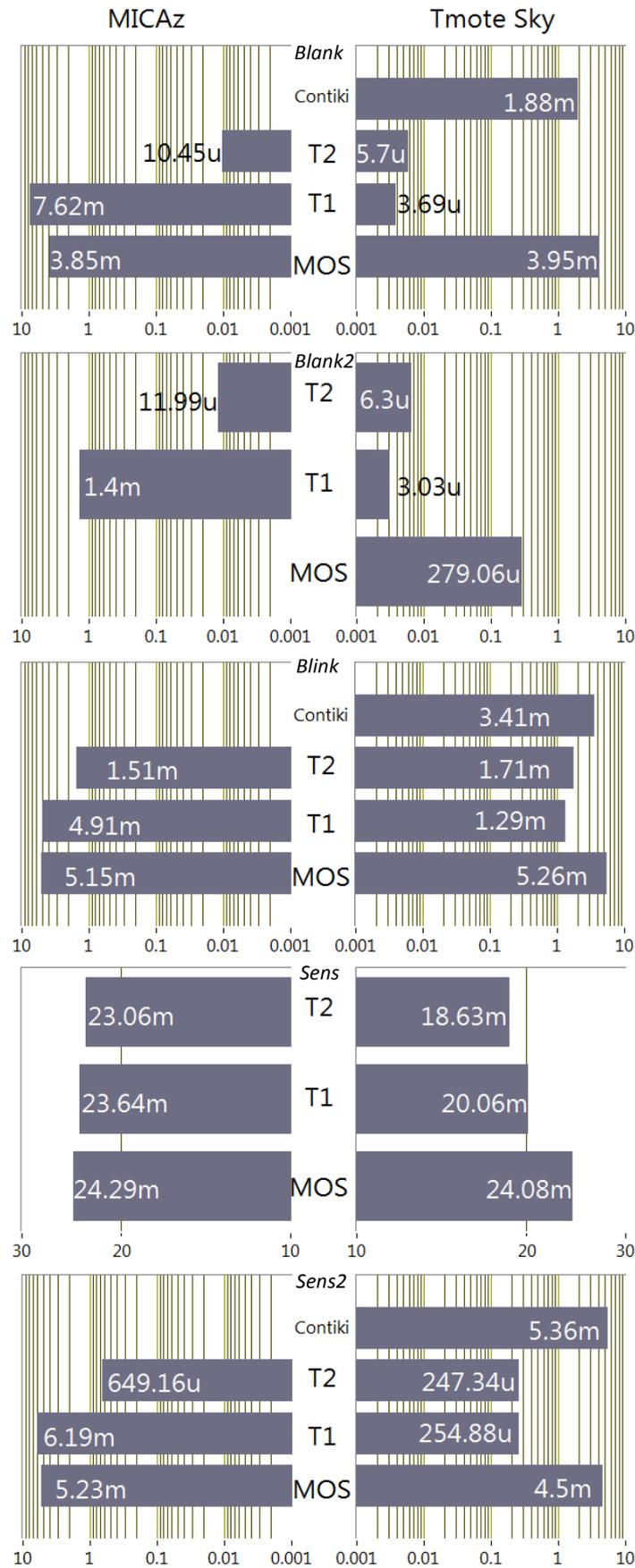


Figura 3-2: Consumo de corriente medio. Primera fila Blank; Segunda fila Blank2 (optimización de consumo); tercera fila Blink; cuarta fila Sens; quinta fila Sens2 (optimización de consumo).

En el caso de la aplicación *Sens*, el consumo de corriente sobrepasa los 23 mA en todos los sistemas operativos en MICAz y se sitúa entre 18 y 24 mA en Tmote Sky. El incremento está relacionado con el modo de operación que se establece para el *transceiver*. *Sens* no hace llamadas explícitas a funciones para el manejo de modos de consumo del microcontrolador o del *transceiver*, por lo tanto éste permanece activo todo el tiempo. De esta forma, de acuerdo con la Tabla 3-2 (obviando la frecuencia de reloj) la corriente debe ser: 500  $\mu$ A (microcontrolador en modo activo) + 17,4 mA (*transceiver* en modo recepción) = 18 mA para Tmote Sky y 5,5 mA + 17,4 mA = 23 mA para MICAz. Como puede verse en las gráficas, el consumo medio para todos los sistemas operativos es muy aproximado a la predicción teórica para MICAz: 23,06 mA para T2, 23,64 mA para T1 y 24,29 mA para MOS. Mientras que para Tmote Sky únicamente T2 se aproxima al valor teórico con un consumo de 18,63 mA, mientras que T1 y MOS añaden al valor esperado 1,5 y 3,5 mA respectivamente. En el incremento de MOS se encuentra la corriente consumida por el interfaz USB.

Finalmente, el último programa, *Sens2*, reduce drásticamente el consumo de corriente respecto al anterior, especialmente en T2. *Sens2* incluye llamadas a las funciones de manejo de potencia de cada sistema operativo, al igual que en *Blank2*. Contiki y MOS muestran un consumo mayor que las dos versiones de TinyOS.

### 3.7.2 Resultados y discusión de las medidas de ruido

Un dato interesante que se ha observado durante la realización de los ensayos ha sido el ruido que cada sistema operativo introduce en la fuente de alimentación del mote debido a los rápidos cambios en el estado y, por tanto en el consumo, de los diferentes elementos del hardware. Por este motivo se ha evaluado el nivel de ruido introducido por el mote en su propia alimentación al ejecutar los programas *Blank* y *Blank2*.

Para medir cuantitativamente el ruido se ha calculado la varianza de las 2500 muestras de cada una de las medidas de consumo instantáneo. En la Tabla 3-4 se muestran estas varianzas para cada sistema operativo y cada mote, donde se aprecia que Contiki es el que más ruido introduce, seguido por MOS.

Tabla 3-4: Varianza de las muestras de las medidas de consumo instantáneo (A<sup>2</sup>).

	Blank		Blank 2	
	MICAz	Tmote Sky	MICAz	Tmote Sky
<b>T1</b>	1,83E-12	2,1E-12	1,68E-12	63,75E-12
<b>T2</b>	20,8E-12	2,87E-12	29,06E-12	11,8E-12
<b>MOS</b>	22,2E-9	5,48E-9	-	35,41E-9
<b>Contiki</b>	-	3,4E-6	-	-

Tal y como se puede ver en las gráficas de la Figura 3-1, MOS genera un nivel de ruido muy significativo comparado con el resto de sistemas operativos.

Si se hace una observación más detallada del ruido, se podrá identificar un patrón que se repite periódicamente. La hipótesis sugerida es que se debe al planificador del sistema operativo. MOS configura un temporizador para que se desborde cada milisegundo, pero el planificador es llamado por defecto cada 20 desbordamientos, es decir, cada 20 milisegundos.

Con el sistema de medida anterior, basado en el SourceMeter, la frecuencia de muestreo no es suficientemente alta para determinar si el origen del ruido puede deberse al temporizador, al planificador o a algún otro elemento. Por tanto, se ha cambiado el método de medida por el otro procedimiento que se comentó en la sección 3.6 Metodología y materiales, el método que utiliza una resistencia *shunt*, amplificador y osciloscopio. Este método es más impreciso en cuanto a la resolución de la corriente medida, pero a cambio aumenta la frecuencia de muestreo de forma considerable. El resultado de la medida se muestra en la Figura 3-3. La señal presenta picos con una periodicidad de 1 ms.

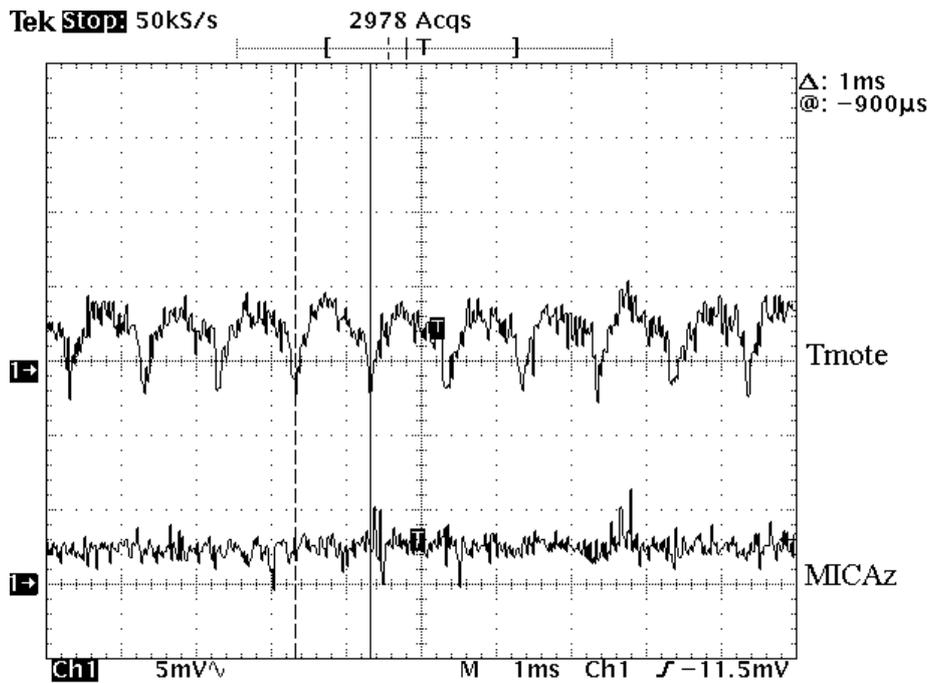


Figura 3-3: Medida de ruido en la línea de alimentación de Tmote Sky y MICAz con MOS.

Para comprobar la hipótesis de si el temporizador es realmente la causa de este ruido, se ha modificado el núcleo de MOS para cambiar el periodo de desbordamiento de 1 a 2 ms. Las mismas medidas con este nuevo tiempo muestran que los picos en la señal de alimentación también han cambiado a 2 ms, lo que confirma la hipótesis inicial.

Finalmente, una vez identificado el ruido falta comprobar si éste afecta y cuánto a las medidas de los sensores. Para realizar la evaluación se han usado los sensores propios de los motes, los de temperatura. En el caso de MICAz es un termistor, el cual es alimentado desde una línea digital del microcontrolador en lugar de desde la alimentación general del sistema. La medida fue realizada para encontrar si el ruido también estaba presente en la tensión de alimentación del sensor.

La gráfica para MICAz permite comprobar que no hay presencia de periodicidad en el ruido presente en la línea digital y éste es de bajo valor.

En el caso de Tmote Sky el caso no es diferente porque, aunque en la gráfica si se aprecia una periodicidad en el ruido y el nivel es mayor que en MICAz, el sensor digital SHT15 también está alimentado desde una línea digital del microcontrolador en la que el ruido es menor, y el caso sería comparable al de MICAz. Este ruido se ha comprobado que para el caso concreto de estas

dos placas con MOS y estos sensores no ha sido significativo, no obstante, debe tenerse en cuenta en nuevos motes o con otros sensores. Una posible explicación es que tanto en Tmote Sky como en MICAz se dispone de condensadores de 0,1  $\mu\text{F}$  en esta línea que pueden filtrar ruidos. En el caso de sensores que puedan ser sensibles a este ruido, se requeriría una etapa de acondicionamiento o amplificador, filtrados, o utilizar como en estos casos, pines digitales para alimentar si las demandas de corriente del sensor y de los pines del microcontrolador lo permiten.

### 3.8 Conclusiones

Después de desarrollar todas las aplicaciones mostradas en este artículo para diferentes sistemas operativos, se puede concluir que la programación en C implica una curva de aprendizaje mucho menor que cuando se realizan aplicaciones en NesC, como es el caso de TinyOS. En NesC el programador tiene que aprender un nuevo paradigma de programación que incluye conceptos como componentes, módulos, configuraciones, interfaces, etc. El lado positivo de la programación en TinyOS es la eficiencia que puede conseguir en términos de tamaño de código (ver Tabla 3-3) y consumo de energía.

El menor consumo en Tmote Sky es debido principalmente al microcontrolador que incorpora, cuyas características en este aspecto son superiores al Atmel. TinyOS 1 y 2 obtienen ventaja de este decremento de la energía consumida en Tmote Sky, mientras que en MOS hay algunas diferencias menos significativas entre MICAz y Tmote Sky; en cualquier caso, como se han mencionado en el capítulo, MOS podría reducir el consumo de la Tmote Sky en 3,5 mA deshabilitando el USB.

De acuerdo a los resultados presentados en el artículo, hay que indicar que en general TinyOS 2 es más eficiente que TinyOS 1 y éste más eficiente que MOS. En el caso de Tmote Sky las diferencias entre T1 y T2 son mínimas, aunque T1 es un poco más eficiente en aplicaciones simples. Por otra parte, se ha visto en el segundo apartado que T2 es más simple que T1 y, por supuesto que MOS con su planificador multihilo, mientras que Contiki se aproxima en uso de espacio de programa a TinyOS pero necesita mucha más RAM. Como era de esperar, en términos de eficiencia energética los sistemas más simples consumen menos energía. La gran cuestión entonces sería si la mejora en el consumo está justificada a cambio de una reducción de características más avanzadas. No hay una respuesta única a esta pregunta, ya que depende de cada aplicación en particular, si se está buscando primar la velocidad de desarrollo, mantenibilidad, etc. o se busca eficiencia. En la mayoría de las aplicaciones que se han desarrollado en este capítulo se ha optado por la segunda opción porque en las redes de sensores los investigadores suelen buscar el punto óptimo de operación.

El tiempo de inicialización requerido por el sistema operativo puede ser visto en las gráficas correspondientes a la aplicación *XTEA*. Este parámetro podría ser crítico en algunas aplicaciones que requieran respuestas rápidas tras una activación, pero no es tan importante si únicamente se considera su contribución al consumo, ya que este tiempo sería despreciable frente a la duración total de ejecución de una aplicación. Las diferencias entre los motes y los sistemas operativos han sido mínimas. Otra lección que se puede aprender de la aplicación *XTEA* es el tiempo en procesar datos. En el caso de la Tmote Sky tarda casi el doble del tiempo

que la MICAz. Esta diferencia en el tiempo de cómputo puede ser explicada por la desigualdad en la arquitectura interna de los microcontroladores y el número de ciclos en ejecutar cada instrucción, como se mencionó en la sección 6.

Finalmente, un punto importante a considerar es la perturbación que una aplicación puede producir en la alimentación del mote. En MOS se puede ver que existe un ruido, mostrado en la Figura 3-3 y la Tabla 3-4. Para evaluar si este ruido supone un riesgo, se debe considerar que el objetivo final de los motes es usualmente sensor variables físicas a través de los sensores que integran. Los sensores requieren un circuito de acondicionamiento, el cual puede estar alimentado por la misma línea que el microcontrolador, por tanto el ruido medido en la línea de alimentación puede afectar también a la estabilidad de las medidas de los sensores. La pregunta sería si realmente este ruido afecta y, en caso afirmativo cuánto. Las arquitecturas utilizadas incluyen sensores de temperatura, en MICAz hay un termistor que es alimentado a través de un pin del microcontrolador. De acuerdo a la Figura 3-3 no se aprecia ruido significativo en la línea de alimentación en esta arquitectura (canal inferior), por lo tanto tampoco afectará a la medida. En la Tmote Sky se puede ver en la Figura 3-3 (canal superior) que el ruido en este caso sí es significativo, lo que confirma el efecto observado para otras arquitecturas compuestas por un PIC16 y un interfaz directo al sensor [6]. En Tmote Sky el sensor de temperatura es el SHT71 de Sensirion, el cual también está alimentado a través de una línea digital del microcontrolador e incluye un condensador de 0,1  $\mu$ F. Se ha realizado una medida como las anteriores en esta línea y no se ha visto la propagación de este ruido desde la línea de alimentación del microcontrolador hasta la línea digital que alimenta al sensor. Este hecho nos hace suponer que tampoco se espera un efecto significativo para la Tmote Sky y su sensor de temperatura y humedad. Sin embargo es una cuestión que puede ser importante en algunos casos y debe tenerse en cuenta durante el desarrollo de nuevos motes o placas de sensores para éstas, ya que pueden darse casos que sí presenten efectos significativos, especialmente en el caso de sensores que requieran un acondicionamiento de señal con amplificación o sensores de interfaz directa. En estos casos es recomendable filtrar la línea de alimentación de los sensores y, en general, separar las alimentaciones para evitar estos efectos.

## 3.9 Bibliografía

- [1] Cormac Duffy and Utz Roedig and John Herbert and Cormac Sreenan: An Experimental Comparison of Event Driven and Multi-Threaded Sensor Node Operating Systems. International Workshop on Sensor Networks and Systems for Pervasive Computing 2007.
- [2] Healy, Michael, Thomas Newe, Elfed Lewis: Power Management in Operating Systems for Wireless Sensor Nodes. IEEE Sensors Applications Symposium. San Diego, California USA, 2007. 1-6.
- [3] Keithley. Keithley 2400 datasheet. (online document, available in <http://www.keithley.com/products/dcac/voltagesource/broadpurpose/?mn=2400>)
- [4] Antonopoulos, C., Prayati, A., Stoyanova, T., Koulamas, C., & Papadopoulos, G. (2009, May). Experimental evaluation of a WSN platform power consumption. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on* (pp. 1-8). IEEE.

- [5] Reverter, F. Pallas-Areny, R. Uncertainty reduction techniques in microcontroller-based time measurements, *Sensors and Actuators A: Physical*. Volume 127, Issue 1, 28 February 2006, Pages 74-79
- [6] Reverter, F. Gasulla, M. Pallas-Areny, R. Analysis of Power-Supply Interference Effects on Direct Sensor-to-Microcontroller Interfaces. *Instrumentation and Measurement, IEEE Transactions on*. Feb. 2007, Volume: 56, Issue: 1, Pages 171-177.
- [7] Kuorilehto, Mauri, Mikko Kohvakka, Jukka Suhonen, Panu Hämäläinen, Marko Hännikäinen, Timo Hämäläinen: Ultra-Low Energy Wireless Sensor Networks in Practice. Theory, realization and deployment. West Sussex: John Wiley & Sons, 2007.
- [8] Gay, David, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, David Culler. The nesC Language: A Holistic Approach to Networked Embedded Systems. *Proceedings of Programming Language Design and Implementation (PLDI) 2003*. San Diego, California, USA: ACM Press New York, 2003. 1-11.
- [9] Handziski, Vlado, Joseph Polastre, Jan Hauer, Cory Sharp, Adam Wolisz, David Culler. Flexible Hardware Abstraction for Wireless Sensor Nodes. *Proceedings of the Second European Workshop on Wireless Sensor Networks, 2005*. 2005. 145-157.
- [10] Levis, P. and Gay, D. and Handziski, V. and others. T2: A second generation OS for embedded sensor networks. Telecommunication Networks Group, Technische Universität Berlin, Technical Report TKN-05-007, 2005
- [11] Abrach, H., et al. MANTIS: System Support for Multimodal Networks of In-situ Sensors. *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*. San Diego, CA, USA: ACM Press New York, 2003. 50-59.
- [12] Bhatti, Shah, et al. MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mobile Networks and Applications* 10, nº 4 (august 2005): 563 - 579.
- [13] Shea, Roy, Chih-Chieh Han, y Ram Rengaswamy. Motivations Behind SOS. Networked Embedded Systems Lab, University of California Los Angeles, Los Angeles, CA, 2004.
- [14] Dunkels, Adam, Björn Grönvall, Thiemo Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. *Proceedings of the First IEEE Workshop on Embedded Networked Sensors*, November 2004.
- [15] Crossbow Technology Inc. TelosB. datasheet, (online document, available in <http://www.xbow.com/>).
- [16] Stan, Alexandru: Porting the Core of the Contiki Operating System to the TelosB and MicaZ Platforms. International University Bremen. May 7, 2007
- [17] Crossbow Technology Inc. MICAz. datasheet, (online document, available in <http://www.xbow.com/>).
- [18] Szewczyk, Levis, Turon, Nachman, Buonadonna, Handziski. Microcontroller Power Management. TinyOS 2 documentation. (online document, available in <http://www.tinyos.net/tinyos-2.x/doc/txt/tep112.txt>)
- [19] UCAR University, General Info about MOS. (online document, available in [https://www.eol.ucar.edu/isf/facilities/isa/internal/MOS\\_hints/MOSnotes.html](https://www.eol.ucar.edu/isf/facilities/isa/internal/MOS_hints/MOSnotes.html))
- [20] Microchip Technology Inc. MCP6041/2/3/4 datasheet. (online document, available in [www.microchip.com](http://www.microchip.com)).
- [21] Texas Instrument Inc. Datasheet MSP430. (online document, available in <http://www.ti.com/lit/ds/symlink/msp430f1611.pdf>).
- [22] Needham, R. M., & Wheeler, D. J. (1997). Tea extensions.
- [23] Atmel. Datasheet ATmega. (online document, available in <http://www.atmel.com/Images/doc2467.pdf>).
- [24] Eswaran, A., Rowe, A., & Rajkumar, R. (2005, December). Nano-rk: an energy-aware resource-centric rtos for sensor networks. In *Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International* (pp. 10-pp). IEEE.

- [25] Cha, H., Choi, S., Jung, I., Kim, H., Shin, H., Yoo, J., & Yoon, C. (2007, April). RETOS: resilient, expandable, and threaded operating system for wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (pp. 148-157). IEEE.
- [26] Texas Instruments. Datasheet CC2420. (online document, available in <http://www.ti.com/lit/ds/symlink/cc2420.pdf>)

# 4. MODELADO

---

## 4.1 Introducción

Las redes de sensores son un campo de investigación en el que se construye una red de nodos comunicados inalámbricamente. Los nodos suelen ser pequeñas placas electrónicas, autónomos y reubicables. La función típica de los nodos es sensar información y transmitirla.

Los nodos de las redes de sensores tienen numerosas restricciones. Una de ellas es la energía, ya que son nodos portables, autónomos y potencialmente desatendidos. Por ello, es de vital importancia conocer y optimizar la gestión de su energía disponible.

Las fuentes de energía de las redes de sensores han sido tradicionalmente dos: baterías y recolección de energía. En la primera de ellas se utilizan baterías alcalinas o de litio para alimentar un nodo. En la segunda hay gran diversidad: recolección de energía solar, viento, generadores piezoeléctricos, obtención de energía de RF del ambiente, etc. También pueden encontrarse sistemas mixtos donde se encuentren estas dos fuentes [1]. En todos los casos hace falta un mecanismo de almacenamiento de energía, ya sea una batería recargable o un supercondensador.

La optimización del consumo energético se ha abordado desde diferentes perspectivas, en gran medida complementarias. Una de las perspectivas es la optimización de protocolos. Otra es la optimización de los componentes semiconductores y de comunicaciones. También la optimización de los algoritmos que ejecuta el propio nodo. Desde algunas investigaciones se señala que donde se obtiene más ahorro energético se puede obtener es optimizando el consumo en el dispositivo antes que en los protocolos de comunicaciones [2].

En este contexto, algunos investigadores apuntan que es necesario desarrollar modelos de baterías y descarga de baterías precisos para poder realizar mejores estimaciones de la capacidad de la batería y, por tanto, mejoras en la gestión del consumo y un consiguiente aumento de la esperanza de vida de la red. Esta estimación puede ser esencial para establecer un adecuado ciclo de trabajo de la aplicación y/o de la capa MAC, tomar decisiones de encaminado de paquetes, etc. Este modelado todavía es un problema abierto y hay muy pocos trabajos enfocados a las redes de sensores.

El objetivo principal de este capítulo es idear un método para la construcción de diferentes modelos que permitan estimar mejor el nivel y consumo de energía en un nodo sensor. También se propondrán diversos modelos, cada uno con diferentes características de precisión y carga de procesado.

El método se aplicará a baterías de litio por ser ampliamente utilizadas en este tipo de aplicaciones; no obstante sería aplicable a otros sistemas como supercondensadores, aunque la mayor linealidad entre carga y tensión de éstos los hace un poco menos atractivos.

La organización del capítulo es: sección 2 para el estado del arte, donde se ve una introducción de las formas de gestión de energía. En la sección 3 se describen los pasos necesarios para construir los modelos. El hardware y software involucrado se presentan en la sección 4 y, posteriormente, se presentan los resultados experimentales. En la sección 5 se utilizan las medidas anteriores para estudiar diversos algoritmos de aprendizaje automático para modelar el sistema. En la sección 6 se validará el modelo anterior utilizando nuevos datos. Finalmente, en la última sección se exponen las conclusiones del estudio y el trabajo futuro.

### 4.2 Estado del arte

Existen distintos métodos para modelar el consumo en redes de sensores inalámbricas. Algunos se centran en modelar el *transceiver* [3][4][5], otros tienen en cuenta también el microcontrolador y los sensores [6]. Estos modelos pueden servir para optimizar los protocolos que trabajan sobre las redes de sensores [2][3][7][8] y los algoritmos que ejecutan los nodos [9][10].

Los nodos de las redes de sensores inalámbricas habitualmente tienen un esquema de trabajo basado en estados. El nodo pasa por diversos estados de forma periódica, normalmente un estado para sensar, otro para transmitir (ambos se puede englobar en un macroestado llamado despierto) y otro estado dormido. El consumo en cada estado es diferente, siendo el mínimo en modo dormido; en este estado se intentará estar la mayor parte del tiempo. Ha habido varias propuestas para modelar el consumo de estos estados y de las transiciones entre ellos [11][12].

Estos modelos de consumo pueden mejorar otras áreas de investigación. Por ejemplo se ha observado que los simuladores adolecen de buenos modelos [2], lo que puede afectar negativamente a muchas investigaciones sobre protocolos y arquitecturas [11]. Además, el modelo del consumo se puede aplicar para estimar la esperanza de vida de una red o nodo, o para tomar decisiones que afecten al trabajo de la red, como por ejemplo cambiando las rutas del tráfico [13]. Otra aplicación es detectar o predecir cuándo sería necesario un mantenimiento para cambiar las baterías. Sin embargo, los modelos, por definición, son simplificaciones de la realidad; la mayoría no tienen en cuenta factores que pueden cobrar importancia con el paso del tiempo, como por ejemplo el desgaste que sufren las baterías con el tiempo. Si, por ejemplo, un nodo debe tomar una decisión en base a la energía que le queda y esta medida no es precisa, la decisión puede ser incorrecta.

Un método para estimar el estado de la batería es el presentado en [14], donde se modela en consumo de un nodo de una red de sensores mediante el consumo medio de cada uno de sus estados. El estado de carga (SoC) es estimado exclusivamente por software, en el cual se lleva un registro de la corriente consumida en cada periodo de tiempo. Este modelo está basado en el concepto de contador de culombios. No obstante, presenta algunas desventajas, como la

acumulación de errores, necesidad de calibración para cada aplicación, tener que conocer el estado inicial de la batería, etc.

La forma más habitual de estimar el tiempo de vida de un nodo sensor es medir la tensión de la batería [15]. No obstante, la tensión de las baterías de litio durante su descarga no es lineal, teniendo una caída muy abrupta en sus últimos instantes. Esto se debe a que durante la descarga de una batería de litio típica hay influencia de varios tipos de polarización; durante la fase de polarización óhmica hay un comportamiento bastante lineal, pero después se pasa a una fase llamada '*concentration polatization*' donde se produce el fuerte descenso de tensión [16]. Otros modelos utilizados en dispositivos hardware, en lugar de medir la tensión, se basan en medir la corriente consumida, pero al basarse en una integración, los errores se acumulan con el tiempo [17]. Un ejemplo de esta aproximación es el presentado en [18]. Este método obtiene el estado de la batería utilizando un modelo basado en una aproximación polinómica, cuyos coeficientes se obtienen de un ajuste de varias medidas de la descarga de la batería. Para evitar la dependencia de la carga, se realiza un segundo ajuste para obtener otra aproximación polinómica de los coeficientes de la primera aproximación. Sin embargo este modelo se ve afectado por el consumo irregular de corriente.

Otros modelos analíticos intentan proporcionar una representación de la batería utilizando redes neuronales combinadas con filtros de Kalman extendidos [19]. Desafortunadamente su implementación es bastante compleja.

Por otra parte, el envejecimiento de las baterías se debe a la modificación de la estructura interna de los electrodos por la pérdida de material activo por disolución [20]. Este desgaste se debe a varios factores, por ejemplo la edad, temperatura de almacenaje y porcentaje de carga y el número de ciclos de carga/descarga. Incluso los ciclos de trabajos pulsados pueden reducir su capacidad [10]. Esto se traduce en una reducción de la capacidad nominal de la batería con el paso del tiempo, por lo que las medidas basadas en un modelo pueden ser correctas en un momento dado, pero incorrectas en otro momento.

La mayoría de modelos permiten estimar el tiempo de vida de un nodo respecto a la capacidad nominal de la batería. Estos modelos se pueden complementar con otros que tengan en cuenta el envejecimiento. Tradicionalmente se ha utilizado la ley de Peukert para tener en cuenta el desgaste, no obstante esta fórmula se ideó para baterías de plomo-ácido y no es directamente aplicable en todas las situaciones para las baterías de litio [21].

Otros métodos más completos tienen en cuenta la temperatura, edad, resistencia interna y la carga para estimar el desgaste, y de ahí obtienen el estado de carga [22][23][24]. Todas las baterías tienen una impedancia interna. La resistencia interna, tanto en AC como en DC, es uno de los parámetros más interesantes para estimar la pérdida de capacidad por envejecimiento. Esta resistencia es la suma de una resistencia eléctrica debida a los materiales y contactos y una resistencia iónica debida a factores electroquímicos como la conductividad de los electrolitos [25]. Hay diversos métodos de medida, los más sencillos miden la parte resistiva y los más completos tienen también en cuenta la reactancia. Los métodos más precisos requieren una instrumentación específica y realizar ensayos, por lo que no serían adecuados para realizarse '*in vivo*'. Un método sencillo para estimar la resistencia interna de las baterías es el que se ha utilizado con baterías para vehículos eléctricos [26], este método

consiste en aplicar distintos niveles de corriente y medir la tensión en cada una de ellas, la tensión de la batería se decrementará a mayor demanda de corriente [27]. Así la resistencia se puede calcular como:

$$R = \frac{V_2 - V_1}{I_2 - I_1} \quad (4.1)$$

Este método se ha utilizado incluso para aplicarlo a supercondensadores [28]. Tiene algunas limitaciones como no considerar la parte reactiva y falta de linealidad con la pérdida de capacidad [29], pero a pesar de ello se sigue utilizando ampliamente por ser muy sencillo de aplicar. Este parámetro se debe medir con la batería en circuito abierto, por lo que medir la resistencia de la misma batería que está alimentando el circuito tendrá errores debido al efecto de carga.

### 4.3 Metodología

En este capítulo se seguirá una secuencia de pasos para realizar completamente la generación y validación de los modelos de descarga de baterías y de predicción del SoC. Estos pasos se describen a continuación:

1. **Medidas.** En este paso se realizará la adquisición de datos experimentales durante la operación normal de un nodo sensor. La aplicación consistirá en adquisición de datos, por ejemplo tensión, corrientes y temperaturas. Después se enviarán estos datos a la estación base. Las sesiones comenzarán con la batería totalmente cargada y finalizará cuando el nodo deja de enviar datos. Cada sesión tendrá unas condiciones particulares, definidas con una temperatura y número de ciclos de carga/descarga de la batería.
2. **Modelado.** En primer lugar se filtrarán los datos para reducir el ruido. Después se calcularán nuevos parámetros a partir de los datos originales. También se introducirán como parámetros los datos de las condiciones de las medidas (temperatura y número de ciclos). El SoC es calculado para cada muestra utilizando el tiempo que queda hasta finalizar la sesión y la duración total de la sesión; por tanto el SoC será el porcentaje de tiempo que queda para que la batería se agote. Finalmente se seleccionan unos datos representativos de cada sesión y se combinan para ser utilizados en la generación de los modelos. En la generación de los modelos se descartarán los parámetros que proporcionen menos información y se probarán varios algoritmos de minería de datos para predecir el SoC.
3. **Validación.** En esta sección se probarán los modelos generados antes. En primer lugar se presentarán los resultados de los modelos utilizándose sobre los datos sobre los cuales se han generado. Sin embargo estos modelos podrían predecir muy bien los datos con los que han sido generados pero no con datos nuevos, por ello se realizarán nuevas sesiones experimentales con condiciones diferentes y se probarán los modelos anteriores con estos nuevos datos.

En el primer anexo se muestra un diagrama de flujo detallado de este proceso, indicando los programas, ficheros y plataformas involucradas en cada caso.

## 4.4 Medidas

En esta sección se describe el montaje experimental para tomar una serie de medidas que serán la base para desarrollar el modelo de descarga de las baterías en los nodos de una red de sensores. El hardware recopilará los parámetros más relevantes que permitan predecir su comportamiento. Los datos incluirán muestras en diferentes condiciones. Para variar una de estas condiciones, el número de ciclos de carga y descarga, se construirá un sistema de desgaste de baterías para acelerar este proceso.

### 4.4.1 Setup

#### 4.4.1.1 Hardware

En esta sección se describirá cómo se realiza la adquisición de la tensión y corriente de la batería conectada en el nodo sensor.

Las baterías utilizadas son PRT-11316 de 40 mAh [30] de litio-polímero. Se ha elegido este tipo porque su baja capacidad permite realizar las medidas rápidamente, pero tienen suficiente capacidad para permitir el funcionamiento normal del nodo. Así se podrá validar el método de modelado, el cual posteriormente se podrá aplicar con otras baterías.

En nodo de la red de sensores es un Tmote Sky [31]. Esta placa cuenta con un microcontrolador TI MSP430F1611 a 8 MHz, un *transceiver* TI CC2420, interfaz USB, 3 LEDs, memoria y sensores de temperatura, humedad y luminosidad. Los LEDs cuentan con unas resistencias de 470, 220 y 100  $\Omega$ . Teniendo en cuenta la caída de tensión de cada uno y considerando una tensión de activación de 3,7 V generan unas corrientes teóricas de 4,3, 7,3 y 16 mA respectivamente. El microcontrolador tiene una arquitectura de 16 bits, con varios modos de bajo consumo y un *sleep* profundo. En modo activo consume 500  $\mu$ A a 1MHz y en modo *standby* 2,6  $\mu$ A. Además cuenta con un ADC de 12 bits y 8 canales.

El *transceiver* [26] funciona a 2,4 GHz y 250 kbps, consumiendo 18,8 mA en recepción, 17,4 mA en transmisión. 426  $\mu$ A en *sleep*, 20  $\mu$ A en *power down* y 0,02  $\mu$ A apagado.

El sensor de temperatura utilizado es el Sensirion SHT11 [33]. Tiene un convertidor de 14 bits, una resolución de 0,01  $^{\circ}$ C y una precisión de  $\pm 0,4$   $^{\circ}$ C. El consumo medio es de 90  $\mu$ W, con un modo de *sleep* de 2  $\mu$ W y de 3 mW cuando realiza una medida.

Se ha añadido el circuito mostrado en la Figura 4-1 para realizar medidas sobre la batería. En primer lugar se ha colocado una resistencia *shunt* ( $R_{shunt}$ ) para medir la corriente que demanda el nodo. La tensión en uno de los terminales de la resistencia es la presente en los terminales de alimentación del microcontrolador y el resto de elementos en el nodo. Esta tensión se mide conectando una referencia de tensión de 2,5V en el ADC del microcontrolador y dividiendo por dos internamente la tensión de alimentación. En el siguiente apartado se explicará más extensamente este punto.

La tensión en el otro terminal se mide a través de un divisor de resistencias externo ( $R$  y  $R$  en la figura); el valor de las resistencias del divisor se pueden elegir llegando a un compromiso entre el aumento de corriente que representan y la impedancia que puede ver el ADC, si las resistencias fueran muy grandes habría que utilizar un amplificador operacional como seguidor

antes del ADC. En este caso, teniendo en cuenta el valor de impedancia de entrada del ADC en el datasheet del microcontrolador, se ha elegido un valor de 1 k $\Omega$ . El valor de la resistencia *shunt* en nuestro caso es de 32  $\Omega$ . Para obtener la corriente simplemente se aplica la ley de Ohm.

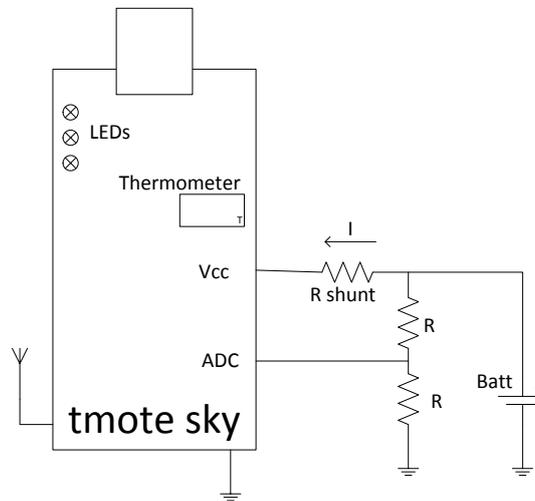


Figura 4-1: Esquema de medición de la tensión y corriente de la batería.

#### 4.4.1.2 Software

El programa que ejecuta el nodo es una aplicación típica de los nodos finales de las redes de sensores. Se realizará una tarea de medida periódica, el resultado de esta medida se enviará de forma inalámbrica a un receptor y el sistema se dormirá.

El nodo Tmote Sky lleva incorporado el sistema operativo TinyOS [34] para gestionar los recursos. Este sistema operativo es ampliamente utilizado en redes de sensores, y su combinación con Tmote Sky permite realizar aplicaciones de muy bajo consumo [35]. Tiene un flujo de ejecución basado en eventos y un lenguaje de programación propio llamado NesC.

El periodo de muestreo es de 1 segundo. El programa, una vez despertado, realizará la siguiente secuencia:

1. Activación del microcontrolador. Medida del nivel de tensión de la batería y del *shunt* ( $V_{low}$  y  $I_{low}$ ).
2. Medida de temperatura (Temp).
3. Activación del *transceiver* y construcción de un paquete de datos.
4. Envío de datos.
5. Activación de los LEDs.
6. Medida del nivel de tensión de la batería y del *shunt* ( $V_{high}$  y  $I_{high}$ ).
7. Desactivación de LEDs y apagado del *transceiver* y microcontrolador.

Hay dos tipos de medidas analógicas diferentes:

- La medida de alimentación ( $V_{cc}$ ) se realiza con el componente de TinyOS `VoltageC`. El mecanismo para hacer la medida consiste en configurar el microcontrolador para dividir la tensión de alimentación por 2 y utilizar una referencia interna en el ADC. Todo esto se realiza internamente en el microcontrolador. Por defecto TinyOS utiliza

una referencia interna de 1,5 V en el MSP430, pero se ha cambiado por una de 2,5 V para ampliar el rango de tensiones.

- La tensión de la batería ( $V_{batt}$ ) se mide a través del divisor de resistencias que se ha añadido externamente. Esta es una medida normal del ADC, la única peculiaridad es que es corregida respecto a la tensión de alimentación ( $V_{cc}$ ) en ese instante.

Las muestras de  $V_{high}$  e  $I_{high}$  correspondientes a un instante, se envían en el paquete de datos de la siguiente muestra.

En la Figura 4-2 se puede ver que existe una variación de corriente entre los estados *low* y *high*. La medida corresponde a la tensión en la resistencia *shunt* de la línea de alimentación de la Tmote Sky utilizando un osciloscopio. En estado *low* hay un consumo de 625  $\mu$ A durante 272 ms, mientras que en estado *high* hay un consumo de 28 mA durante 28 ms aproximadamente (a estos consumos habría que sumar el del divisor de tensión de la figura anterior), esta corriente se correspondería principalmente con el *transceiver*, sumada al microcontrolador y LEDs.

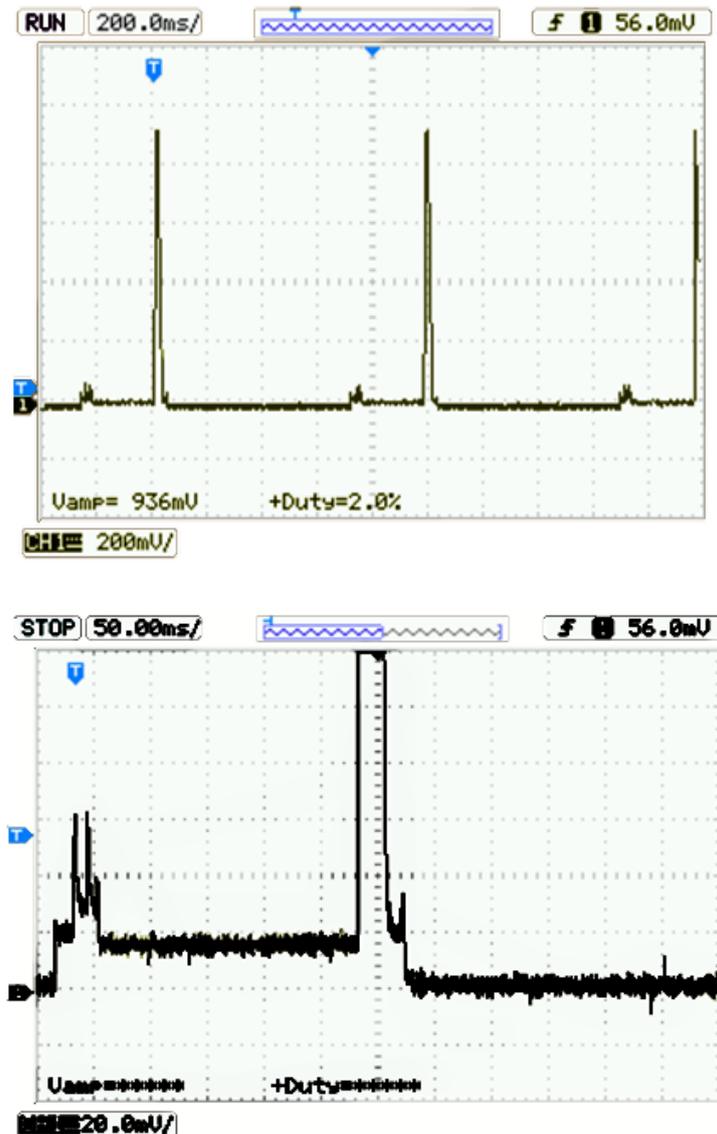


Figura 4-2: Consumo de corriente de la Tmote Sky. Resistencia *shunt* de 32  $\Omega$ ; abajo: detalle de la gráfica.

En la recepción se utiliza otra placa igual con el programa `BaseStation` para transmitir por puerto serie los datos recibidos. Esta placa está conectada a un PC, donde un programa LabVIEW recibe los datos y realiza los cálculos para ajustar los valores de corriente y tensión. Todos estos parámetros se guardan en un fichero en formato CSV.

La medida acaba cuando el nodo emisor se queda sin energía para transmitir. En ese momento se calcula el porcentaje de tiempo respecto a la duración total de cada muestra —SoC— y se añade al fichero. Así, el SoC representa el porcentaje de carga útil que le queda a la batería, siendo del 100% cuando está cargada (utilizando la capacidad real, no la nominal) y del 0% cuando ya no puede entregar la suficiente energía al nodo para mantener en funcionamiento la aplicación.

Una vez que se almacenan los datos, se les aplicará un preprocesado y se calculará una serie de parámetros derivados de estas medidas, como se explicará más adelante. Aunque estos parámetros se calculan en el PC, en una implementación posterior en fase de aplicación podrán ser calculados por un microcontrolador.

El número de ciclos de carga/descarga de la batería durante la fase de modelado se puede determinar mediante el banco de stress que se expondrá en la siguiente sección. En la fase de aplicación de los modelos se podrá calcular desde el microcontrolador monitorizando la tensión de carga de la batería como se explica en [23], cuyo diagrama simplificado se muestra en la Figura 4-3.

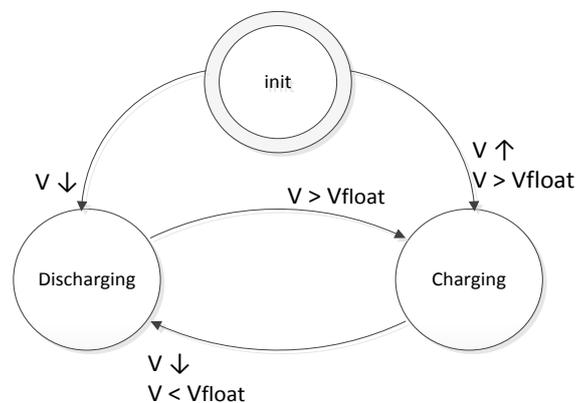


Figura 4-3: Máquina de estados para monitorizar el estado de la batería.

Este método consiste en monitorizar constantemente la batería. Por una parte se captura su nivel de tensión ( $V$ ) y por otra, usando las variaciones de este nivel, se determina su pendiente ( $V \uparrow$  ó  $V \downarrow$ ).

El nivel de tensión se utiliza para comprobar si está en la fase de sobretensión, fase «Charging» en la figura. Esto se produce cuando  $V > V_{float}$ , siendo  $V_{float}$  la tensión de la batería en circuito abierto, esto es, la tensión nominal cuando está cargada. Si esta condición se produce, significa que se está produciendo una carga de la batería. Típicamente, si la tensión nominal de la batería es de 3,6 V, por tanto un valor superior a 4 V significará que el cargador de la batería está funcionando, inyectándole corriente.

Por otro lado, cuando la tensión de la batería es igual o menor a la tensión nominal, se podrá detectar la fase de descarga, «Discharging», comprobando que la tensión en la batería tiene pendiente negativa, es decir, desciende.

Cuando se completa un ciclo de carga y descarga se aumentará el contador del número de ciclos y se podrán hacer los cálculos correspondientes.

En [23] se distingue entre los distintos estados que forman el proceso de carga, por lo que el algoritmo propuesto es más complejo, pero si no es necesario distinguir entre estos estados, el mecanismo se podrá simplificar por el mostrado.

Este sistema tiene algunas limitaciones como la detección de ciclos de carga o descarga completos. Esto se discutirá en la siguiente sección y al finalizar el capítulo, viendo cuánto de importante será, aunque en cualquier caso se puede limitar su aparición evitando recargas cortas, por ejemplo de la forma que se explicará en el capítulo 5.

#### 4.4.1.3 Banco de stress

Se ha desarrollado un sistema para realizar un desgaste de las baterías de forma automática, mostrado en la Figura 4-4. El desgaste consiste en aplicar ciclos de carga y descarga de forma periódica a las baterías.

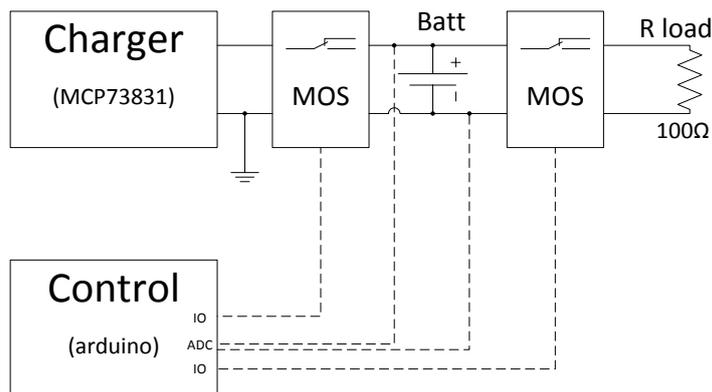


Figura 4-4: Sistema de desgaste automático de baterías.

El sistema consta de un cargador de baterías LiPo MCP73831 [36]. Este cargador se encarga de regular la tensión, la carga tiene tres fases: *bulk*, *absorption* y *float*. La fase *bulk* es una fase inicial y muy rápida donde se eleva la tensión de la batería hasta un valor por encima de la tensión en circuito abierto, es decir, si la batería a plena carga y sin nada conectado suele tener 3,7 V ( $V_{float}$ ), la tensión al final de la fase de *bulk* es de unos 4,2 V; además en esta fase la inyección de corriente es constante y de nivel máximo. La fase de *absorption* es la que más tiempo dura, en ella se mantiene esta sobretensión constante mientras se va regulando poco a poco la corriente. Finalmente en la fase *float* se detiene la sobretensión y la batería mantiene su tensión nominal hasta que se conecte a una carga. Se puede resumir en una primera fase de corriente contante, otra de sobretensión constante y una final en circuito abierto.

La corriente durante la fase de carga de la batería se ha configurado respecto al valor de la corriente nominal de la batería seleccionada, en este caso se ha limitado a un máximo de 40 mAh.

La resistencia de carga simulará el consumo de un nodo sensor, pero se ha aumentado la demanda de corriente para acelerar el proceso. El desgaste, entre otras variables, también depende del ratio de descarga [37], pero en este modelo se simplificará suponiendo desgaste constante por simplificación y porque, como se indica en la referencia anterior, esa dependencia apenas es apreciable. Se realizará una descarga de hasta el valor nominal de la batería, es decir a 1C, por lo que el valor máximo de corriente será de 40 mA, con lo que el valor de la resistencia será de 100  $\Omega$ .

Además, el sistema consta de dos *switches* de tipo MOS en cada lado de la batería para controlar cuándo se debe cargar y cuándo descargar.

Un sistema de control abrirá y cerrará los *switches* y monitorizará la tensión de la batería. El control comenzará detectando el estado de la batería: cargada ó descargada. Después comenzará a realizar un ciclo determinado de cargas y descargas. Cuando la tensión ha descendido por debajo de 2,7 V se considera que se ha descargado y cuando está por encima de 3,7 V y además se desactiva la señal STAT del cargador (que indica el final de la carga) se considera que está cargado. El sistema de control puede controlar hasta tres bancos de desgaste independientes. Cada uno de ellos es implementado mediante una máquina de estados finitos diferente.

Además se ha utilizado una tarjeta DAQ mediante entradas analógicas diferenciales para monitorizar la evolución de la tensión en las baterías del banco. Se puede ver un ejemplo en la Figura 4-5.

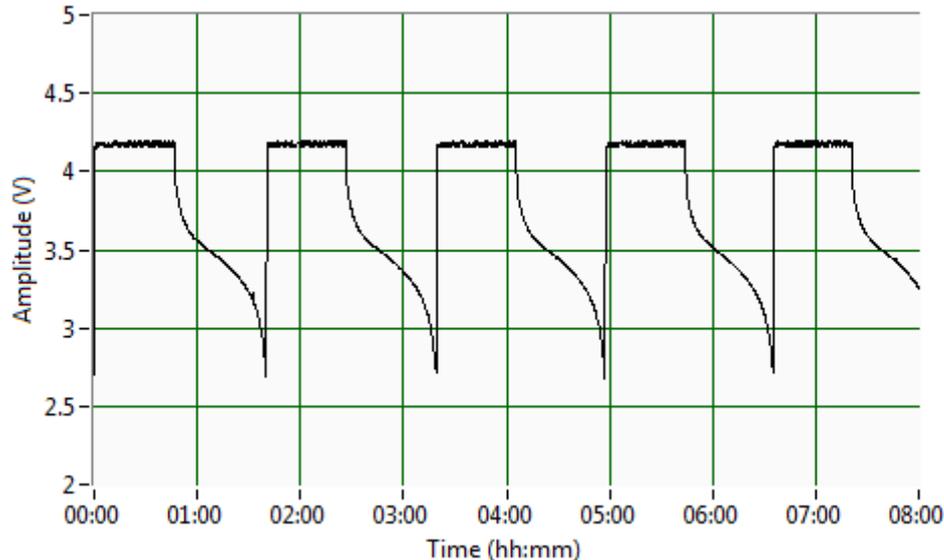


Figura 4-5: Evolución de la tensión de una batería en el banco de stress.

El circuito esquemático se muestra en el anexo 10.2.1 de esta tesis.

### 4.4.2 Resultados de las medidas

En esta sección se describirá cómo se han realizado las medidas que servirán de base para crear el modelo y el resultado de las mismas.

Se han hecho medidas utilizando dos Tmote Sky aleatoriamente, a diferentes temperaturas y con baterías con diferentes desgastes. Las medidas realizadas se han hecho con:

- Desgaste de batería:
  - Batería nueva: 0-10 ciclos de carga/descarga.
  - Batería con desgaste moderado: 40-50 ciclos de carga/descarga.
  - Batería con desgaste elevado: 200-220 ciclos de carga /descarga.
- Temperatura:
  - Baja temperatura: 0-5 °C.
  - Temperatura ambiente: 20-30 °C.

Los ciclos de desgaste de las baterías se fijan a priori mediante el banco de stress de la Figura 4-4 y es un dato conocido y constante para cada traza. La temperatura se consigue en una sala regulada mediante termostato y se emplea el propio sensor del nodo para medirla.

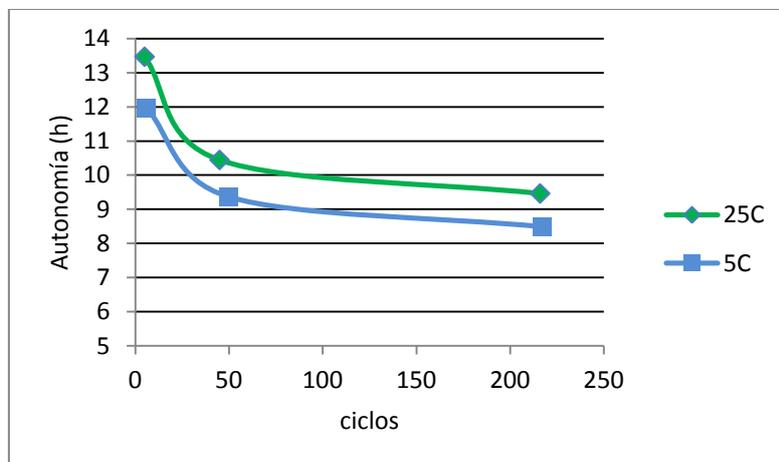
En cada una de las condiciones anteriores se han tomados 6 medidas distintas de las que posteriormente se seleccionarán 4, consistentes en la captura de los niveles de tensión y corriente de la batería y la temperatura ambiente en cada momento. Para ello un nodo sensor estaba conectado a la batería a testear y recogía las medidas mediante el circuito de la Figura 4-1. Otro nodo recibía los datos y los enviaba al PC.

Todas las medidas se han hecho en el canal 16 del CC2420 (2,48 GHz), con una potencia en el emisor de 0 dBm y una distancia entre el emisor y receptor entre 1 y 2 metros.

Las medidas se resumen en la Tabla 4-1 y la representación de la evolución se muestra en la gráfica de la Figura 4-6.

**Tabla 4-1: Resumen de autonomía (horas) de las medidas para cada desgaste y temperatura. Media  $\pm$  desviación estándar (error típico).**

Ciclos carga/descarga	Temp. baja (0-5 °C)	Temp. ambiente (20-30 °C)
Batería nueva (0-10 ciclos)	11,96 $\pm$ 1,15 (0,436)	13,47 $\pm$ 1,45 (0,593)
Desgaste medio (40-50)	9,4736 $\pm$ 1,02 (0,415)	10,44 $\pm$ 0,59 (0,242)
Batería muy desgastada (200-220)	8,49 $\pm$ 1,43 (0,584)	8,91 $\pm$ 1,36 (0,554)



**Figura 4-6: Evolución de las horas de autonomía respecto al número de ciclos en las baterías.**

#### 4. Modelado

Es interesante comprobar cómo a raíz de las medidas hechas para el tipo de batería elegido, la autonomía parece tener un comportamiento logarítmico respecto al número de ciclos. Esta información servirá para utilizar como parámetro el logaritmo de los ciclos en los algoritmos de aprendizaje.

Como referencia para ver la mejora respecto al método tradicional consistente en aplicar una ecuación lineal [38], se ha obtenido la recta que mejor se ajusta a los valores extremos de la descarga de una batería nueva a temperatura ambiente. Este es el método más habitual para estimar el estado de carga en sistemas embebidos. Para ello se mide la tensión inicial ( $V_{low}$ ) de la batería como  $3,94 \pm 0,04$  V y la tensión final  $2,53 \pm 0$  V. Con estos datos la ecuación es:

$$SoC(\%) = 70,92 \cdot V - 179,43 \quad (4.2)$$

De las medidas realizadas se elegirán algunas de ellas para realizar los modelos. La representación de la tensión  $V_{low}$  de estas medidas se puede ver en la Figura 4-7, donde también se han añadido histogramas para ver la dispersión y media de las medidas. En la Figura 4-8 se muestra la tensión leída en la batería en estado *low* como variable independiente respecto al SoC ideal como variable dependiente; esta forma de onda es la que se pretende linealizar a través de los diferentes modelos.

Como se puede apreciar, las medidas presentan cierta dispersión, de ahí la superposición de trazas de diferentes ciclos, no obstante, en los histogramas se aprecia cómo las medianas de las medidas son un buen predictor del desgaste. Debido a esta dispersión, cada medida se ha repetido varias veces

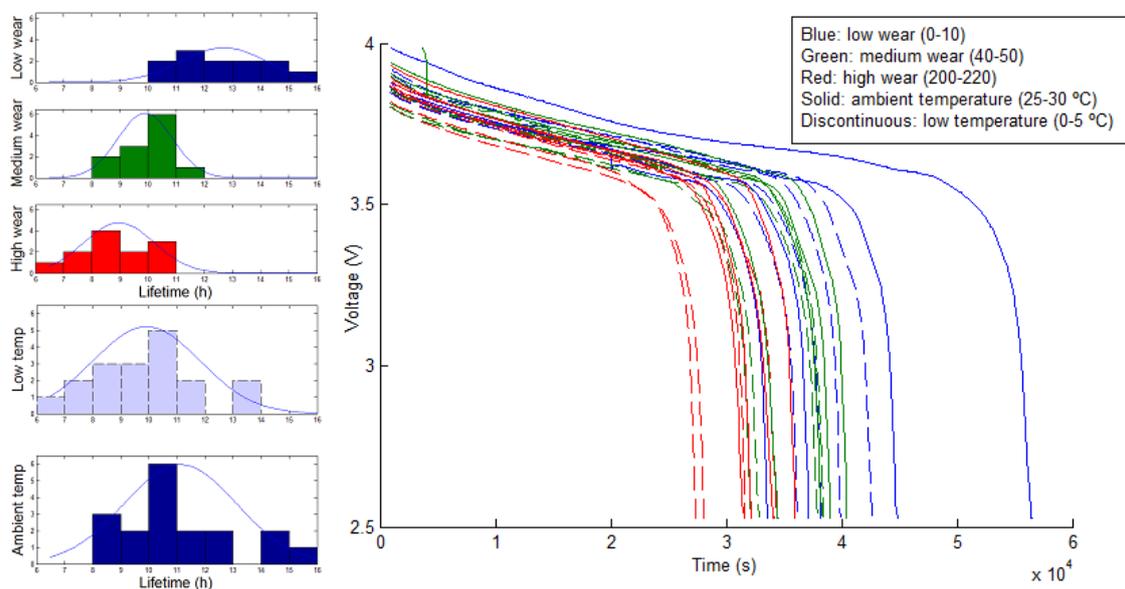


Figura 4-7: Curvas de la tensión de descarga de las medidas.

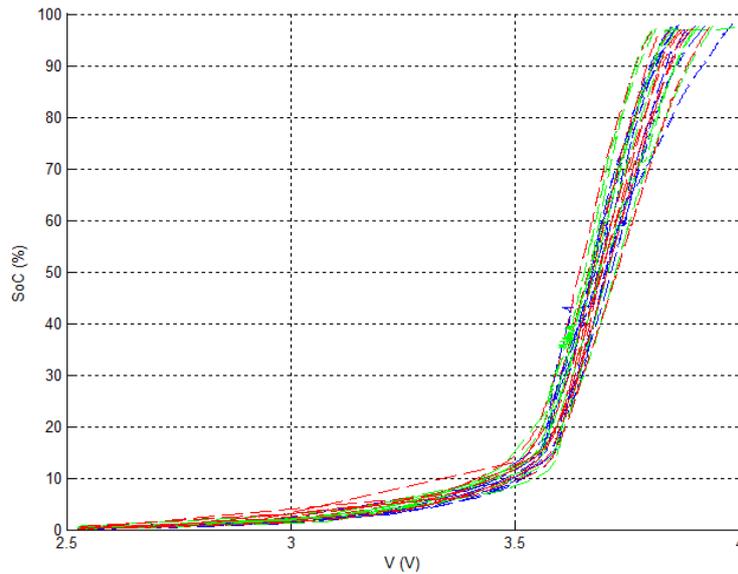


Figura 4-8: Curva de tensión vs SoC ideal.

## 4.5 Modelado

El primer paso en esta sección es el preprocesado de los datos para eliminar posibles datos ectópicos. Para ello se detectan valores fuera de rango y pendientes muy pronunciadas para restaurar los valores (si es una muestra puntual) o eliminarlos si no se puede restaurar.

Después, los datos provenientes de las medidas de tensión se han filtrado con un filtro IIR de orden 2 y frecuencia de corte 0,02 Hz para suavizar los cambios, ya que la naturaleza de esas señales corresponde a cambios muy suaves.

Sobre los datos filtrados, para cada muestra se han obtenido varios parámetros derivados de las medidas originales (V, I y temperatura) para comprobar si aportan nueva información. En un paso posterior se descartarán los parámetros que no aportan información relevante. Los parámetros finales son:

- V\_high, V\_low, I\_high, I\_low y temperatura. Estos son los datos originales capturados directamente por el nodo sensor.
- Exponenciación de V\_low. Potencias de la 2 a la 5 del nivel de tensión. Mediante estos parámetros se explorarán los ajustes polinómicos de la tensión de la batería.
- Pendiente del nivel de tensión, calculada cada 100 muestras.
- Potencias de la 2 a la 5 de la pendiente.
- Exponencial de la tensión ( $10^{V_{low}}$ ).
- Diferencia entre la tensión a alta corriente y a baja.
- Relación entre tensiones y corrientes a alta y baja corriente (R). Esta medida está inspirada en la medida de resistencia interna de la batería, pero se debe señalar que al no realizarla sobre circuito abierto, no se le puede denominar así.
- Ciclos de desgaste de la batería y logaritmo del mismo.
- Tiempo respecto al inicio de la sesión (este parámetro sólo se utiliza como referencia en gráficas, no se utiliza en el modelado).

De cada una de las condiciones de medida se han seleccionado cuatro de los seis ficheros disponibles para realizar este preprocesado, los cuales contenían las muestras originales con una frecuencia de muestreo de un segundo. Después se ha seleccionado 1000 muestras aleatorias de cada uno para combinar todas las medidas en un único fichero. De estas 1000 muestras se han reservado 100 para la última parte del fichero, de manera que esté sobrerrepresentado en la muestra final. El motivo es que la pendiente de caída final es muy corta, por lo que podría darse la situación de obtener buenos resultados globales pero muy malos en esta última parte. Finalmente se combinan todas las muestras en un único fichero de 24000 muestras. Se trabajará con este fichero en lo que resta del capítulo para obtener los diferentes modelos.

Para determinar qué parámetros aportan más información para desarrollar un modelo se ha utilizado un algoritmo basado en correlaciones presentado en [39]. Se ha utilizado Weka [40] evaluando dos familias de algoritmos: regresiones y redes neuronales.

### 4.5.1 Regresiones

En esta sección se parte del fichero anterior con 24000 muestras correspondientes a 24 de las sesiones realizadas, con los datos preprocesados como se acaba de explicar. Este fichero de datos se aplica a diferentes algoritmos de minería de datos implementados en Weka, dichos algoritmos devolverán los diferentes parámetros (tales como coeficientes, tipos de funciones, etc.) que permitirán implementar estos algoritmos.

Se ha evaluado una regresión lineal basada en un parámetro y en varios. La regresión lineal básica estaría basada en el nivel de tensión a nivel bajo de corriente ( $V_{low}$ ). En este caso la expresión obtenida a partir de las medidas anteriores sería la siguiente:

$$SoC(\%) = 88,6 \cdot V_{low} - 273,3 \quad (4.3)$$

Se ha seleccionado el parámetro  $V_{low}$  como variable  $V$  de la ecuación anterior porque es el equivalente a una lectura clásica de la batería, esta lectura se produce cuando el microcontrolador está activo pero con el *transceiver* apagado.

Los coeficientes de este modelo y de los siguientes son presentados automáticamente por Weka una vez configurados los datos y el algoritmo seleccionado. En este caso y el resto de modelos de esta sección, Weka utiliza una regresión lineal.

Una primera variante sobre esta ecuación sería realizarla por segmentos. En la parte plana de la curva de la tensión se utilizará una ecuación y en la curva otra función. Para elegir la tensión de umbral se puede utilizar un algoritmo de construcción de árboles de decisión como *Decision Stump*, el cual da un valor de 3,6 V. Dividiendo las muestras en dos grupos con el criterio anterior se obtienen las siguientes ecuaciones mediante regresión:

$$SoC(\%) = \begin{cases} V_{low} > 3,6 \rightarrow 253,7 \cdot V_{low} - 884,3 \\ V_{low} \leq 3,6 \rightarrow 23,5 \cdot V_{low} - 67,2 \end{cases} \quad (4.4)$$

Este modelo ajustará muy bien la primera parte de la descarga, ya que es donde la tensión sigue una recta. En la parte final de la descarga es donde presentará mayores errores por ser menos lineal.

Sin embargo el parámetro de la tensión de la batería no tiene porqué ser el único o el mejor. Evaluando los subconjuntos de atributos de forma individual mediante el método de [39] se obtiene que los mejores parámetros para predecir el resultado son la exponencial de la tensión y el número de ciclos de desgaste. Utilizando estos parámetros el modelo de regresión es:

$$SoC(\%) = -0,18 \cdot Temp + 0,016 \cdot 10^{V_{low}} + 1,13 \cdot \log(Cycles) - 26 \quad (4.5)$$

Finalmente hay que comentar que con todos los valores o mediante una transformación basada en análisis de componentes principales (PCA) se obtienen resultados similares o peores que los anteriores, pero con más procesamiento de datos.

#### 4.5.2 Redes neuronales

Se ha evaluado también el perceptrón multicapa (MLP) [41] ya que es un algoritmo que se ha utilizado previamente en aplicaciones de vehículos [42][43]. En este caso, tanto los datos originales como el resultado deben normalizarse y la salida, opcionalmente, desnormalizarse.

La red neuronal se puede entrenar en el PC y después implementarla en un software o en los nodos sensores. Las redes se han configurado de forma que tengan dos capas, 500 épocas de entrenamiento como máximo y sigmoide como función de activación. Los coeficientes de los modelos de esta sección provienen de este entrenamiento de tipo *backpropagation* [44].

Se comenzará como en el caso de las regresiones: evaluando únicamente el nivel de tensión. En este caso se utilizará una red con dos neuronas (utilizar un perceptrón simple sin salida de activación equivaldría a una regresión lineal), cuya expresión queda:

$$SoC(norm) = th_2 + k_2 \cdot \frac{1}{1 + e^{-(k_1 \cdot V_{low} + th_1)}} \quad (4.6)$$

Con los valores:

$$k_1 = -10,38; th_1 = 5,96; k_2 = -1,85; y th_2 = 0,95.$$

Utilizando otros parámetros para tener en cuenta el número de ciclos, la temperatura, la relación entre tensiones y corrientes a nivel alto y bajo (*R*) y la pendiente (*pend*) de la tensión se puede diseñar otra red neuronal:

$$\begin{aligned} n_1(norm) &= \frac{1}{1 + e^{-(k_{11} \cdot R + k_{12} \cdot V_{low} + k_{13} \cdot Temp + k_{14} \cdot slope + k_{15} \cdot \log(Cycles) + th_1)}} \\ n_2(norm) &= \frac{1}{1 + e^{-(k_{21} \cdot R + k_{22} \cdot V_{low} + k_{23} \cdot Temp + k_{24} \cdot slope + k_{25} \cdot \log(Cycles) + th_2)}} \\ n_3(norm) &= \frac{1}{1 + e^{-(k_{31} \cdot R + k_{32} \cdot V_{low} + k_{33} \cdot Temp + k_{34} \cdot slope + k_{35} \cdot \log(Cycles) + th_3)}} \end{aligned} \quad (4.7)$$

$$SoC(norm) = th_4 + k_{41} \cdot n_1 + k_{42} \cdot n_2 + k_{43} \cdot n_3$$

Donde:

$$k_{11} = -3,6; k_{12} = -8,98; k_{13} = -0,07; k_{14} = -0,17; k_{15} = -0,04; th_1 = 6,61$$

$$k_{21} = 0,73; k_{22} = 5,22; k_{23} = -0,41; k_{24} = -0,52; k_{25} = -5,62; th_2 = -6,7$$

$$k_{31} = -4,4; k_{32} = 8,55; k_{33} = 0,14; k_{34} = -0,55; k_{35} = -6,15; th_3 = -6,76$$

$$k_{41} = -2,03; k_{42} = -1,33; k_{43} = 1,04; th_4 = 1,07$$

## 4.6 Validación y resultados

El último paso ha sido evaluar los modelos propuestos en el punto anterior. La evaluación comenzará utilizando los datos con los que se ha generado el modelo, es decir, sobre el fichero de 24000 muestras utilizado en el modelado. Los resultados son:

Tabla 4-2: Resultados de los modelos propuestos sobre las medidas originales.

Modelo	Correlación	Error medio (SoC)	Error cuadrático (SoC)	Error relativo (%)	Error cuadr. relativo (%)
(4.2)(recta)	0,79	30,19	35,11	121,55	121,92
(4.3)(reg1)	0,79	14,67	17,62	59,08	61,18
(4.4)(reg segm)	0,97	5,72	7,11	23,04	24,69
(4.5)(reg3)	0,93	8,63	10,30	34,75	35,75
(4.6)(MLP1)	0,98	5,20	6,46	20,93	22,42
(4.7)(MLP5)	0,99	3,41	4,59	13,71	15,93

En la tabla anterior se muestran diversos parámetros para evaluar y cuantificar la bondad del ajuste. Los parámetros de error cuantifican el ajuste mediante diversos mecanismos dados por las ecuaciones (4.8) a (4.11). El error medio (*mae*) mide la desviación de cada muestra (*y*) respecto a la ideal (*x*) promediado por el número de muestras (*n*). El error cuadrático (*rmse*) es similar al anterior, pero realizando una media cuadrática. El error relativo (*rae*) mide el valor la desviación absoluta de una muestra respecto la ideal, normalizada. El error cuadrático relativo vuelve a ser la anterior utilizando una media cuadrática. Finalmente, la correlación está dada por la ecuación (4.12) y mide el parecido entre dos señales, en este caso el SoC ideal respecto al calculado en cada modelo.

$$mae = \frac{\sum |x - y|}{n} \quad (4.8)$$

$$rmse = \sqrt{\frac{\sum (x - y)^2}{n}} \quad (4.9)$$

$$rae = \frac{\sum |x - y|}{\sum |\bar{y} - y|} \cdot 100 \quad (4.10)$$

$$rse = \sqrt{\frac{\sum (x - y)^2}{\sum (\bar{y} - y)^2}} \cdot 100 \quad (4.11)$$

$$r = \frac{\sum(x - \bar{x}) \cdot (y - \bar{y})}{(n - 1) \cdot s_x \cdot s_y} \quad (4.12)$$

Para evaluar si el estado de salud de la batería (SoH) podría tener algún efecto en el cálculo de nuestro estado de carga, se han agrupado las trazas de más de 200 ciclos por un lado y las de menos de 10 por otro. Se ha hecho una comparativa de los valores de correlación y del *rae*. En la Figura 4-9 se puede el resultado obtenido. Se puede apreciar que las menores diferencias tomando en cuenta ambos parámetros se dan en las ecuaciones 5 (0,35 y 1,18%) y 7 (0,38 y 9,81%), que son las que tienen en cuenta varios parámetros, entre otros el número de ciclos de carga/descarga; mientras que las otras bien en alguno o en ambos parámetros sí tienen variaciones más elevadas (eq2: 3,66 y 6,21%; eq3: 3,66 y 0,75%; eq4: 0,22 y 15% y eq6: 0,46 y 21,5%).

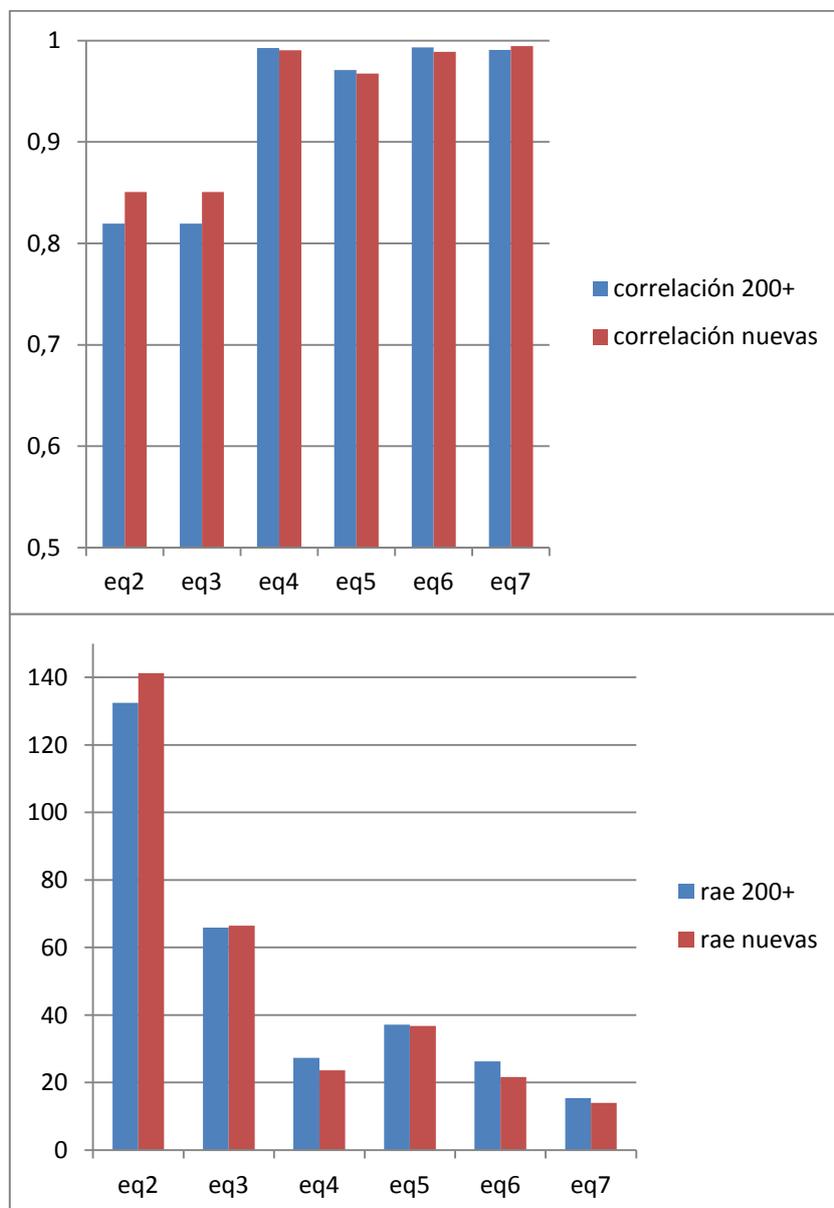
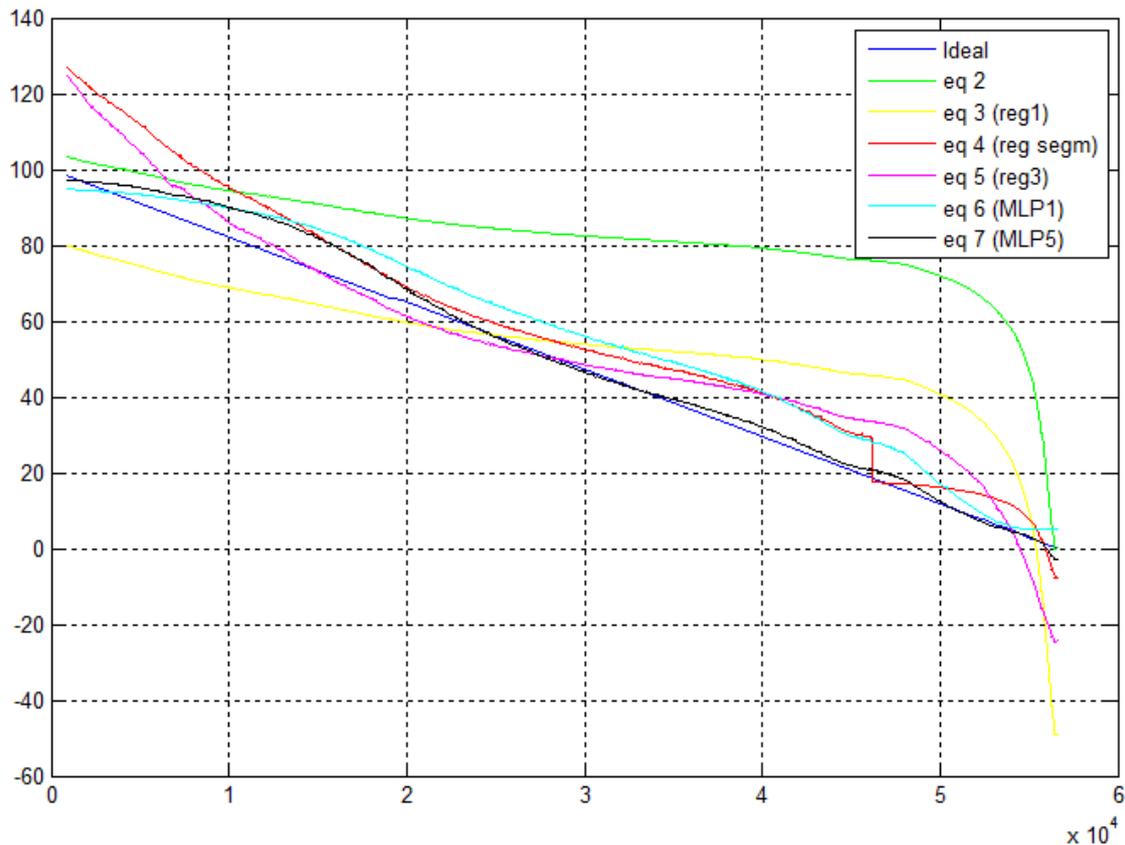


Figura 4-9: Efecto del SoH de la batería sobre los resultados de los modelos.

En el anexo 10.3 se muestran los resultados individuales para cada una de las 24 trazas utilizadas en la construcción del modelo.

En la Figura 4-10 se muestra la representación gráfica del ajuste del SoC para una sesión de descarga elegida de forma aleatoria, donde la recta azul es la ideal, calculada a partir de la duración total de la sesión y de los instantes de cada muestra; el resto de rectas corresponden a cada uno de los modelos propuestos.

El resto de trazas presentan comportamientos similares.



**Figura 4-10: Comparativa del ajuste entre la estimación de los modelos respecto al valor real durante una sesión completa de funcionamiento. Batería con 1 ciclo de desgaste y temperatura a 25 °C.**

La recta calculada a partir de los puntos extremos presenta el mayor error y únicamente da valores precisos en los puntos de ajuste. La recta a partir de la regresión de la tensión es igual a la anterior, por lo que da valores de correlación idénticos, pero presenta menos error porque éste lo minimiza distribuyéndolo a lo largo de todo el rango. La regresión segmentada basada en la tensión presenta una muy buena correlación porque durante la mayor parte del rango de la tensión, ésta es lineal, no obstante el error en los últimos tramos es grande y presenta otro gran problema: puede haber una discontinuidad en el momento de pasar de un segmento a otro. La regresión con tres parámetros hace que se tenga en cuenta una compensación por temperatura y por ciclos de carga/descarga; además el cálculo de la potenciación linealiza más la tensión. Las dos redes neuronales, tal como se puede ver, ajustan el resultado mucho más al valor deseado en todo el rango al dar como resultado funciones no lineales que se ajustan a las medidas iniciales.

Los resultados anteriores son valores utilizando los mismos datos con los que se generó el modelo. Para estudiar la bondad del modelo y su generalidad se hará lo mismo pero respecto a un conjunto de muestras distintas a las utilizadas para generar el modelo. Para comprobar si el modelo puede ser extrapolable a otras aplicaciones se ha utilizado:

- Una batería diferente a las utilizadas en la generación de modelos, pero del mismo tipo.
- Temperaturas y ciclos de desgaste diferentes (10 y 35 °C; 20 y 720 ciclos).
- Una batería de tipo y capacidad diferente (110 mAh).

La estimación del SoC por los modelos propuestos utiliza parámetros puntuales, es decir, no toma en consideración el histórico de la sesión, como se haría si se incluye la integral de la corriente consumida (excepto la pendiente, que utiliza la tensión de 100 muestras antes), por lo que los modelos generados pueden utilizarse sin que los ciclos de carga y descarga sean completos (asumiendo que el desgaste de un ciclo incompleto es equivalente a uno completo).

Tabla 4-3: Resultados de los modelos propuestos sobre las medidas no consideradas en la generación.

Sesión	Algoritmo	Correlación	Error medio (SoC)	Error cuadrático (SoC)	Error relativo (%)	Error cuadr. relativo (%)
40 mAh, 10 °C, 3 ciclos	recta	0,83	32,66	38,61	135,01	138,20
	Reg1	0,83	17,13	19,58	70,81	70,10
	segm	0,98	5,82	7,70	24,08	27,56
	Reg3	0,95	9,89	11,89	40,87	42,57
	MLP 1	0,99	7,41	9,04	30,62	32,37
	MLP 5	0,99	4,72	5,30	19,61	19,04
	MAX	0,99	7,96	9,87	31,83	34,20
40 mAh, 35 °C, 21 ciclos	recta	0,85	33,45	38,64	139,93	139,96
	Reg1	0,85	16,01	18,36	66,97	66,52
	segm	0,99	7,71	8,71	32,27	31,55
	Reg3	0,97	7,61	8,84	31,83	32,02
	MLP 1	0,97	9,33	10,60	39,05	38,39
	MLP 5	0,99	8,55	9,44	36,17	34,57
	MAX	0,95	6,51	9,38	26,05	32,48
40 mAh, 10 °C, 726 ciclos	recta	0,87	31,87	36,40	134,11	132,61
	Reg1	0,87	14,09	16,06	59,28	58,53
	segm	0,99	6,23	6,76	26,23	24,64
	Reg3	0,97	8,11	9,73	34,13	35,45
	MLP 1	0,98	6,63	7,76	27,89	28,25
	MLP 5	0,99	6,41	7,45	26,96	27,16
	MAX	0,99	5,18	6,80	20,71	23,56
110 mAh, 10 °C, 1 ciclo	recta	0,92	35,19	40,28	141,90	140,64
	Reg1	0,93	16,00	18,94	64,53	66,13
	segm	0,98	13,55	14,52	54,64	50,71
	Reg3	0,96	12,76	15,01	52,49	53,43
	MLP 1	0,98	12,81	13,82	51,64	48,27
	MLP 5	0,99	3,10	4,33	12,76	15,41
	MAX	0,99	4,55	5,02	18,20	17,39

Además, para comparar la validez de los modelos respecto a soluciones que hay en el mercado, se utilizará un *fuel gauge* MAX17043 [45]. Este circuito integrado implementa un medidor de baterías de litio que no necesita un proceso de calibración, elimina la necesidad de la medición de corriente y tiene compensación de temperatura.

Como se puede ver, todos los modelos mejoran respecto a la referencia (ecuación 2). Los resultados tienden a ser mejores cuanto más complejo es el modelo. Los modelos con mejores resultados son los basados en redes neuronales, siendo comparables al MAX17043 en precisión, pero con la ventaja de no requerir ningún chip adicional.

Los únicos modelos equivalentes que se han encontrado en la bibliografía son [14] y [18]. Otros modelos no son adecuados para redes de sensores, fundamentalmente por su coste computacional. El método [14] es un estimador basado en un contador de culombios que no utiliza medidas de la tensión o corriente durante el funcionamiento del nodo; su error sobre una aplicación en particular está alrededor del 30%. El método propuesto en [18] es más comparable. Este método está descrito en la sección 2 y su resultado es una ecuación cuadrática utilizando la medida de la tensión de la batería; los coeficientes de la ecuación son obtenidos de ajustes previos. No tiene en cuenta directamente factores como la temperatura o los ciclos de desgaste. En este método podemos asumir que para una aplicación particular y periódica tenemos una carga constante, en este caso el método lo podemos simplificar realizando una regresión lineal sobre los parámetros  $V_{low}$  y  $V_{low}^2$  de nuestras medidas originales. En este caso el mejor ajuste obtenido es:

$$DoD = -142,9557 \cdot V_{low}^2 + 861,4243 \cdot V_{low} - 1184,2632 \quad (4.13)$$

$$SoC = 100 - DoD \quad (4.14)$$

Este modelo consigue una correlación de 0,93 sobre los datos originales y un error cuadrático relativo del 41,8 %. Como se puede esperar, los resultados son mejores que una regresión simple, pero peores que otros métodos con más capacidad para linealizar, como las redes neuronales.

Se han implementado los cinco modelos anteriores sobre TinyOS en Tmote Sky para evaluar su coste computacional. En los casos que son necesarios, también se ha incluido el cálculo de las corrientes, pendiente y resto de parámetros; en el caso de las redes neuronales se ha incluido la normalización de los parámetros de entrada. En la medida no se han incluido la captura de los datos, los filtros y otras operaciones de preprocesado, o la des-normalización de la salida de las redes neuronales porque son operaciones optativas. La medida se ha realizado utilizando como base el componente de TinyOS `Counter32khz32C` que implementa un reloj de 32 kHz utilizando un temporizador del microcontrolador. El coste de los modelos que utilizan regresiones lineales son de 17, 20 y 734 ciclos de reloj del contador respectivamente; el coste de las redes neuronales ha sido de 249 y 664 ciclos respectivamente. El alto coste de la regresión que utiliza 3 parámetros se debe fundamentalmente a la función de potenciación.

Los tiempos anteriores se pueden traducir a consumo energético si se realiza una serie de suposiciones. Se partirá del consumo medio de la aplicación actual, que es de 954  $\mu$ A. En este caso se asumirá que el microcontrolador consumirá 2,5 mA (MSP430 @ 4 MHz) durante la computación del modelo, y también se asumirá que estos modelos son ejecutados cada segundo. Así el consumo se incrementará en +1%, +1%, +4,8%, +2% y +4% respectivamente. En la Tabla 4-4 se muestran los valores numéricos.

**Tabla 4-4: Coste computacional y energético de los modelos propuestos.**

<b>Modelo</b>	<b>Tiempo (ms)</b>	<b>I media (mA)</b>
Eq 2 & Eq 3 Reg1	0,531	0,955
Eq 4 segm	0,625	0,955
Eq 5 Reg3	23	1
Eq6 MLP 1	7,8	0,973
Eq7 MLP 5	20	0,994

Finalmente, también se ha evaluado el coste computacional de (4.13). Su implementación usando las condiciones anteriores necesita un tiempo de ejecución de 19 ciclos (0,593 ms y 0,955 mA), el cual es similar a los modelos basados en regresiones lineales.

Obviamente, en una implementación práctica el periodo para el cálculo del SoC podrá ser mayor que el considerado anteriormente (1 s), con lo que el consumo será todavía menor.

## 4.7 Conclusiones

Se ha comprobado que el método propuesto es válido para nodos de redes de sensores. El método se compone de varias etapas y usa software y hardware abierto. El primer paso incluye la adquisición de la temperatura, corriente y tensión de la batería durante la operación de un nodo durante un funcionamiento típico. El número de ciclos de carga/descarga se ha variado por medio de un banco de desgaste que aplica continuamente ciclos acelerados. Una vez que los datos se han adquirido, la etapa de procesamiento elimina ruido y errores para que los datos tengan la máxima calidad antes de generar los modelos. Los modelos propuestos se basan en regresiones lineales y redes neuronales. Posteriormente estos modelos se han validado con un nuevo conjunto de datos.

Comparando los resultados entre baterías con muchos ciclos de desgaste y muy pocos puede verse que no hay variación significativa en la precisión de los modelos, a lo que habría que sumar que la traza con 720 ciclos sigue dando resultados similares; de lo que se deduce que éstos son independientes —más bien tienen en cuenta— el estado de salud de la batería a lo largo de su vida.

En los mejores casos —los modelos basados en MLP—, los resultados han obtenido valores de 0,99 en la correlación y un error absoluto en torno a 5. Los modelos basados en regresiones se han utilizado como referencia para comparar la mejora obtenida.

Todos estos modelos tienen distintos costes computacionales, pudiéndose aplicar unos u otros en función del balance de precisión/coste deseado. En general estos modelos tienen un bajo

coste computacional para caracterizar el consumo de un nodo sensor, lo que los habilita para poder ser ejecutados en un microcontrolador en la mayoría de aplicaciones. Aunque también podrían ser aplicados en *routers* o nodos centrales.

Los modelos han sido capaces de funcionar para baterías diferentes a la probada originalmente y en situaciones distintas a las que se daban al tomar las medidas originales. El ciclo de desgaste se ha evaluado en valores muy alejados de los del modelo y han seguido funcionando bien. Para el caso de la temperatura, los valores muy alejados del rango inicial dan más error, lo que sugiere que en esos valores su efecto no puede modelarse como lineal, el mismo efecto se ha observado en el *fuel gauge*.

Además, este método podría ser utilizado también en programas de simulación para hacer simulaciones más realistas y predecir con más precisión la esperanza de vida de los nodos y de la red.

En el futuro se puede ampliar este trabajo añadiendo más parámetros a la generación de los modelos, como la edad de la batería, aplicación de varias funciones de ajuste sobre la tensión, temperatura, etc. También se pueden explorar nuevos algoritmos para buscar métodos más precisos o con menor coste computacional.

## 4.8 Bibliografía

- [1] Alberola, J., Pelegri, J., Lajara, R., & Perez, J. J. (2008, May). Solar inexhaustible power source for wireless sensor node. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*(pp. 657-662). IEEE.
- [2] Wang, Q., & Yang, W. (2007, June). Energy consumption model for power management in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on* (pp. 142-151). IEEE.
- [3] Wang, Q., Hempstead, M., & Yang, W. (2006, September). A realistic power consumption model for wireless sensor network devices. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on* (Vol. 1, pp. 286-295). IEEE.
- [4] Baoqiang, K., Li, C., Hongsong, Z., & Yongjun, X. (2008). Accurate energy model for WSN node and its optimal design. *Journal of Systems Engineering and Electronics*, 19(3), 427-433.
- [5] Kan, B., Cai, L., Zhao, L., & Xu, Y. (2007, September). Energy efficient design of WSN based on an accurate power consumption model. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on* (pp. 2751-2754). IEEE.
- [6] Zhao, L., Zhang, W., Xu, C., Xu, Y., & Li, X. (2006). Energy-aware system design for wireless sensor network. *ACTA Automatica Sinica*, 32(6), 892.
- [7] Norouzi, A., & Sertbas, A. (2011). An Integrated Survey in Efficient Energy Management for WSN Using Architecture Approach. *International Journal of Advanced Networking and Applications*, 3(1), 968-977.

- [8] Bougard, B., Catthoor, F., Daly, D. C., Chandrakasan, A., & Dehaene, W. (2008, January). Energy efficiency of the IEEE 802.15. 4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In *Design, Automation, and Test in Europe* (pp. 221-234). Springer Netherlands.
- [9] Tan, H. P., Lee, P. W., Seah, W. K. G., & Eu, Z. A. (2009, May). Impact of power control in wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) for railroad health monitoring. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on* (pp. 804-809). IEEE.
- [10] Reddy, B. V., & Reddy, D. R. K. Maximizing the usable battery capacity in WSN with load harmonization.
- [11] Zhou, H., Luo, D., Gao, Y., & Zuo, D. C. (2011). Modeling of Node Energy Consumption for Wireless Sensor Networks. *Wireless Sensor Network*, 3(1), 18-23.
- [12] Wang, Q., & Yang, W. (2007, June). Energy consumption model for power management in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on* (pp. 142-151). IEEE.
- [13] Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. (2006). An energy-efficient ant-based routing algorithm for wireless sensor networks. In *Ant Colony Optimization and Swarm Intelligence* (pp. 49-59). Springer Berlin Heidelberg.
- [14] Castagnetti, A.; Pegatoquet, A.; Belleudy, C.; Auguin, M., An efficient state of charge prediction model for solar harvesting WSN platforms, *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, vol., no., pp.122,125, 11-13 April 2012.
- [15] Kim, N., Choi, S., & Cha, H. (2008, September). Automated sensor-specific power management for wireless sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* (pp. 305-314). IEEE.
- [16] Winter, M., & Brodd, R. J. (2004). What are batteries, fuel cells, and supercapacitors?. *Chemical reviews*, 104(10), 4245-4270.
- [17] Evanczuk, S. (2013). Advanced ICs Simplify Accurate State-of-Charge Measurement for Lithium-Ion Batteries. (online document, available in <http://www.digikey.com/us/en/techzone/energy-harvesting/resources/articles/advanced-ics-simplify-state-of-charge-measurement.html>)
- [18] Buchli, B., Aschwanden, D., & Beutel, J. Battery state-of-charge approximation for energy harvesting embedded systems. In *Wireless Sensor Networks, Springer Lecture Notes in Computer Science Volume 7772, 2013, pp 179-196*
- [19] Charkhgard, M.; Farrokhi, M., State-of-Charge Estimation for Lithium-Ion Batteries Using Neural Networks and EKF, *Industrial Electronics, IEEE Transactions on*, vol.57, no.12, pp.4178,4187, Dec. 2010
- [20] Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, 241, 680-689.
- [21] Doerffel, D., & Sharkh, S. A. (2006). A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. *Journal of Power Sources*, 155(2), 395-400.

- [22]Barreras, J. V., Schaltz, E., Andreasen, S. J., & Minko, T. (2012, October). Datasheet-based modeling of Li-Ion batteries. In *Vehicle Power and Propulsion Conference (VPPC), 2012 IEEE* (pp. 830-835). IEEE.
- [23]Buchli, B., Aschwanden, D., & Beutel, J. (2013). Battery state-of-charge approximation for energy harvesting embedded systems. In *Wireless Sensor Networks* (pp. 179-196). Springer Berlin Heidelberg.
- [24]Rao, R., Vrudhula, S., & Rakhmatov, D. N. (2003). Battery modeling for energy aware system design. *Computer*, 36(12), 77-87.
- [25]Battery Internal Resistance (2005). Technical bulletin Energizer. (online document, available in <http://data.energizer.com/PDFs/BatteryIR.pdf>)
- [26]Remmlinger, J., Buchholz, M., Meiler, M., Bernreuter, P., & Dietmayer, K. (2011). State-of-health monitoring of lithium-ion batteries in electric vehicles by on-board internal resistance estimation. *Journal of Power Sources*, 196(12), 5357-5363.
- [27]Williard, N. D. (2011). *Degradation Analysis and Health Monitoring of Lithium Ion Batteries* (Doctoral dissertation).
- [28]Zhao, S., Wu, F., Yang, L., Gao, L., & Burke, A. F. (2010). A measurement method for determination of dc internal resistance of batteries and supercapacitors. *Electrochemistry Communications*, 12(2), 242-245.
- [29]Buchmann, I. Does internal resistance reveal battery capacity?. (online document, available in <http://www.buchmann.ca/article25-page1.asp>)
- [30]Xtra-power. Rechargeable 3.7V Li-ion Polymer Battery Specification, 40 mAh. Available online: [http://www.xtra-power.com.hk/Products/Li-polymer/Standard%20Type/Xtra-power%20051213\(40mAh\).pdf](http://www.xtra-power.com.hk/Products/Li-polymer/Standard%20Type/Xtra-power%20051213(40mAh).pdf) (accessed on 1 12 2013)
- [31]Moteiv Corporation. Datasheet tmote sky (2006). (online document, available in <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>)
- [32]Texas Instruments. Datasheet CC2420. (online document, available in <http://www.ti.com/lit/ds/symlink/cc2420.pdf>)
- [33]Sensirion. Datasheet SHT11. (online document, available in [http://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/Humidity/Sensirion\\_Humidity\\_SHT25\\_Datasheet\\_V2.pdf](http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT25_Datasheet_V2.pdf))
- [34]Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., ... & Culler, D. (2005). TinyOS: An operating system for sensor networks. In *Ambient intelligence* (pp. 115-148). Springer Berlin Heidelberg.
- [35]Lajara, R.; Pelegri-Sebastia, J.; Perez-Solano, J.J. Power consumption analysis of operating systems for wireless sensor networks. *Sensors* 2010, 10, 5809-5826.
- [36]Microchip. Datasheet MCP73831/2. (online document, available in <http://ww1.microchip.com/downloads/en/DeviceDoc/21984F.pdf>)
- [37]Choi, S. S., & Lim, H. S. (2002). Factors that affect cycle-life and possible degradation mechanisms of a Li-ion cell based on LiCoO<sub>2</sub>. *Journal of Power Sources*, 111(1), 130-136.
- [38]Kim, N., Choi, S., & Cha, H. (2008, September). Automated sensor-specific power management for wireless sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* (pp. 305-314). IEEE.

- [39]M. A. Hall (1998). Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand.
- [40]Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter, 11(1), 10-18.
- [41]Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.
- [42]Gong, Q., Li, Y., & Peng, Z. (2009, June). Power management of plug-in hybrid electric vehicles using neural network based trip modeling. In American Control Conference, 2009. ACC'09. (pp. 4601-4606). IEEE.
- [43]Sheikhan, M., Pardis, R., & Gharavian, D. (2013). State of charge neural computational models for high energy density batteries in electric vehicles. Neural Computing and Applications, 1-10.
- [44]Rummelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, 323(9), 533-536.
- [45]Maxim Integrated. Datasheet fuel gauge MAX17044. (online document, available in <http://www.maximintegrated.com/datasheet/index.mvp/id/6546>)



# 5. FUENTE DE ENERGÍA SOLAR PERPETUA PARA NODOS SENSORES

---

## 5.1 Introducción

Este capítulo se enfoca a la construcción de una fuente de energía basada en paneles solares para nodos sensores. Observando la tendencia de la tecnología se puede apreciar que el consumo de los circuitos integrados, especialmente los dedicados a comunicaciones, tiende a reducirse. De esta forma es viable utilizar energías renovables para hacer que algunos dispositivos sean energéticamente autónomos. Hay varios artículos que discuten sobre la cantidad de energía obtenida de varias fuentes de energía ambientales [1-8]. De un análisis de todas ellas se observa que en la actualidad la energía solar es la que más energía puede aportar y a la que se puede tener acceso en muchas aplicaciones de las redes de sensores. Un inconveniente de las energías renovables es que no siempre se puede garantizar su aportación, por lo que es necesario introducir algún mecanismo de almacenaje, típicamente baterías. Actualmente (año 2007 —fecha de la que data este trabajo—), los paneles solares son capaces de cargar una batería y alimentar un nodo típico de bajo consumo de las redes de sensores [9-11]. Por lo tanto, el principal reto de diseñar una fuente de alimentación perpetua para nodos sensores no es simplemente extraer la mayor cantidad posible de energía de la fuente, sino principalmente almacenarla. Así, es esencial una cuidadosa selección de componentes en el circuito de alimentación para lograr alta eficiencia y bajas pérdidas. Para lograr que la fuente de energía sea realmente longeva se debe tener en cuenta que las baterías sufren envejecimiento, el cual reduce drásticamente su capacidad en dos o tres años si hay que recargar diariamente.

## 5.2 Estado del arte

La arquitectura típica de un nodo sensor alimentado con energía solar es: panel solar, circuito de control de carga y una unidad de almacenaje [12][13]. Desafortunadamente el problema del envejecimiento de las baterías siempre está presente. Por tanto, el diseño de una fuente de alimentación perpetua debe prestar atención a los elementos de almacenaje, y utilizar algunos que tengan poca fatiga. Un elemento de almacenaje con poca fatiga son los supercondensadores. Como se describe en [14], «Prometheus» usa una arquitectura basada en dos tipos de elementos de almacenaje de energía o buffers: dos supercondensadores y una batería Li+. En ese sistema se utiliza un microcontrolador para controlar los ciclos de recarga de la batería y seleccionar los caminos que recorrerá la energía hasta la carga a través de varios *switches*. Esta arquitectura es muy dependiente del microcontrolador, que supervisa la

conmutación y los umbrales de tensión. Esta supervisión necesita ser continua, o al menos periódica con una frecuencia relativamente alta. Sin embargo las redes de sensores normalmente trabajan con esquemas despierto/dormido con ciclos de trabajo muy bajos. Un sistema de alimentación como el descrito antes, aplicado a un nodo sensor puede provocar la caída del sistema si el tiempo entre monitorizaciones del sistema de energía se retrasa y no se conmutan los *switches* a tiempo. Por otra parte, el consumo de un microcontrolador exclusivo para supervisar la energía puede ser comparable al del resto del nodo debido a su frecuencia de trabajo y es comparable o superior al de circuitos integrados supervisores de tensión funcionando todo el tiempo.

Por lo comentado antes, se evaluará una nueva arquitectura, mostrada en la Figura 5-1. Esta arquitectura es independiente de un microcontrolador. Una unidad de control discreta y automática controlará el flujo de energía hasta la carga. Además, la batería es recargada del panel solar independientemente del estado de los supercondensadores. Son los supercondensadores los que proporcionan principalmente la energía y los que sufren los ciclos de carga y descarga, la batería sólo se utiliza en caso que los supercondensadores se agoten.

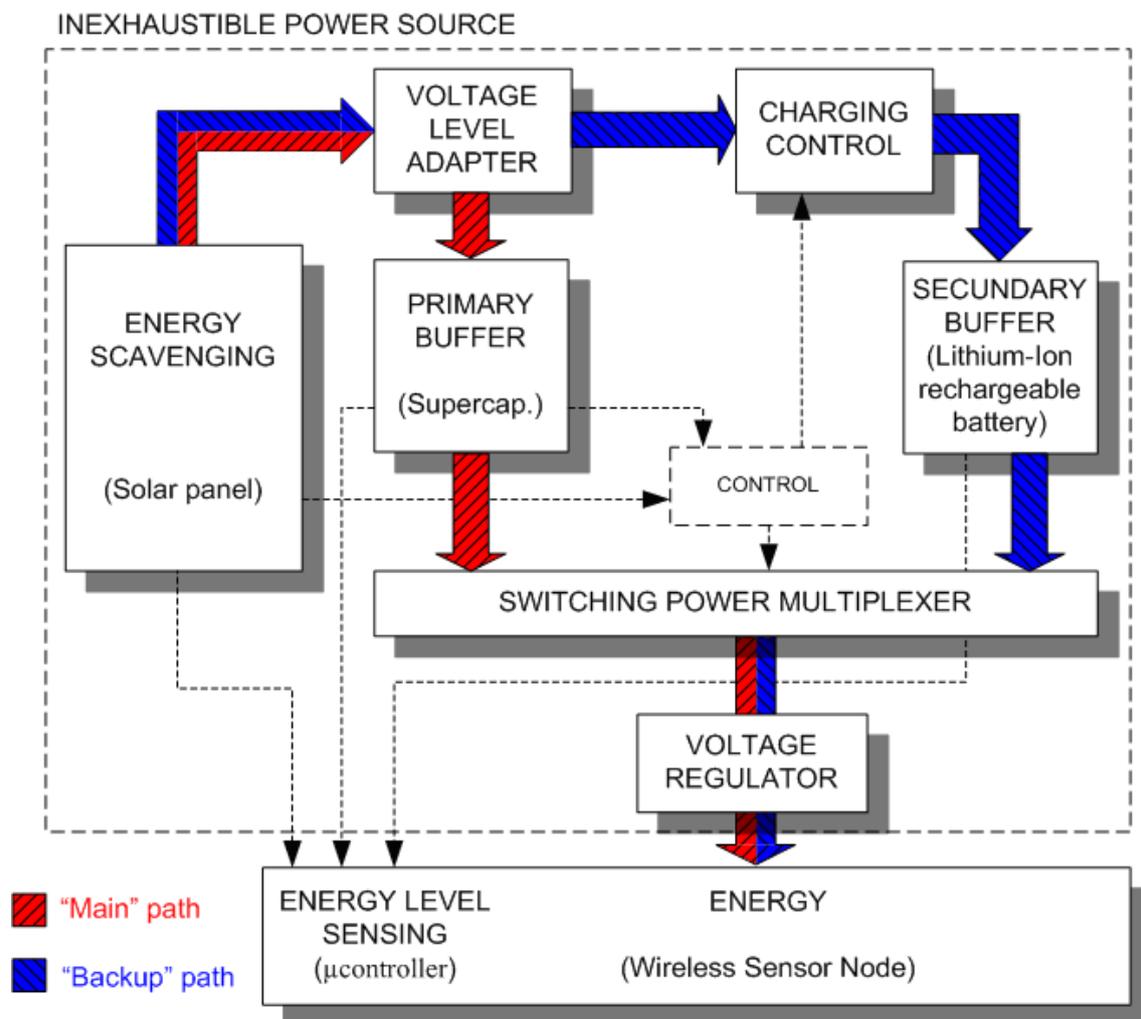


Figura 5-1: Diagrama de bloques de la arquitectura del sistema.

En este capítulo se propondrá y evaluará una nueva arquitectura de un sistema de alimentación, la cual se muestra en la Figura 5-1. Esta arquitectura es independiente de un

microcontrolador que la supervise. En su lugar hay una unidad de control automático que gestiona el flujo de energía a la carga. Además, la batería puede recargarse a partir de la energía del panel solar, independientemente del estado de los supercondensadores, lo que permite que los supercondensadores mantengan su carga y así se minimice el uso y desgaste de la batería. También se realizará un análisis de los requisitos de corriente en los dos caminos del flujo de corriente (*Main path* y *Backup path* en la figura) tanto en la carga como en la descarga, lo que permitirá maximizar la eficiencia de cada camino eligiendo los componentes más adecuados para cada corriente requerida.

### 5.3 Diseño

La fuente de alimentación propuesta debe generar al final un voltaje de 3,3 V para alimentar un nodo sensor, el cual está formado por un microcontrolador, un *transceiver* inalámbrico CC2420 y sensores. Este tipo de dispositivos inalámbricos suelen trabajar en esquemas dormido/desperto con muy bajos ciclos de trabajo, es decir, permanecen la mayor parte del tiempo en estado de bajo consumo, y de forma puntual activan los diversos periféricos para realizar una tarea. El motivo es para minimizar el consumo de energía y aumentar la autonomía. Los ciclos de trabajo típicamente van entre 1% y 10% en aplicaciones de sensado [15][16]. Por lo tanto, el diseño se enfocará a maximizar la eficiencia para manejar corrientes muy pequeñas. Dado que la mayor parte del tiempo, como se acaba de mencionar, el sistema estará en estado dormido, se prestará especial atención a este modo. Los modernos microcontroladores pueden tener un consumo en el modo más profundo de *sleep* por debajo del microamperio; si se suma esta corriente a la del *transceiver* y el resto de dispositivos, se puede tener una corriente en *sleep* de unos 6  $\mu$ A.

En el caso del modo activo, el consumo principal correspondería al *transceiver* (17 mA), más el microcontrolador en activo y el resto de dispositivos puede ser una corriente total de unos 22 mA. El tiempo en modo activo es dependiente de la aplicación, pero en este análisis se ha elegido arbitrariamente una duración de 100 ms porque es un orden de magnitud bastante habitual.

El resto del tiempo se considerará que el sistema está en modo *sleep*.

En la Figura 5-2 las líneas rectas corresponden con la corriente consumida por el dispositivo, donde se distinguen los estados *sleep* (baja corriente) y activo (alta corriente). La línea punteada representa el consumo medio teniendo en cuenta los consumos en cada estado y la duración de los mismos. Los valores en la gráfica son aproximados, y su propósito es modelar un sistema típico de un nodo inalámbrico.

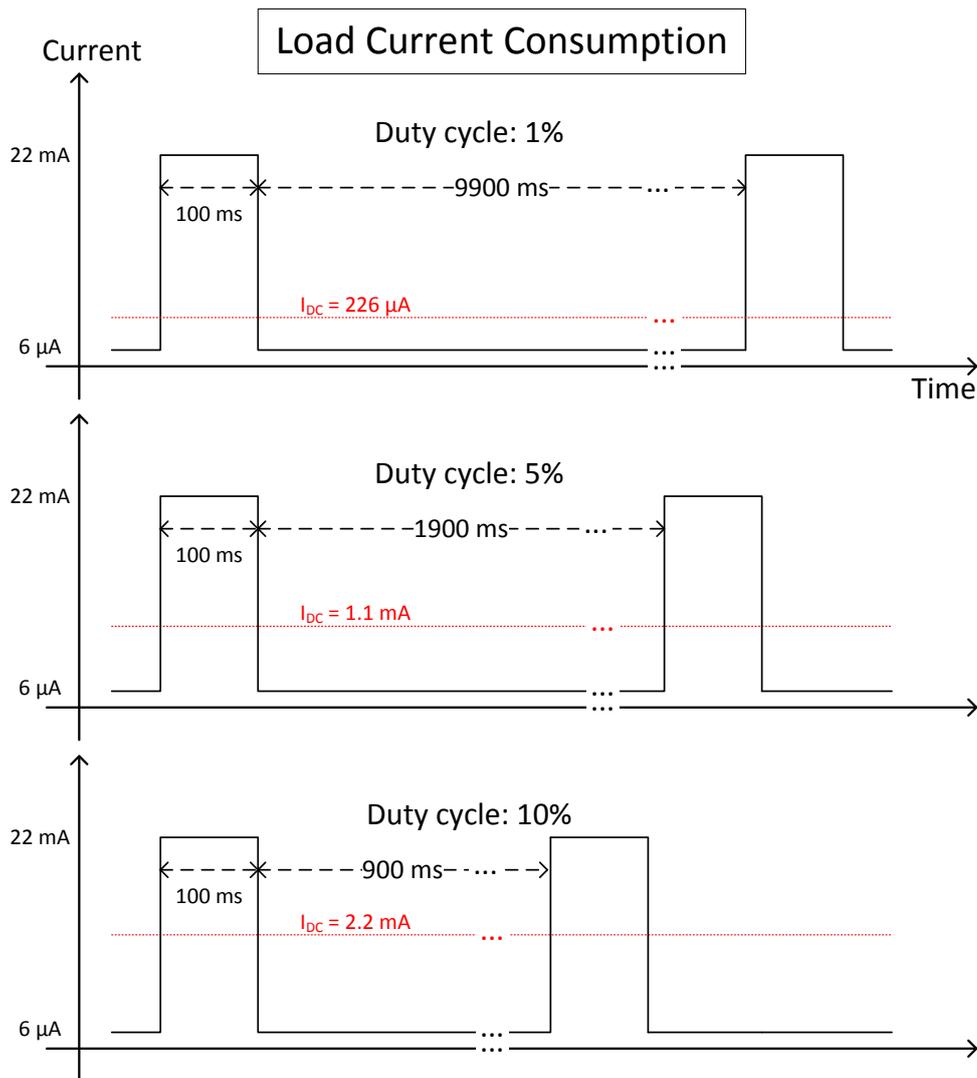


Figura 5-2: Modelos de consumo de un nodo sensor típico a diferentes ciclos de trabajo.

Como se ha mencionado anteriormente, se analizarán los dos caminos de la Figura 5-1 por separado para adecuar los componentes a los requisitos de cada uno de ellos.

### 5.3.1 Main path

Este camino es el principal, por él fluirá la corriente de forma predeterminada cuando el panel solar está generando energía. Cuando esto ocurre, el bloque de control da a este camino la mayor prioridad. Durante el funcionamiento normal, este camino siempre estará activo para enviar la energía desde el panel solar al buffer principal (supercondensadores) y a la carga (el nodo inalámbrico). De esta forma, la utilización del buffer secundario (la batería) se reduce al mínimo y la vida útil de todo el sistema es ampliada.

El primer elemento de este camino es el panel solar. Desde la fuente se debe suministrar una corriente de al menos 22 mA a la carga, aunque será deseable que sea un valor lo más alto posible para poder cargar los elementos de almacenaje a la vez que se alimenta el nodo inalámbrico en todo momento. Además, cuanto mayor sea la corriente, mayor será la velocidad de recarga de los supercondensadores y la batería. Por lo tanto, el único límite superior de la corriente lo determina el tipo y tamaño del panel solar para una aplicación

determinada. Los paneles solares se comportan como fuentes de corriente limitadas en tensión [17][18][13], un comportamiento contrario a las baterías, las cuales son fuentes de tensión. Los paneles solares tienen un punto de operación conocido como «punto de máxima potencia» (MPP, *Maximum Power Point*) en el cual será deseable trabajar. El punto de funcionamiento se mueve a lo largo de las curvas V-I del panel solar en función de la radiación solar incidente, de manera que para obtener la máxima potencia hay que hacer un seguimiento del punto MPP en las curvas V-I. Una primera aproximación sería utilizar un convertidor DC-DC avanzado que ajusta dinámicamente su ciclo de trabajo [19][20]. Desafortunadamente, en el momento de realizar este sistema (año 2007) no se encuentran disponibles soluciones comerciales de seguimiento de MPP para bajas potencias. Otra aproximación sería dedicar un microcontrolador para implementar esta característica, pero éste implicaría un mayor gasto energético, además de entrar en contradicción con la filosofía del diseño de prescindir de estos elementos. Por lo tanto sólo quedan dos opciones viables para acoplar de forma eficiente el panel solar a los supercondensadores: directamente o utilizando un convertidor DC-DC típico. La opción elegida ha sido el convertidor DC-DC porque eleva la tensión del panel solar a un nivel adecuado incluso cuando hay poca radiación solar, como en días nublados.

El siguiente elemento en el *Main path* es el buffer principal. Este es el elemento crítico en el diseño. La clave del diseño para hacer que el sistema sea autónomo es que el buffer primario se recargue cada día sin acusar un desgaste notable. Al mismo tiempo, se necesita una capacidad de almacenaje suficientemente alta para mantener el sistema en marcha durante las noches, lo que minimizará el uso del buffer secundario. Tal como se describe en el sistema Prometheus [14], la única solución actual disponible comercialmente son los supercondensadores. Existen otros elementos que tienen mayor capacidad de almacenaje, como las baterías, sin embargo el problema de envejecimiento que sufren los hace inviables. Este problema consiste en la pérdida de capacidad que se manifiesta en un aumento de resistencia interna causada por la oxidación producida en los sucesivos ciclos de recarga [21][22]. Por ejemplo, una batería de litio sufre una degradación del 10 al 20% en la capacidad y resistencia después de únicamente 300-500 ciclos de carga/descarga, lo que supone entre uno y dos años de recargas diarias. Por el contrario, los supercondensadores han alcanzado grandes capacitancias, suficientes para aplicaciones como las tratadas en este capítulo, a precios competitivos y sin estos problemas que se acaban de mencionar. En comparación con una batería de litio, un supercondensador tiene una vida útil de 500.000-1.000.000 ciclos de carga/descarga para tener un desgaste comparable al ejemplo anterior (10-20%), lo que supondría más de 1.000 años de vida útil con un ciclo diario de carga/descarga [23][26]. Por otra parte, los supercondensadores tienen ventajas adicionales para aplicaciones pulsantes, como las de las redes de sensores, como la alta densidad de potencia, alta eficiencia, tiempo de recarga corto y gran ciclo de vida. En contraste, la principal desventaja de los supercondensadores es una densidad de energía comparativamente inferior a las baterías [29]. Sin embargo, a pesar de esta desventaja siguen siendo una buena solución para aplicaciones de baja potencia y bajos ciclos de trabajo donde la energía solar puede recargarlos diariamente.

Por otra parte, cuando no hay energía solar disponible, los supercondensadores comenzarán a suministrar corriente al nodo y, poco a poco, su tensión irá bajando. Para asegurar un nivel de

tensión mínimo y estable será necesario utilizar un regulador de tensión antes de conectar la carga. Este regulador debe ser lo más eficiente posible con el fin de maximizar el aprovechamiento de la energía almacenada en los supercondensadores, reduciendo así el uso del buffer secundario.

### 5.3.2 Backup path

Este segundo camino proporcionará energía a la carga cuando el camino principal falla, es decir, cuando no hay energía solar y cuando la tensión de los supercondensadores baja por debajo del voltaje mínimo para funcionar. Estas dos condiciones se deben producir simultáneamente. Como se ha mencionado anteriormente, el camino principal se ha diseñado para funcionar tanto como sea posible, por lo que en condiciones ideales, este camino no debería ser utilizado; sólo funcionará en situaciones excepcionales como un largo periodo de tiempo sin energía solar o consumos extraordinariamente altos. Como este camino se activará muy pocas veces, se reduce el desgaste de su elemento de almacenaje de energía.

Este camino se caracteriza por tener un elemento para almacenar energía con muy alta densidad de energía, al que se ha llamado buffer secundario. Este buffer es una batería que asegura que habrá energía en las condiciones en que el camino principal falle. Como el tiempo de oscuridad es el principal motivo para que entre en funcionamiento este camino, y éste depende de factores como el tiempo, la ubicación geográfica y la climatología, no se puede determinar con exactitud el valor ideal de la batería. Por lo tanto, la solución óptima dependerá del equilibrio coste-tamaño de la batería y de la propia aplicación. El tipo de batería con mayor densidad de energía y menor efecto memoria son las de litio [21][22], por lo que será el elegido para esta aplicación. Como desventaja, estas memorias requieren un método de carga más complejo, por lo que se utilizará un chip cargador que sólo será habilitado cuando el panel solar genera un exceso de energía, es decir, cuando el buffer principal está totalmente cargado y el panel solar mantiene una alta tensión en sus terminales.

Por otro lado, la capacidad del buffer secundario puede ser mucho mayor que la del primario. Por tanto este camino podrá aportar energía a la carga durante mucho más tiempo, en principio varias semanas. Este tiempo debería ser suficiente para que el panel solar volviera a funcionar durante algún momento y recargara el buffer primario. Debido a esta posibilidad de sobredimensionamiento de este elemento, este camino secundario puede permitirse utilizar elementos menos eficientes que el primario y recargarse más lentamente.

## 5.4 Implementación

El diseño descrito anteriormente se ha implementado para suministrar una tensión regulada de 3,3 V a un nodo de una red de sensores inalámbricos. La placa prototipo contiene un panel solar, dos supercondensadores, una batería de ión-litio y los circuitos integrados de gestión de la energía como los reguladores y comparadores que pueden verse en la Figura 5-3.

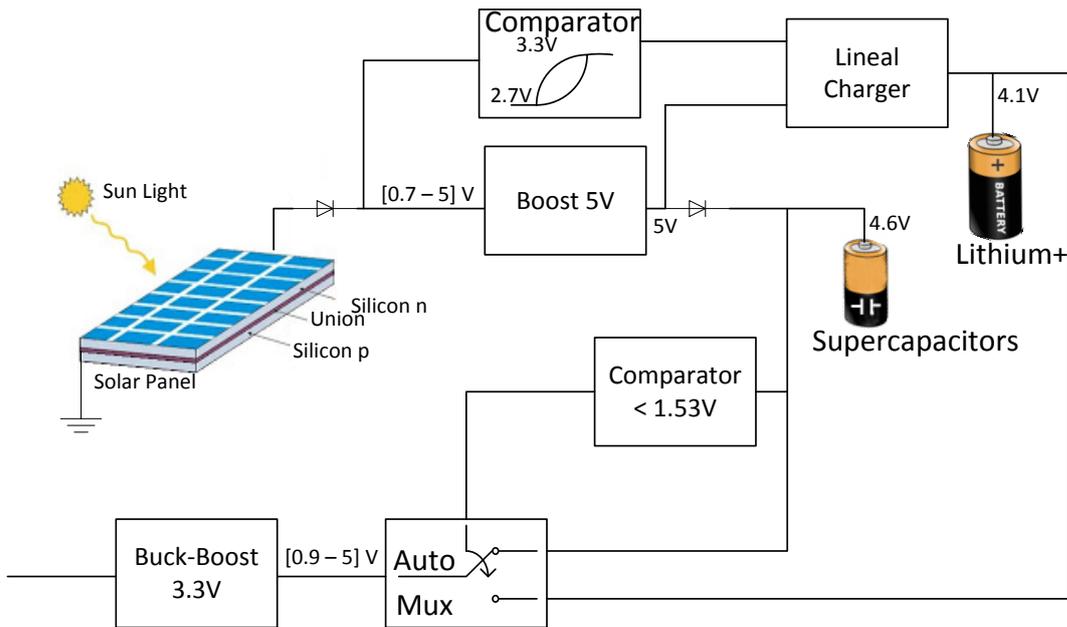


Figura 5-3: Esquema electrónico de la fuente de alimentación y sus niveles de tensión.

La placa proporciona también un conector de 10 pines para la medición de todas las tensiones importantes en el circuito. Este conector se ha utilizado con una tarjeta de adquisición de datos para leer valores que ayudaran a la depuración y evaluación del diseño.

El esquemático completo de este circuito se presenta en la sección 10.2.2, al final de esta tesis.

### 5.4.1 Hardware

Esta sección describe la selección de componentes de cada uno de los bloques de la Figura 5-1 y la Figura 5-3.

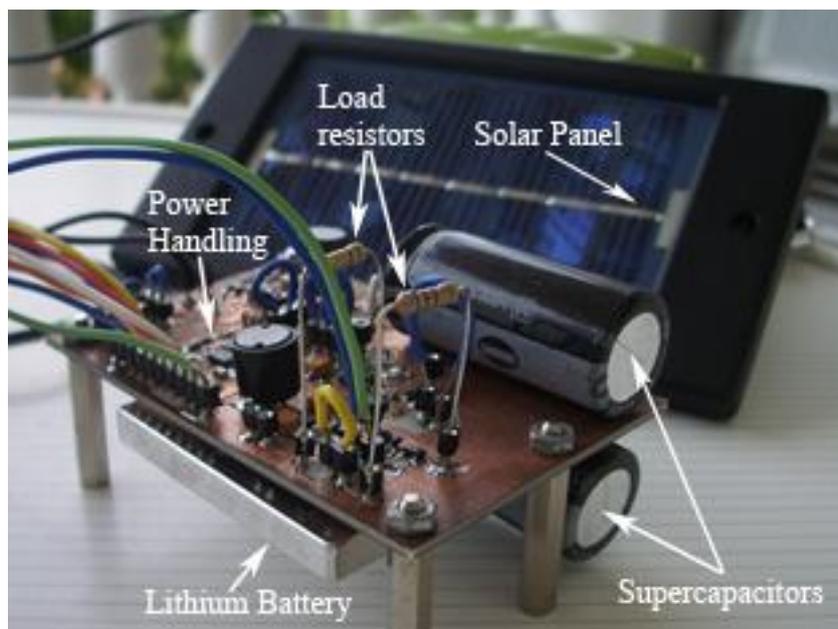


Figura 5-4: Fotografía del prototipo de la fuente de alimentación con resistencias de carga para simular el consumo de un nodo sensor.

### 5.4.1.1 Panel solar

El panel solar utilizado es el MSX-005F (144 x 66 mm) de 0,5 W de Solarex. Fue seleccionado por varios motivos, el principal fue por disponibilidad, entre los disponibles se seleccionaron los que sería mecánicamente viables utilizar en la caja de la aplicación del capítulo 8 por sus dimensiones, y finalmente, porque su rango de tensiones está próximo a los de los componentes que le siguen. Por otra parte, ofrece 150 mA de máxima corriente (con máxima radiación solar), que es mucho mayor de la corriente que necesita el nodo modelo para funcionar (22 mA), por lo que con esa corriente permitiría el funcionamiento del nodo y la posible recarga de los buffers.

En la Figura 5-5 se muestra el resultado de la caracterización del panel solar seleccionado, donde se ha representado su curva de corriente respecto a tensión; se puede ver que el punto de máxima potencia está aproximadamente en 3,4 V.

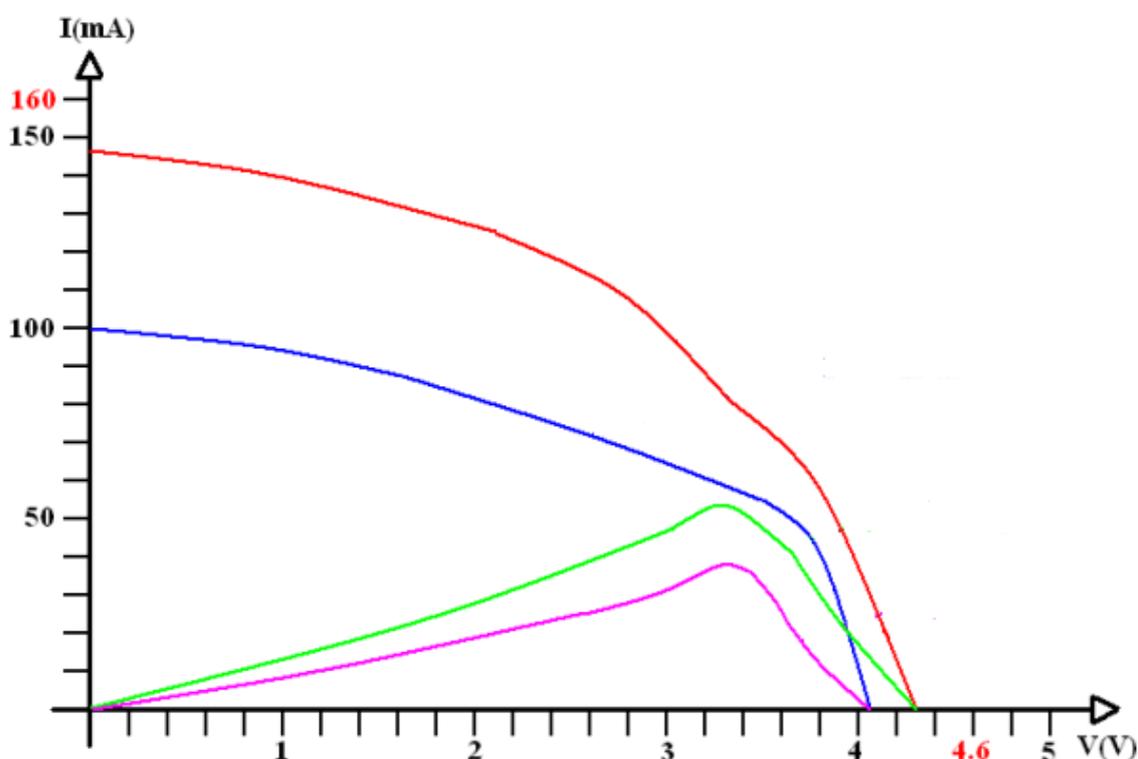


Figura 5-5: Curvas de corriente y potencia para el panel solar seleccionado. Curva roja: corriente y tensión con buenas condiciones ambientales; curva azul: corriente y tensión un día nublado; curva verde: potencia escalada por 10 con buenas condiciones ambientales; curva morada: potencia escalada por 10 un día nublado.

### 5.4.1.2 Adaptador de niveles de tensión

Se utilizará el elevador MAX1795 entre el panel solar y los supercondensadores. Así se alcanzará la tensión requerida para cargar la batería de litio (4,35 V) y también se tiene un nivel que permite aprovechar eficientemente los supercondensadores. El convertidor eleva la tensión a 5 V y permite que la corriente fluya hacia los supercondensadores, incluso cuando la luz solar es débil.

Durante la fase de carga, este regulador tenderá a imponer 5 V a su salida, pero la gran capacidad que tiene conectada hace que necesite un tiempo para alcanzarla. El comportamiento que sigue es el típico de una fuente de alimentación cargando un circuito RC

hasta llegar a los 5 V, pero con las limitaciones propias del regulador, como la máxima corriente que puede suministrar (teóricamente 0,25 A, por lo que el límite real es la corriente entregada por el panel solar). A su entrada, la tensión seguirá el voltaje del panel, como se ve en la Figura 5-6 (a) en la línea de color naranja.

Además se aprovechará un comparador que tiene integrado este circuito para construir un comparador con histéresis para habilitar la carga de la batería únicamente cuando el panel solar tenga en sus bornes más de 3,3 V (y hasta que caiga por debajo de 2,7 V), es decir, cuando haya buenas condiciones que permitan hagan pensar que se podrá mantener una carga con suficiente corriente durante cierto tiempo.

#### **5.4.1.3 Buffer principal (supercondensadores)**

Dado que cada condensador sólo admite 2,3 V, se conectarán dos en serie para cargarlos a 4,6 V y reducir las corrientes de fugas. Cuanto mayor es la capacidad del supercondensador, menor es su corriente de fugas, pero el precio aumenta mucho más.

Al conectar dos condensadores de gran capacidad en serie se debe hacer un balanceo o equilibrado para asegurarse que la tensión en el punto medio es la mitad, porque si no podría darse el caso que un condensador se descargue más que el otro, lo cual es indeseable. En este caso se ha realizado un balanceo pasivo por medio de un divisor de resistencias. El valor de estas resistencias debe conseguir un equilibrio entre la velocidad del balanceo y la corriente de pérdidas que habrá por ellas, experimentalmente se han seleccionado resistencias de 10 M $\Omega$ .

En este diseño se han utilizado dos supercondensadores de 50 F de PANASONIC [24] debido a la disponibilidad y el precio. Al estar en serie, la capacidad equivalente será la mitad, 25 F. La resistencia equivalente de fugas equivalente de 50 m $\Omega$  y presentan una ESR de 0,1  $\Omega$  a 1 kHz.

#### **5.4.1.4 Multiplexor**

El multiplexor para seleccionar el canal de la corriente es el Texas Instruments TPS2113PW. La principal característica de este circuito es su baja resistencia en directa, de 84 m $\Omega$ , además de un consumo muy bajo y diodos de bloqueo de corrientes inversas y cruzadas.

El circuito integrado del multiplexor lleva incorporado un comparador con histéresis que permite conmutar sus señales automáticamente. Se cambiará el canal entre el camino principal y el secundario en función de la tensión de los dos supercondensadores cuando ésta sea menor de 1,5 V, ya que es una tensión próxima a la tensión mínima de entrada del convertidor posterior.

#### **5.4.1.5 Bloque de control**

El bloque de control está en realidad distribuido entre otros bloques, por este motivo aparece con líneas punteadas en la Figura 5-1. Está compuesto por un comparador utilizado en el bloque del adaptador de nivel y otro comparador para conmutar el multiplexor. El primer comparador, el del elevador de tensión, se utiliza para permitir la carga de la batería de Litio. Esta carga sucede cuando la tensión medida en el panel solar está por encima de 3,3 V (MPP), y se deshabilita por debajo de 2,7 V. Este comparador tiene un margen de histéresis muy amplio para evitar una falsa carga, la cual puede producirse por la caída de tensión cuando la batería comienza a recargarse.

El segundo comparador está en el bloque del multiplexor. Este comparador cambiará automáticamente las entradas del multiplexor para conmutar entre los dos caminos de corriente, dando prioridad al camino principal. Mediante este mecanismo se consigue que el camino secundario únicamente sea seleccionado cuando el nivel de tensión en los supercondensadores esté por debajo de un umbral mínimo. En este caso el umbral es de 1,53 V. Esta comparación también tiene una pequeña histéresis, de alrededor de 60 mV, para evitar falsas conmutaciones entre las entradas.

### **5.4.1.6 Regulador de tensión**

Este bloque está compuesto por un chip con función de convertidor y elevador de tensión. La salida es una tensión fija de 3,3 V. Se utiliza el TPS61025 de Texas Instruments. Este chip acepta un amplio rango de tensiones de entradas, de 0,9 a 6,5 V, lo que permitirá aprovechar más la tensión en los supercondensadores.

Este tiene una eficiencia muy plana (entre 90-94%) incluso a bajas corrientes (de 1 mA a 40 mA) y un modo y un modo llamado *Power Save Mode* para mantener la eficiencia cuando la carga es muy pequeña, mejorando la caída de eficiencia para bajas corrientes propias de los sistemas basados en PWM.

### **5.4.1.7 Buffer secundario (batería)**

Se ha utilizado una batería de Ión-litio (Li+) de 1 Ah debido a su disponibilidad y bajo precio.

### **5.4.1.8 Control de carga**

Como la batería se va a utilizar en exteriores, se necesita hacer un proceso de limitación de corriente de carga para ser seguro. En esta aplicación se ha utilizado el chip MAX1811, el cual limita la corriente y protege la batería. Se ha seleccionado este tipo de cargador lineal porque es más rápido que otros de tipo conmutado en cargar la batería.

Además, mediante un comparador de tensión en el chip del adaptador de niveles (MAX1795), y un divisor de resistencias externo a éste, R5 y R9, se habilitará o deshabilitará el cargador de la batería. Además se ha incluido una histéresis por medio de R8 y Q2 para evitar recargas triviales que podrían hacer envejecer a la batería en caso que el panel no pudiera proporcionar la suficiente energía mantener la recarga.

## **5.4.2 Carga**

En la placa prototipo del sistema de carga se han incluido dos MOSFET, cada uno con una resistencia entre su drenador y la salida de tensión. El valor de las resistencias se calculado para obtener las corrientes de carga de la Figura 5-2. De esta manera es posible cambiar digitalmente las corrientes que demandaría la carga en los diferentes modos de trabajo (22 mA y 6  $\mu$ A) para realizar una simulación.

La conmutación de los MOSFETs se gestiona digitalmente por un microcontrolador situado fuera e independientemente de la fuente, utilizado exclusivamente para este propósito y alimentado con su propia fuente.

### 5.4.3 Sistema de adquisición

Al conector con las tensiones en diferentes puntos del circuito se conectará una tarjeta de adquisición de datos NI USB-6008 de National Instruments para verificar que las tensiones son correctas. Dado que las tensiones en la fuente de alimentación no varían rápidamente, se utilizará una frecuencia de muestreo de 1 Hz.

El sistema de adquisición está conectado a un PC por medio de un cable USB que, además, la alimenta. Además, un programa en LabVIEW almacena y representa los datos adquiridos.

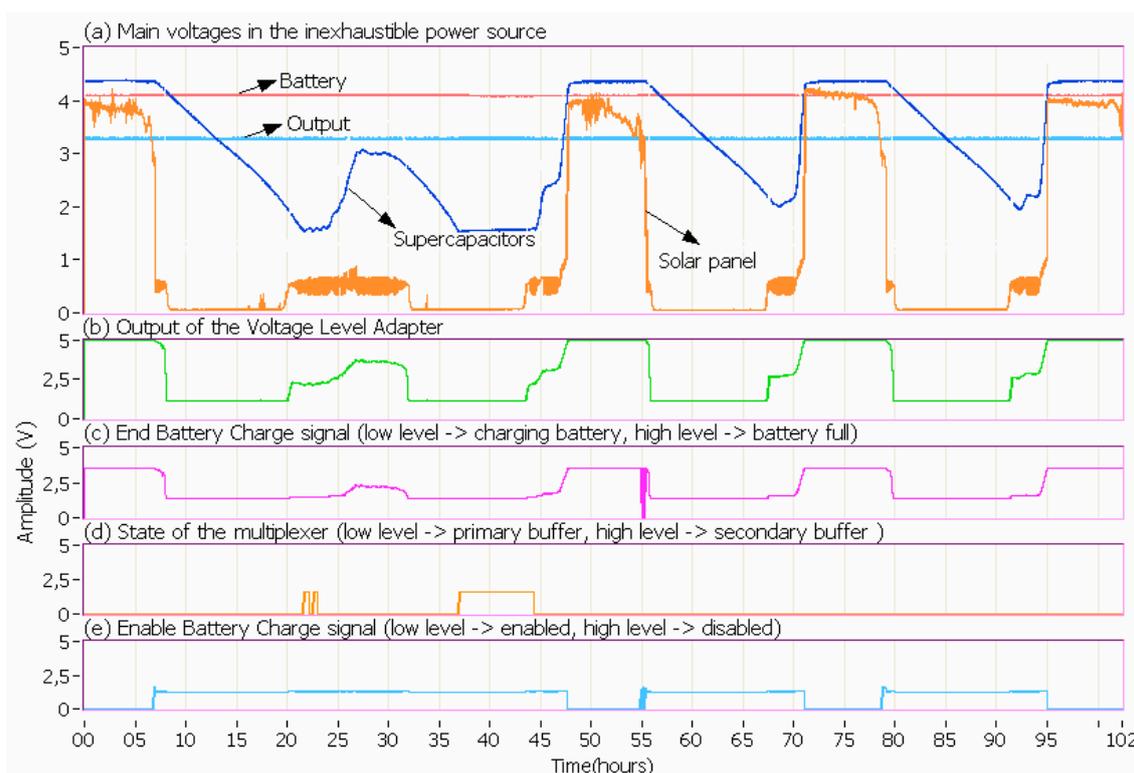
## 5.5 Resultados y discusión

El sistema desarrollado se ha ubicado en una azotea durante varios días. Se ha utilizado un ordenador portátil conectado al sistema de adquisición muestreando cada segundo para probar la fuente de energía, muestreando las tensiones cada segundo los siguientes nodos:

- Terminales del panel solar.
- Señal de fin de la carga de la batería.
- Salida de la tensión del adaptador de niveles.
- Habilitación de la carga de la batería.
- Tensión en los supercondensadores.
- Estado del multiplexor.
- Salida de la fuente de alimentación.
- Tensión en la batería.

La Figura 5-6 se muestra la evolución de las tensiones en los puntos anteriores durante cinco días consecutivos. Se suministran 3,3 V a la carga simulada, utilizado un ciclo de trabajo de 4%. Las mediciones comienzan el 22 de septiembre a las 12:00 del mediodía con pleno sol. Los supercondensadores se recargan totalmente a 4,35 V y la batería comienza completamente cargada en 4,1 V (Figura 5-6 (a)). En ese momento el panel solar suministra energía directamente a la carga a través del camino principal. Después de siete horas el sol comenzó a ocultarse y los supercondensadores asumieron el papel de alimentación principal. Los supercondensadores alimentaron a la carga durante la primera noche.

El segundo día amaneció lloviendo, por lo que la radiación solar era demasiado débil para reponer los supercondensadores. Como éstos siguieron alimentando a la carga debido a la baja energía suministrada por el panel solar, alcanzaron el umbral de tensión mínimo de 1,53 V (22 horas desde el comienzo de la prueba) y se conmutó al camino secundario, como puede verse en la Figura 5-6 (d). Eventualmente, la energía residual que generaba el panel solar cargó los supercondensadores como se aprecia en la Figura 5-6 (b), pero la carga era más lenta, como se aprecia en la pendiente de la tensión, y no pudo llegar a la tensión máxima antes de que volviera a anochecer. Así, los supercondensadores comenzaron la segunda noche a 3 V y mantuvieron la energía de la carga hasta que volvieron a llegar a la tensión mínima, por lo que la batería entró de nuevo en funcionamiento.



**Figura 5-6: Evolución de los niveles de tensión en los puntos muestreados.**

El tercer día volvió a salir el sol a las 8 de la mañana, en esta ocasión fue un día soleado. Los supercondensadores llegaron rápidamente a su tensión máxima de 4,35 V en menos de tres horas. Por otra parte, la batería de litio también se recarga para recuperar la energía perdida la noche anterior, como puede apreciarse en la Figura 5-6 (c). En la Figura 5-6 (e) se muestra cuando se habilita la recarga de la batería, pero debe advertirse que la habilitación no implica recarga, sino que ésta depende del chip cargador de la batería, de la carga previa de la batería y de la tensión de salida del adaptador de niveles de tensión. La carga de la batería realmente se aprecia en (c) cuando el nivel de esta señal es 0 hacia la hora 55, lo que indica que se está cargando; un nivel distinto de 0 indica que el cargador detecta que está cargada y no le inyecta corriente.

Los siguientes dos días fueron soleados. La fuente de alimentación mantuvo la alimentación de la carga sin utilizar la batería de litio, únicamente se utilizaron los supercondensadores. Este es el modo de funcionamiento deseable de la fuente, que es el que evita el desgaste de la batería y alarga su vida útil.

En la Figura 5-7 se muestra una segunda medida llevada a cabo para evaluar el tiempo de recuperación de la batería de litio. Inicialmente se dejaron los supercondensadores y la batería a 1,57 V y 3,67 V respectivamente. Después se expuso el panel solar a la radiación solar durante tres días. En la figura se muestra la evolución de la tensión en los supercondensadores y en la batería mientras se alimenta a una carga con el 4% del ciclo de trabajo.

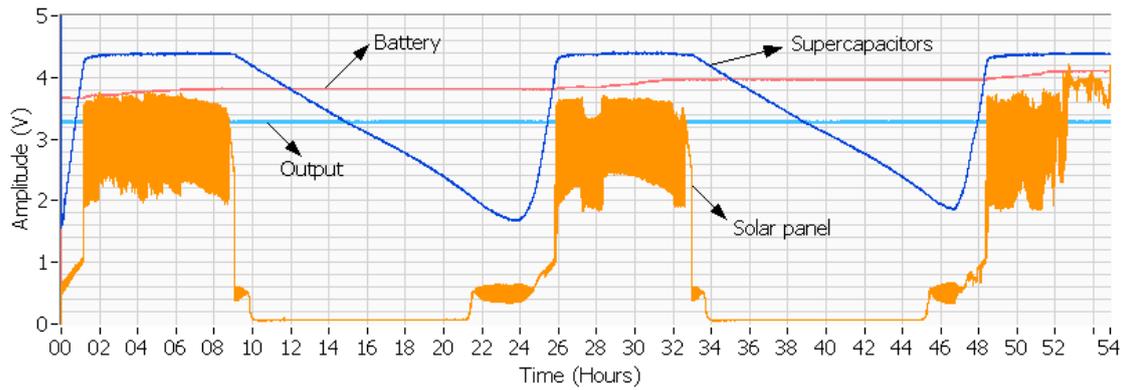


Figura 5-7: Recuperación de la batería de litio después de una descarga profunda intencionada.

En primer lugar, la energía del panel solar se utiliza para recargar totalmente los supercondensadores, desde 1,57 a 4,35 V, lo que necesitó menos de 2 horas. Después de esto, aunque el segundo día era nublado, la batería pasó de 3,67 V a 4,1 V en dos días y medio mientras también se alimentaba la carga. Estas condiciones demuestran que la fuente de energía puede trabajar incluso en condiciones muy desfavorables.

Finalmente, se llevaron a cabo otros experimentos para evaluar la longevidad de la fuente de alimentación diseñada. Uno de ellos fue adquirir todas las tensiones en los supercondensadores durante varios días y noches con distintos ciclos de trabajo. La información más interesante se obtiene por las noches, ya que es cuando los supercondensadores son la única fuente de energía que alimenta a la carga. En la Figura 5-8 se muestra la evolución de la tensión en los supercondensadores durante cuatro noches consecutivas con ciclos de trabajo del 1, 3, 4 y 5%. Se parte de una carga completa de los supercondensadores y se han superpuesto las cuatro curvas de una misma noche a la misma hora. En esta gráfica se puede determinar que el ciclo de trabajo del 4% es el máximo en el cual la fuente no necesita utilizar la batería, ya que en el 5% se alcanza el umbral mínimo del 1,53 V para las condiciones solares en las que se ha probado.

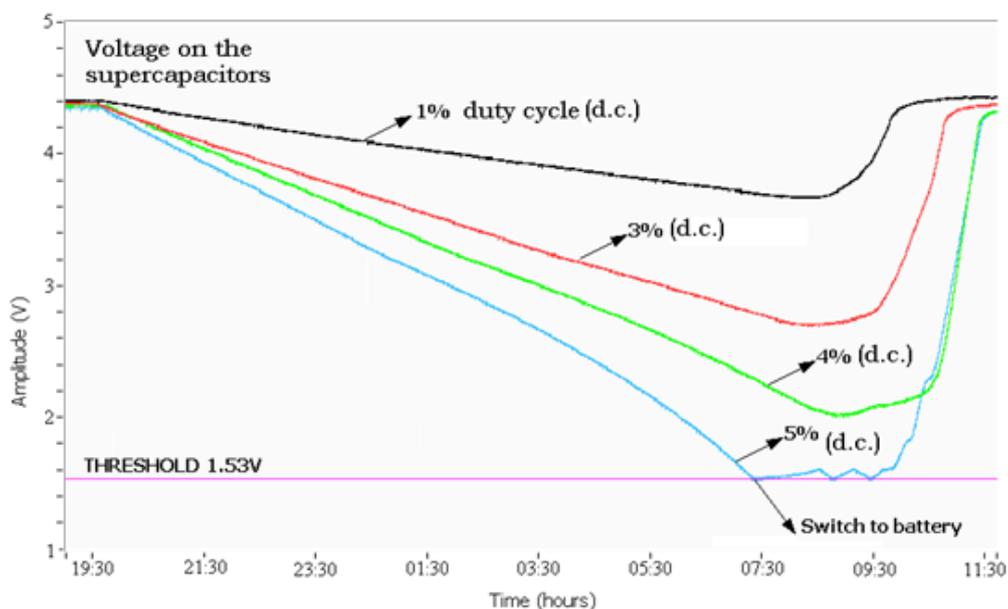


Figura 5-8: Descarga de los supercondensadores durante 4 noches superpuestas a diferentes cargas de trabajo.

Por otra parte, sería posible que la aplicación del nodo sensor utilizara un ciclo de trabajo variable para adaptar su consumo a la energía disponible, así la fuente diseñada podría funcionar con ciclos de trabajo superiores al 4% durante el día, cuando se alimenta directamente desde el panel solar, y se reduzca durante la noche.

La predicción del tiempo de vida es una tarea complicada porque depende de muchos parámetros, algunos como las condiciones climatológicas con bastante incertidumbre. Sin embargo se espera un ciclo de vida mucho mayor a 3 años, ya que es la duración típica de las baterías de litio si se utilizan diariamente [21].

Además, se espera que la batería no llegue a descargarse totalmente nunca, garantizando así el funcionamiento ininterrumpido del nodo. Si se aplica un ciclo de trabajo del 4%, se tendría un consumo de corriente medio de  $886 \mu\text{A}$ , como se obtiene en la ecuación (1). Con este consumo de corriente, asumiendo que no hay pérdidas y eficiencias ideales en todos los componentes y utilizando los valores de dos supercondensadores del buffer principal, se puede hacer una estimación del tiempo que los supercondensadores podrían alimentar el nodo utilizando la ecuación 2 [29]. Una batería de 1 Ah, sin ninguna recarga por parte del panel solar, podría mantener el nodo en funcionamiento durante 47 días (ecuación 3).

$$I_{\text{load}} = \frac{(4 \cdot 22\text{mA}) + (96 \cdot 6\mu\text{A})}{100} \approx 886\mu\text{A} \quad (5.1)$$

$$t_{\text{supercap}} = \frac{C \cdot (V_{\text{max}}^2 - V_{\text{min}}^2)}{I_{\text{load}} \cdot (V_{\text{max}} + V_{\text{min}})} = 1\text{day} \quad (5.2)$$

$$t_{\text{battery}} = \frac{1\text{Ah}}{886\mu\text{A} \cdot 24\text{hours}} \approx 47\text{days} \quad (5.3)$$

donde:

$C$  es la capacidad nominal de los supercondensadores en Faradios.

$I_{\text{load}}$  es la corriente media suministrada a la carga.

$V_{\text{max}}$  y  $V_{\text{min}}$  son los umbrales máximo y mínimo de la tensión.

$t$  es el tiempo que se mantendría el dispositivo en funcionamiento.

La vida útil estimada del nodo alimentado únicamente con la batería es de 47 días. En este tiempo se asume que habrá algún momento en el que el panel solar podría volver a generar energía para alimentar el nodo, recargar los supercondensadores y la batería y extender este límite llegando a una duración casi perpetua en lo que respecta a la vida de los mecanismos de suministro de energía.

No se han considerado las eficiencias del TPS61025 porque no se dispone tal valor para una corriente de  $6\mu\text{A}$ . Además, el modelo SPICE proporcionado por el fabricante no permite calcular la eficiencia en una simulación para esos niveles de corriente. Sin los datos reales, se puede presuponer un caso que se estima desfavorable, tal como eficiencias del 90% y 40% para los estados activo y dormido. En tal caso se tendría una reducción del tiempo de autonomía, tanto para el supercondensador como para la batería, de entorno al 10,8%.

La eficiencia ante variaciones de tensión, de acuerdo al datasheet, es relativamente baja, especialmente para las corrientes a las que trabajará el sistema. La variación estaría en torno a un 2% al pasar de 2,4 a 1,8 V en 20 mA, y en ambos casos por encima del 90%.

En el capítulo 8 se realizará una aplicación mediante un nodo sensor que implementará una estación meteorológica inalámbrica. Dicho nodo se alimentará utilizando la fuente de alimentación que se ha presentado aquí.

## 5.6 Conclusiones

En este capítulo se ha discutido un método de diseño e implementación de una fuente de alimentación solar perpetua que no requiere que ninguna intervención humana controle y recargue ninguno de sus dos sistemas de almacenamiento. Tampoco requiere el uso de una lógica programable que esté continuamente monitorizando el estado, ya que únicamente tiene componentes electrónicos discretos.

Después de varias semanas de funcionamiento continuo, la salida de 3,3 V nunca ha fallado y la batería sólo ha tenido que funcionar una vez en la que hubo un día totalmente lluvioso. Se ha demostrado que la operación autónoma de nodos sensores con un consumo medio de  $886\mu\text{A}$ ; este valor corresponde con una carga que trabaje al 4% del ciclo de trabajo fija, no obstante también podrían utilizarse nodos con un ciclo de trabajo mayor si éste se ajusta dinámicamente. Por ejemplo, si el nodo sensor monitoriza la energía disponible en la fuente podría reducir su actividad durante las noches y aumentarla durante el día, donde probablemente haya un exceso de energía. Sin embargo, el ciclo de trabajo depende de una aplicación concreta y la fuente de alimentación diseñada puede cubrir un amplio espectro de aplicaciones donde los ciclos de trabajo suelen ser del 1 % o menores [26, 29-30].

La nueva arquitectura del sistema, en la cual esta fuente de alimentación está basada, permite explorar también otras nuevas fuentes de energía porque el uso adecuado de supercondensadores como buffer primario puede eliminar la necesidad de grandes y degradables baterías.

Se pueden estudiar algunas modificaciones o mejoras, como la posibilidad de desconectar el regulador de salida cuando el nodo está dormido y que éste se alimente mediante un condensador si la carga tuviera un periodo de trabajo regular y conocido. Con esto en lugar de reducir un 10,8% la autonomía al considerar las eficiencias presupuestas anteriormente, se pasaría a 10%, y ver si con el consumo de la circuitería añadida permitiría obtener algún beneficio. Otra alternativa cuando no se conozca el ciclo de trabajo a priori sería conmutar a otro regulador de una tecnología más eficiente para bajas corrientes.

Por último, con una revisión actualizada (2014) de este trabajo (2007), también se considera que se mejoraría el diseño empleando soluciones tales como los circuitos integrados BQ24210 (datado del año 2011) o el LTC3105 (del año 2010). Estos circuitos integran seguidores del punto de máxima potencia de paneles solares, lo que permite mantenerlos trabajando en su punto óptimo. Otro elemento actual que se podría introducir en el diseño de esta fuente de alimentación serían las baterías de estado sólido como las de cymbet (empezó a comercializar en 2009), excellaton, infinity power solution o front edge technology. Se trata de baterías encapsuladas en chips de baja capacidad (equivalente a condensadores del orden de milifaradios), bajas pérdidas por autodescarga, recargables y alta velocidad de carga, lo que los hace adecuados para combinarse con sistemas de *energy harvesting* y aplicaciones de bajo consumo.

### 5.7 Bibliografía

- [1] Raghunathan, V.; Kansal, A.; Hsu, J. & Friedman, J.; Srivastava, M. (2005). Design Considerations for Solar Energy Harvesting Wireless Embedded Systems, IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 457-462, Los Angeles, CA, USA, April 25-27, 2005.
- [2] Roundy, S.; Steingart, D.; Frechette, L.; Wright, P. K.; Rabaey, J. M. (2004). Power Sources for Wireless Sensor Network, Proc. 1st European Workshop on Wireless Sensor Networks (EWSN), Berlin, Germany, 2004.
- [3] Park, C.; Chou, P. H. (2004). Power Utility Maximization for Multiple Supply Systems by a Load Matching Switch. IEEE International Symposium on Low Power Electronics and Design (ISLPED), pp. 168–173, Newport Beach, California, USA, 2004.
- [4] Park, C.; Liu, J.; Chou, P.H. (2005). Eco: An ultra-compact low-power wireless sensor node for real-time motion monitoring, IEEE International Conference on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, April 25-27, 2005.
- [5] Hande, A.; Polk, T.; Walker, W.; Bhatia, D. (2006). Indoor Solar Energy Harvesting for Sensor Network Router Nodes, Journal of Microprocessors and Microsystems Special Issue on Sensor Systems, 2006.
- [6] Chou, Pai H.; Park, C. (2005). Energy-Efficient Platform Designs for Real-World Wireless Sensing Applications, IEEE International Conference on Computer Aided Design, (ICCAD), San Jose, CA, 2005.
- [7] Kansal, A.; Potter, D.; Srivastava, M.B. (2004). Performance Aware Tasking for Environmentally Powered Sensor Networks, ACM Joint International Conference on Measurement and Modelling of Computer Systems (SIGMETRICS), New York, USA, 2004.
- [8] Alberola, J.; Pelegri, J.; Lajara, R.; Perez, Juan J. (2008). Solar Inexhaustible Power Source for Wireless Sensor Node, IEEE International Instrumentation and Measurement technology Conference, I2MTC 2008, Victoria-Canada.
- [9] Arms, S.W.; Townsend, C.P. & Churchill, D.L.; Galbreath, J.H.; Mundell, S.W. (2005). Power Management for Energy Harvesting Wireless Sensors, International Symposium on Smart Structures & Smart Materials (SPIE), 9 March 2005, San Diego, CA.

- [10]Rahimi, M.; Shah, H.; Sukhatme, G.S.; Heideman, J.; Estrin, D. (2003). Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network, Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 19-24, Taipei, Taiwan, May 2003.
- [11]Hsu, J.; Friedman, J.; Raghunathan, V.; Kansal, A.; Srivastava, M. (2005). Helimote: Enabling self-sustained wireless sensor networks through solar energy harvesting, International Symposium on Low Power Electronics and Design (ISLPED), San Diego, California, August 8-10, 2005.
- [12]Dreher, R. (2004). Remote Observation Station, Circuit Cellar: The magazine for computer applications, Issue 162 January 2004.
- [13]Panasonic, (1999). Solar Cells Technical Handbook 98/99. <http://downloads.solarbotics.com/PDF/sunceramcat.pdf>
- [14]Jiang, X.; Polastre, J.; Culler, D. (2005). Perpetual Environmentally Powered Sensor Networks. IEEE International Conference on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, April 25-27, 2005.
- [15]Roundy, S.; Otis, B. P.; Chee, Y.; Rabaey, J. M.; Wright, P. (2003). A 1.9GHz RF Transmit Beacon using Environmentally Scavenged Energy, IEEE International Symposium on Low Power Electronics And Devices (ISLPED), Seoul, Korea, 2003.
- [16]Polastre, J.; Szewczyk, R.; Sharp, C. & Culler, D. (2004). The Mote Revolution: Low Power Wireless Sensor Network Devices, Hot Chips 2004, Aug 22-24, 2004.
- [17]Würfel, P. (2005). Chapter 1. Introduction of Physics of Solar Cells: From Principles to New Concepts, 2005 WILEY-VCH Verlag gmbh & Co. kgaa, Weinheim. ISBN: 3-527-40428-7.
- [18]Chuck Wright Consulting, LLC, (2006). The Solar Sprint PV Panel, website. <http://www.chuck-wright.com/SolarSprintPV/SolarSprintPV.html> Cooper Industries, (2006), Aerogel Supercapacitors – b series datasheet. <http://www.cooperet.com/products/products.cfm?page=supercapacitors>
- [19]Koutroulis, E.; Kalaitzakis, K.; Voulgaris, N. C. (2001). Development of a Microcontroller- Based, Photovoltaic Maximum Power Point Tracking Control System, IEEE Trans. Power Electronics, vol. 16, no. 1, Jan 2001.
- [20]Lim, Y. H. & Hamill, D.C. (2000). Simple maximum power point tracker for photovoltaic arrays. Electronics Letters, IEE 2000, 24 February 2000.
- [21]Buchmann, I. (2005). How to prolong lithium-based batteries (BU34), Cadex Electronics Inc., Vancouver. <http://www.batteryuniversity.com/parttwo-34.htm> Burke, A. (2000). Ultracapacitors: why, how, and where is the technology, Journal of Power Sources, Vol. 91, Issue 1, Nov. 2000, pp. 37-50.
- [22]MPower Solutions for All Your Specialist Portable Power Needs, (2006). <http://www.mpoweruk.com/performance.htm>
- [23]A. Burke. Ultracapacitors: why, how, and where is the technology, Journal of Power Sources, Vol. 91, Issue 1, Nov. 2000, pp. 37-50.
- [24]Panasonic. EECHW0D506 supercap 50F, 2.3V datasheet. Online document, available in <http://industrial.panasonic.com/www/cgi/jvcr13pz.cgi?E+PZ+3+ABC0008+EECHW0D506+7+WW>
- [25]Naik, J. Performing Accurate PFM Mode Efficiency Measurements. Texas Instruments. Application Report (2006) Online document, available in <http://www.ti.com/lit/an/slva236/slva236.pdf>

- [26]Cooper Industries, Aerogel Supercapacitors – b series datasheet.
- [27]Cooper Industries. Supercapacitors. Online document, available in <http://www.cooperet.com/products/products.cfm?page=supercapacitors>
- [28]Zhang, P.; Sadler, C.M.; Lyon, S.A.; Martonosi, M. (2004). Hardware Design Experiences in ZebraNet, ACM Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, Maryland, USA, November 3–5, 2004.
- [29]Stor, P.; Bussmann C. (2007) Application Note: Design Considerations In Selecting Aerogel Supercapacitors, <http://www.cooperbusmann.com/3/PowerStor.html>
- [30]Werner-Allen, G.; Johnson, J.;Ruiz, M.; Lees, J.; Welsh, M.; Marcillo, O. (2005). Monitoring Volcanic Eruptions with a Wireless Sensor Network, Second European Workshop on Wireless Sensor Networks (EWSN), Istanbul, Turkey, January-February, 2005.
- [31]Noda, C.; Fernández, J.; Pérez, C.; Altshuler, E. (2006). Measuring Activity in Ant Colonies: The Wireless Perspective, International Centre for Theoretical Physics (ICTP), February 2006, Trieste, Italy.

# 6. VIABILIDAD DE RECOLECCIÓN DE ENERGÍA RF PARA WSN

---

En este capítulo se tratará la viabilidad de la posibilidad de extraer energía de señales de estaciones de radio y televisión comerciales. Este método evitaría tener una estación de emisión de señal de RF que alimentase a los nodos de una red sensora, los cuales deben diseñarse con un consumo energético extremadamente bajo.

El flujo de energía obtenido es ininterrumpido, este flujo podrá cargar un condensador por medio de un rectificador y multiplicador de tensión. La corriente media obtenida llega hasta 8  $\mu\text{A}$  en la frecuencia probada de 1,584 MHz de una emisora de radio AM.

Los resultados preliminares indican que el condensador cargado podría alimentar un nodo sensor de funcionamiento pulsado con un bajo ciclo de trabajo.

## 6.1 Introducción

En la actualidad el uso de dispositivos inalámbricos como teléfonos móviles, routers wifi o redes de sensores va en aumento. Muchas aplicaciones necesitan que los nodos trabajen de forma móvil y autónoma. Muchos de estos dispositivos utilizan baterías recargables. Los dispositivos de mano no suelen tener problemas para recargar sus baterías porque el usuario se encarga de ello. No obstante, otros dispositivos como los nodos de las redes de sensores pueden tener dificultades para que una persona recargue sus baterías, ya sea porque están localizados en lugares remotos o inaccesibles, por el alto número de nodos, etc. Por tanto sería deseable que estos dispositivos pudieran trabajar de forma autónoma durante muchos años sin intervención humana. Esto implica una serie de problemas, como el reemplazo o recargas de las baterías, además del peso y tamaño de éstas. El problema se incrementa si además el número de nodos en la red es elevado y están distribuidos en un área amplia o lugares inaccesibles. Por tanto sería deseable disponer de un método viable para alimentar a los nodos sensores en cualquier situación y lugar. En los últimos años el área de investigación llamada *energy harvesting* [1] está tratando de responder a la necesidad que se acaba de mencionar.

En la actualidad el uso de sistemas digitales y sensores con un consumo extremadamente bajo ha hecho posible el desarrollo de aplicaciones de muy bajo consumo. Estos dispositivos son idóneos para utilizar sistemas de *energy harvesting*.

Existen estudios sobre recolección de energía en frecuencias altas, desde 2 GHz a 18 GHz, o frecuencias en la banda ISM de 2,4 GHz [2][3]. Además, existen tecnologías como RFID que

usan una tecnología similar en la frecuencia de 13,56 MHz o en frecuencias UHF-ISM [4][5][6]. Muchos de estos sistemas utilizan altas frecuencias y, por tanto, la distancia entre el emisor y receptor es relativamente baja [7]. Además en el caso de las bandas ISM la potencia de transmisión está limitada.

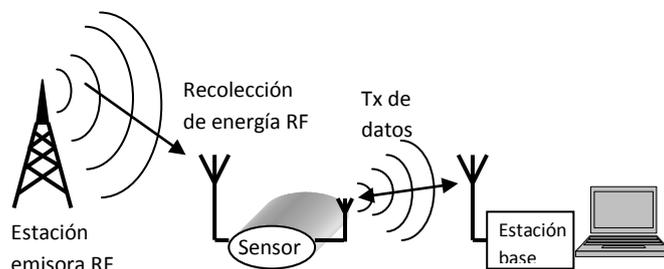


Figura 6-1: Esquema conceptual del sistema de recolección de energía sobre un nodo sensor.

Generalmente el tipo de dispositivos que se alimentan con energía RF sigue el esquema de la Figura 6-1. Este sistema está compuesto por una antena, un circuito de adaptación de impedancias, un circuito rectificador, un filtro paso bajo, un elemento de almacenamiento y una unidad de control [2][8]. Este tipo de dispositivos también se denominan «rectenna», la cual se muestra esquemáticamente en la Figura 6-2. Muchas rectenas y sus circuitos de adaptación de impedancias se han diseñado para tener un gran ancho de banda [9][10].

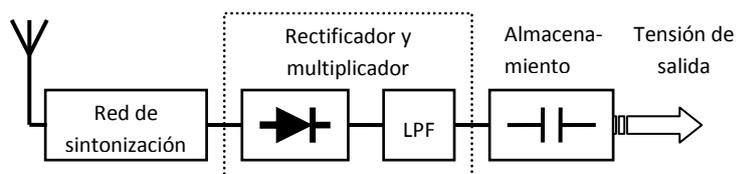


Figura 6-2: Diagrama de bloques de una rectena.

En este capítulo se propone el uso de la energía de una estación emisora comercial de RF como la de radio o televisión para alimentar un nodo sensor de una red inalámbrica. Este método de alimentación puede ser especialmente interesante para nodos colocados en lugares remotos donde otras fuentes de energía como la solar o el viento no están disponibles. No obstante, este mecanismo se puede completar con otros sistemas como los mencionados antes ya que los anteriores, si están disponibles, son más eficientes pero no garantizan su disponibilidad.

## 6.2 Implementación del sistema

Se propone un sistema basado en el principio de las rectenas, aunque con algunas peculiaridades. A continuación se describen las partes del sistema, haciendo especial hincapié en la del doblador de tensión por ser la más interesante para esta aplicación.

### 6.2.1 Antena

La antena es una de tipo L. Este tipo de antenas es un dipolo en el que ambos brazos son perpendiculares entre sí, y presentan un buen rendimiento para la simplicidad de su montaje.

En nuestro caso se ha utilizado una antena de tipo L en horizontal para  $\lambda/4$ . La longitud total es de unos 8 metros de longitud, con 5 metros para un brazo y 3 para el otro. En la Figura 6-4 se puede ver una simulación de la antena con las dimensiones anteriores para la frecuencia de trabajo.

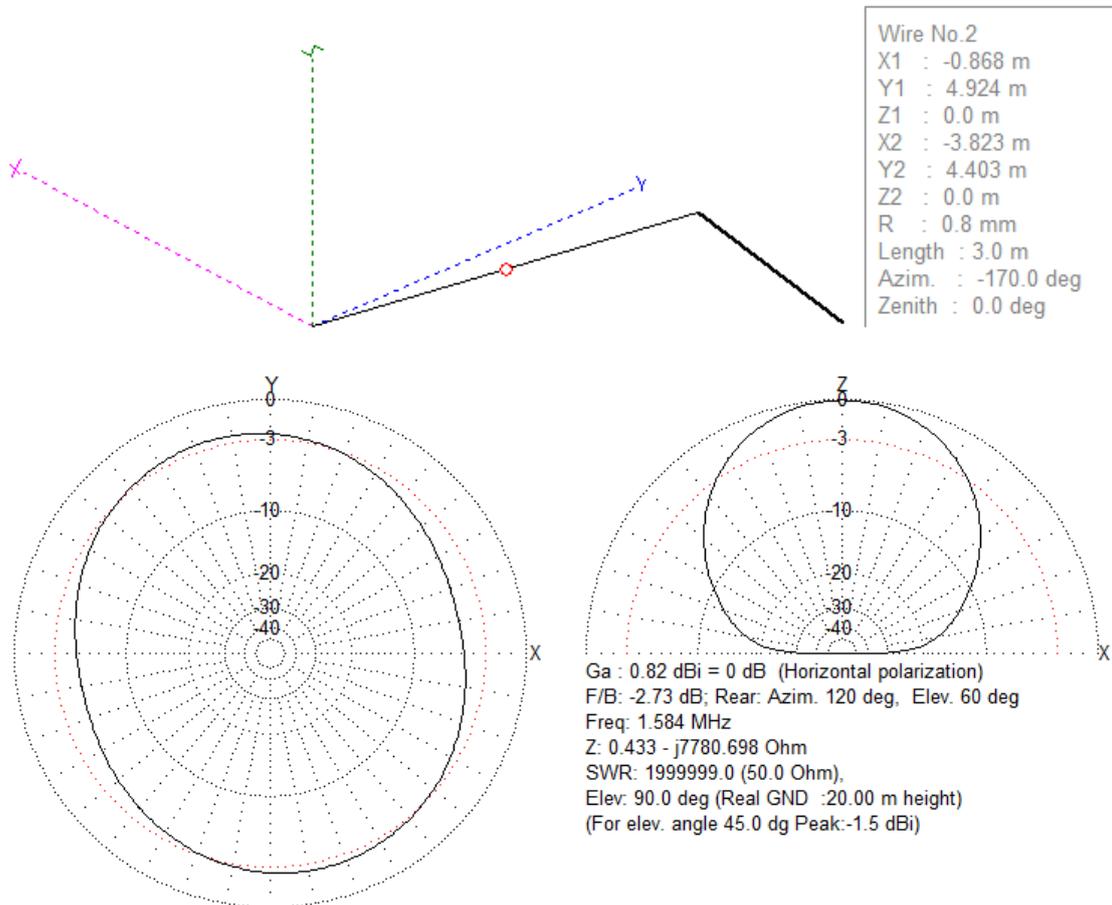


Figura 6-3: Simulación de una antena en L de las dimensiones dadas para 1,584 MHz. Arriba: geometría de la antena; abajo: diagrama de radiación.

## 6.2.2 Etapa de sintonización

La antena está conectada a una etapa de sintonización porque sólo se utilizará un canal de todos los disponibles. El canal se ha seleccionado porque es uno de los que más potencia tiene en el lugar donde se realizarán las pruebas. La frecuencia es de 1,584 MHz, en el que se transmite una señal AM de una emisora local de radio.

La etapa de sintonización está constituida por un autotransformador en resonancia con un condensador variable de 820 pF para permitir la sintonización manual. El bobinado primario del autotransformador se ha hecho con 30 vueltas y está conectado a la antena. El bobinado secundario se ha hecho con 90 vueltas. Ambas bobinas se enrollan sobre el mismo núcleo. El núcleo es una ferrita móvil adaptada a la frecuencia deseada. La conexión común del autotransformador se ha conectado a tierra.

En la siguiente sección se estudiará en más detalle esta parte en conjunto con el doblador de tensión y los cálculos sobre los valores de los componentes.

### 6.2.3 Doblador de tensión

La etapa de sintonización está conectada a un circuito rectificador, el cual está formado por una bomba de carga. Una bomba de carga es un circuito para convertir una tensión en otra basado en condensadores, los cuales se conmutan periódicamente [11]. A diferencia de otros convertidores, en este caso no requieren bobinas.

La base de este circuito es una arquitectura de doblador de tensión. Este circuito es uno basado condensadores y, en algunos casos, diodos, que aumentan la tensión de su entrada; teniendo la tensión de entrada un componente AC. El doblador más simple puede ser un rectificador cuya entrada puede ser una señal AC y la salida una tensión DC con valor medio superior. Uno de los circuitos dobladores más conocido es el de Villard [12], el cual está compuesto por un condensador y un diodo. En este circuito -considerando componentes ideales-, cuando la tensión de entrada es menor que cero el diodo conduce y el condensador se carga con tensión negativa; a continuación, cuando la entrada es positiva, el diodo se queda en circuito abierto y la salida es la suma de la entrada y la tensión acumulada en el condensador; si la entrada es simétrica, la salida será el doble de la entrada. La ventaja de este circuito es su gran simplicidad, el inconveniente es que deja pasar a su salida gran parte del componente AC. Una modificación de este circuito es el de Dickson, el cual cambia la disposición del condensador y la bobina, dejando un circuito con la misma topología que el típico receptor de AM. Este segundo circuito mejora la respuesta DC de la salida. En la Figura 6-4 se muestra en la parte superior una comparativa de las topologías de ambos circuitos y en la parte inferior una simulación de su funcionamiento.

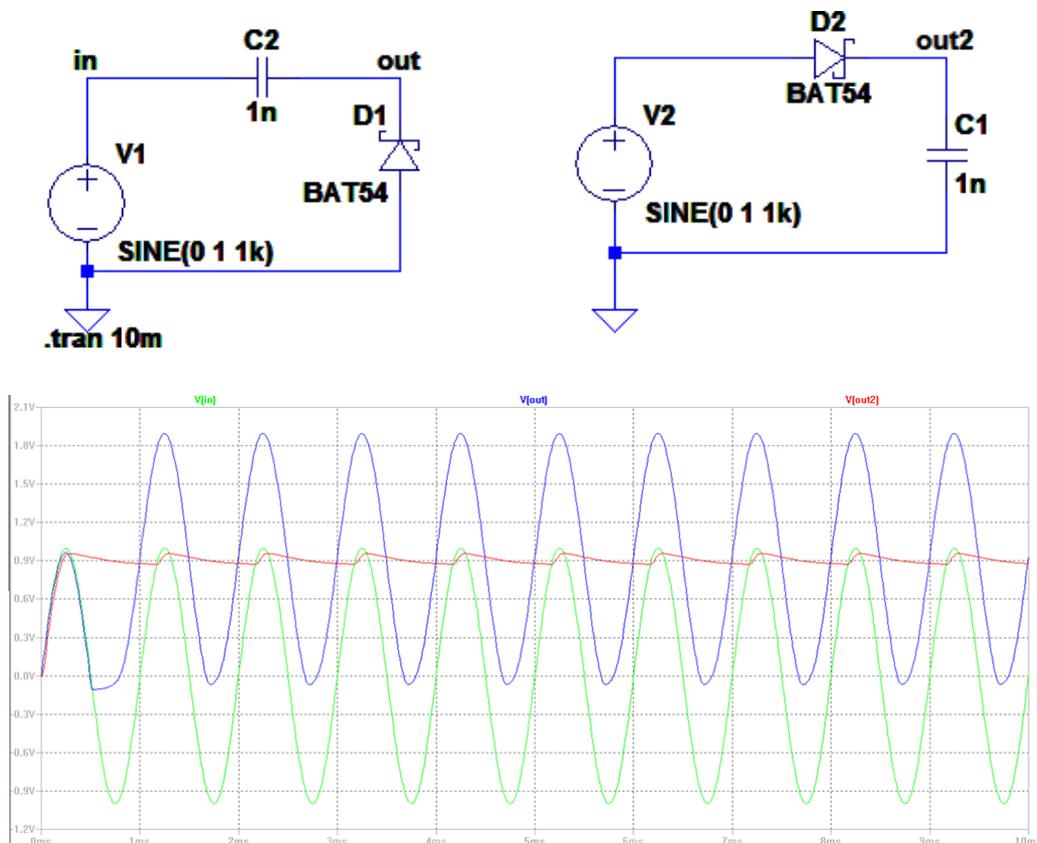


Figura 6-4: Dobladores Villard (izquierda) y Dickson (derecha); y comparativa de sus salidas (gráfica inferior), donde en verde se ve la entrada, en azul la salida del circuito Villard y en rojo la del Dickson.

A estos circuitos se les pueden añadir varias etapas en serie, formadas cada una por la misma topología de la anterior. De esta manera se consigue aumentar todavía más el nivel de tensión de la salida.

Se han llevado a cabo algunos test preliminares con las configuraciones de bombas de carga mencionadas antes. En estas pruebas se ha hecho uso de bobinas para lograr la sintonización con los condensadores del doblador. La inductancia de las bobinas se ha simulado con SPICE para buscar los valores que establecen la frecuencia de resonancia deseada, la cual es la del canal de radio AM seleccionado, que está en 1,584 MHz.

Analíticamente se puede estudiar la salida de un multiplicador de varias etapas de tipo Villard, la cual se puede expresar mediante la siguiente expresión [12]:

$$V_{out} = \left(\frac{n}{2}\right) \cdot (V_{in+} - V_{in-}) - \left(\frac{n^3}{12} + \frac{n^2}{4} - \frac{n}{6} \cdot (3 \cdot D^2 - 3 \cdot D + 1)\right) \cdot \left(\frac{T}{C}\right) \cdot I_{out} \quad (6.1)$$

donde:

$n$  es el número de diodos (etapas).

$V_{in+}$  y  $V_{in-}$  son las tensiones máximas y mínimas de la señal de entrada.

$$D = \frac{T_1}{T}$$

$T$  es el periodo de la señal de entrada.

$T_1$  es el tiempo de la señal de entrada con valor positivo durante un periodo.

$C$  es el valor de los condensadores.

$I_{out}$  es la corriente de salida.

Si la entrada es una onda sinusoidal, el tiempo de conducción de los diodos es muy pequeño y  $n$  es elevado, la impedancia de salida se puede aproximar a:

$$R_{out} \approx \frac{\partial V_{out}}{\partial (-I_{out})} = \frac{n^3}{12 \cdot C \cdot f_s} \quad (6.2)$$

donde  $f_s = \frac{1}{T}$ .

Se ha realizado una medida preliminar indirecta de la  $R_{out}$  de forma práctica para una etapa del circuito; para ello se ha medido la tensión y corriente en la salida en DC. El valor resultante es aproximadamente  $R_{out} \approx 200 \text{ k}\Omega$ .

Se fija el número de etapas en  $n = 4$  también experimentalmente. Más etapas supondrán más tensión a la salida, lo que es deseable porque permitiría cargar a más tensión —y por tanto carga— un condensador; sin embargo más etapas también implican más pérdidas. Añadiendo más etapas se ha visto que a partir de 4 no se obtenía más beneficio.

También se fija  $f = 1,584$  MHz porque es la frecuencia del canal con el que se trabajará. Utilizando la última expresión y estos datos se puede obtener un primer valor teórico mínimo para el valor del condensador:

$$C = \frac{n^3}{12 \cdot R_{out} \cdot f} \approx 16,84 \text{ pF} \quad (6.3)$$

Como se ha visto en la Figura 6-4, la salida presenta cierta componente en AC, especialmente en la configuración de Villard. Para reducir esta componente se elegirá un valor más elevado, en nuestro caso  $C = 820$  pF porque tenderá a reducir las oscilaciones en la tensión.

Con este valor, mediante la expresión  $f = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}}$  se seleccionará un valor para  $L$  de tal manera que la frecuencia quede cercana a la de la emisora. No es necesario el valor exacto, ya que en un paso posterior en la práctica se sintonizará el sistema manualmente. En este caso se ha seleccionado una  $L = 5,6$   $\mu\text{H}$ .

Se han seleccionado diodos de germanio al igual que en los tradicionales demoduladores AM porque tienen una baja caída de tensión (0,3 V) y una baja capacidad parásita (sobre 1 pF).

La Figura 6-5 muestra las diferentes configuraciones que se han evaluado para la bomba de carga. Los esquemas (1) y (2) son los tradicionales, también llamados el multiplicador de tensión Villard o Cockcroft-Walton y el rectificador Dickson [12][13]. Los esquemas (3) y (4) son variaciones de los anteriores que consiguen salidas de tensión más elevadas en un ancho de banda más reducido, manteniendo una respuesta similar a los tradicionales en el resto de la banda [14]. Esto se debe a que, al tener circuitos resonantes, la energía que en condiciones normales estaría distribuida en todo su espectro, ahora se distribuirá en torno a su frecuencia de resonancia.

El proceso de validación de la bomba de carga tiene dos etapas: la primera consiste en obtener la respuesta en frecuencia del circuito. Para ello se introducirá una señal sinusoidal en la entrada con amplitud constante y frecuencia variable. Además se utilizarán simulaciones para ajustar los valores de los componentes para obtener la frecuencia de resonancia más próxima a la deseada.

Una vez que el circuito está sintonizado en la frecuencia de resonancia deseada, la segunda etapa consistirá en la implementación física del circuito, se conectará el circuito rectificador a la antena y se medirán los resultados.

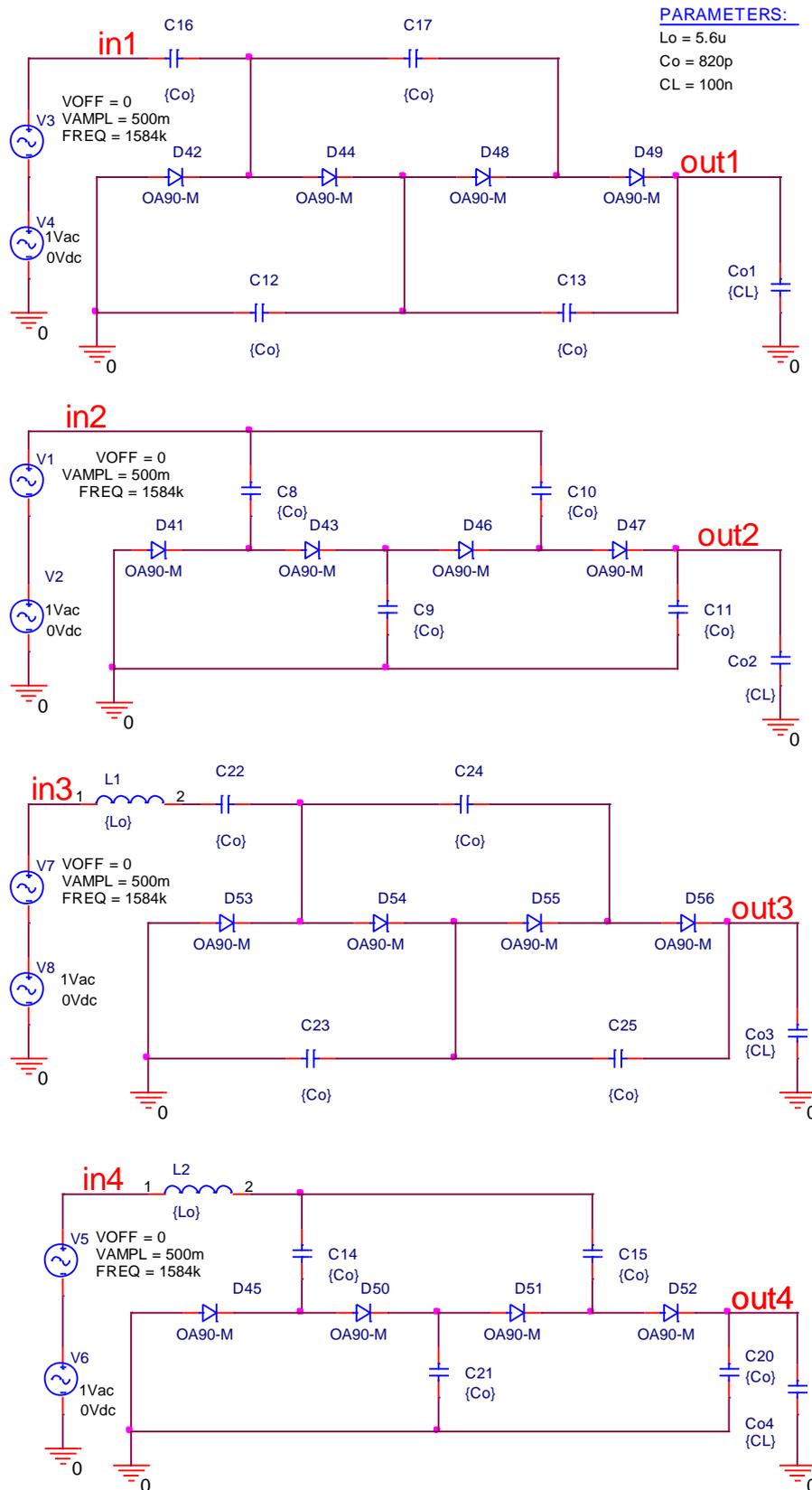


Figura 6-5: Esquemáticos de las diferentes topologías de circuitos. Desde arriba hacia abajo: (1) Villard; (2) Dickson; (3) Villard resonante; (4) Dickson resonante.

### 6.3 Simulación y resultados

Todas las configuraciones se han simulado para comprobar la evolución de la capacidad del condensador (Figura 6-6). Se han probado varios valores de condensadores para almacenar la energía (100  $\mu\text{F}$ , 10  $\mu\text{F}$ , 1  $\mu\text{F}$ , 100 nF, 10 nF, 1 nF) y se ha obtenido la tensión de salida para un tiempo determinado. Después de esto se ha elegido el valor de 100 nF para comparar la respuesta de los distintos circuitos.

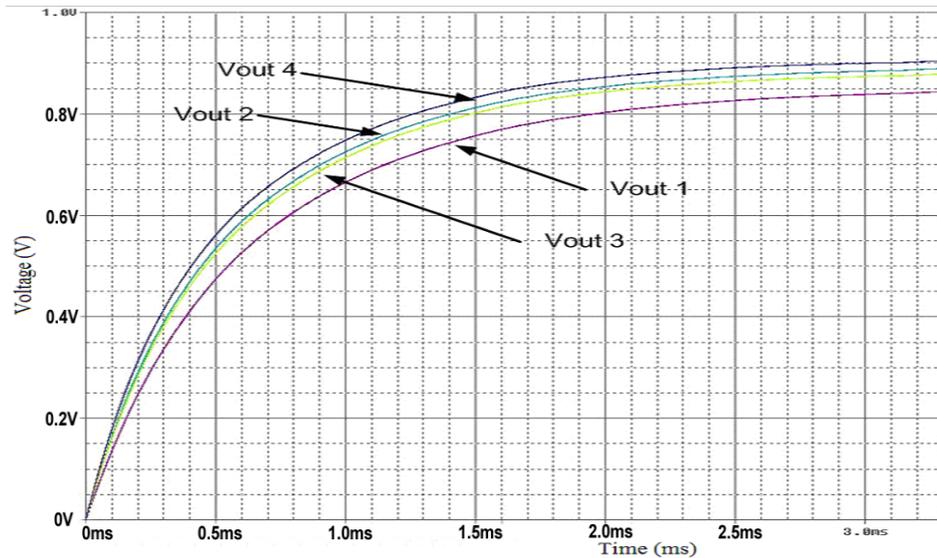


Figura 6-6: Respuesta de las cuatro topologías de circuitos mostrando la tensión en el condensador de salida.

En la Figura 6-6 se puede observar que la configuración resonante Dickson ( $V_{out4}$ ) alcanza la salida de tensión más alta con las mismas condiciones que los demás. Además, las configuraciones resonantes mejoran la eficacia del multiplicador de tensión respecto a los no resonantes.

Por otra parte, se ha llevado a cabo un montaje experimental usando un generador de señal HP33120A y un osciloscopio TDS210, ambos controlados remotamente por un programa LabVIEW. El generador de señal aplica una señal de tensión AC de amplitud constante (200 mVp) y frecuencia variable a los circuitos de la Figura 6-5. El valor RMS del condensador se medirá con el osciloscopio y se enviará al programa de control. Este programa irá variando la frecuencia de la fuente y leyendo la tensión RMS para distintas frecuencias. De esta manera se obtendrá la respuesta en frecuencia de cada configuración.

En la Figura 6-7 se representa la respuesta del circuito resonante Villard, la cual logra un máximo de aproximadamente 6 veces superior a la entrada en la frecuencia de resonancia.

La respuesta del circuito resonante Dickson se muestra en la Figura 6-8 y consigue una tensión máxima aproximadamente 13 veces mayor a la entrada. La respuesta de los circuitos no resonantes es plana en el rango de frecuencias evaluado, como era de esperar.

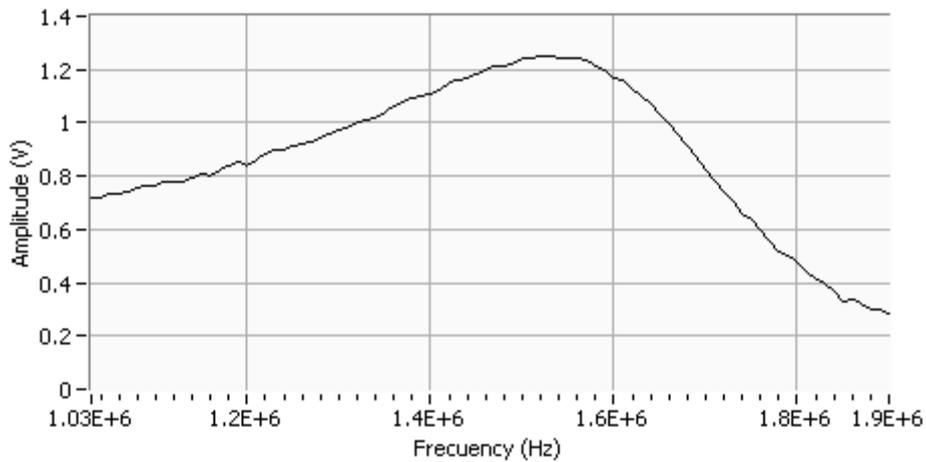


Figura 6-7: Resultado experimental. Tensión de salida vs frecuencia para el circuito resonante Villard.

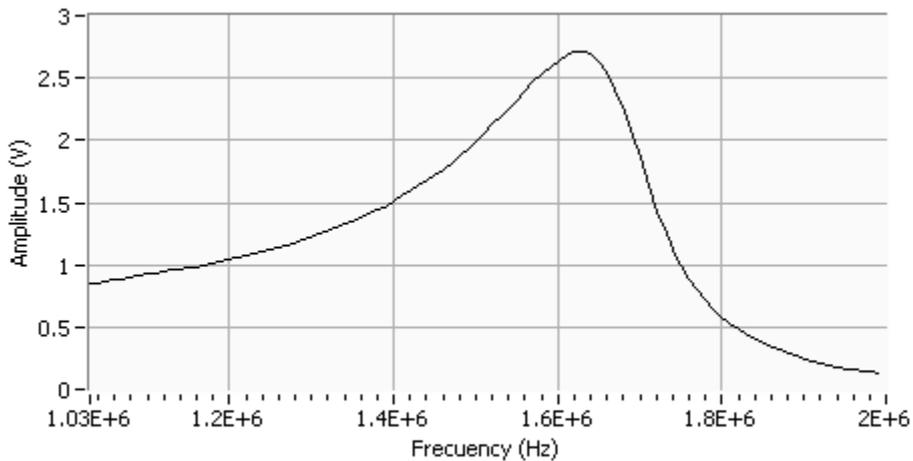


Figura 6-8: Resultado experimental. Tensión de salida vs frecuencia para el circuito resonante Dickson.

La configuración Dickson consigue valores mayores porque el autotransformador L2 está en paralelo con la capacidad C14, por lo que la tensión en el punto de conexión se eleva más. Mientras que en la configuración Villard la bobina L1 está en serie con C22 y no aprovecha tanto la sintonización.

También se ha medido la ganancia diferencial del sistema. Se ha utilizado un analizador de espectros HAMEG HM5014 para obtener la potencia de la señal en diferentes puntos del sistema. En la Figura 6-9 se muestran varias capturas de pantalla de la medida del analizador. La primera imagen muestra la señal recibida desde la estación de AM sin antena (-104,2 dBm). La siguiente imagen es la potencia de la señal con la antena L (-96,2 dBm). La última imagen muestra el efecto de la etapa de sintonización (-69,8 dBm). Como se puede ver, la antena tiene una ganancia de 8 dB y la etapa de sintonización de 26,4 dB.

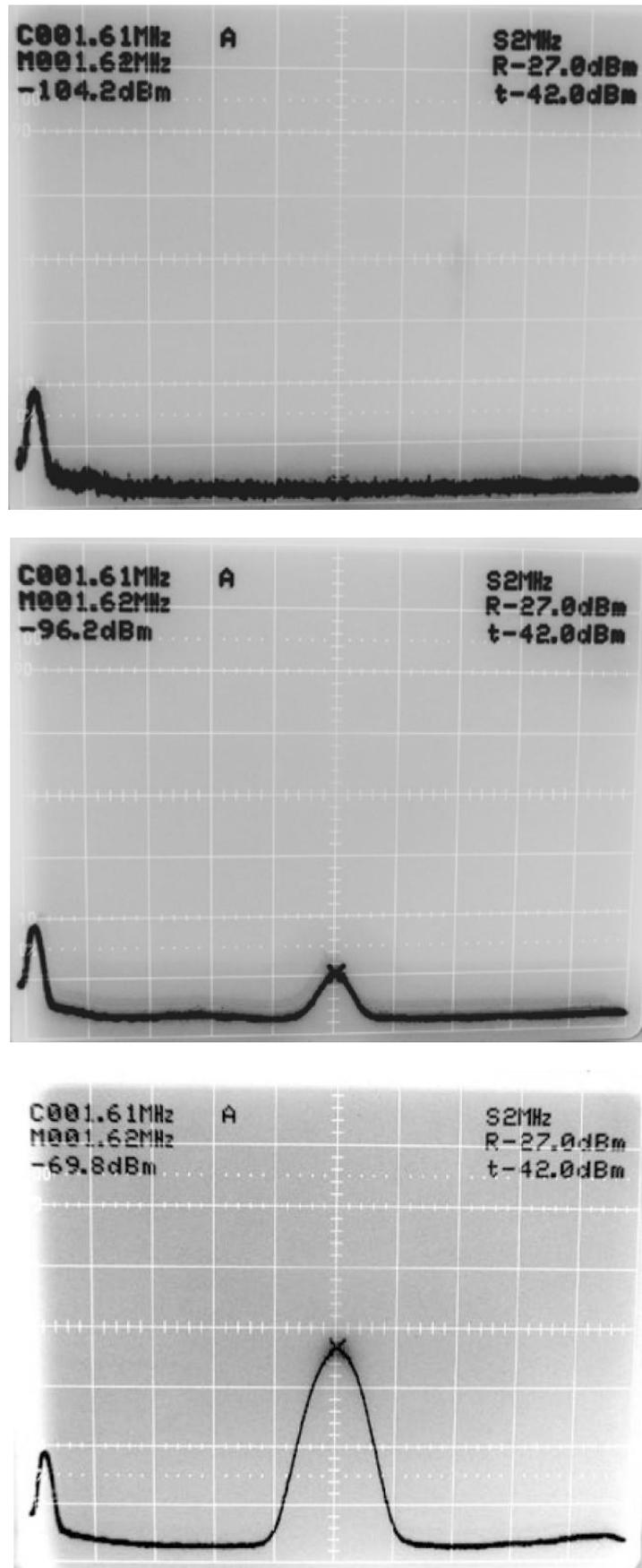


Figura 6-9: Resultados experimentales. Potencia vs frecuencia en diferentes puntos del sistema. Desde arriba hacia abajo: (1) potencia sin antena; (2) potencia con antena de tipo L; (3) potencia con circuito de sintonización.

Finalmente, después de la evaluación mediante el generador de señal, el circuito se ha conectado a la antena en L adaptada a la placa experimental. Se han probado los dos circuitos rectificadores resonantes. Se ha medido la corriente en cortocircuito y la tensión en circuito abierto de ambas configuraciones. Se ha conseguido una tensión de 1,6 V para el circuito resonante Villard y una corriente de 8  $\mu\text{A}$ . Para el Dickson resonante se ha conseguido una tensión de 2,75 V en circuito abierto y 3,5  $\mu\text{A}$ , como se puede ver en la Tabla 6-1.

Tabla 6-1: Resumen de los resultados experimentales en los circuitos resonantes.

	Resonante Villard	Resonante Dickson
Corriente en cortocircuito	8 $\mu\text{A}$	3,5 $\mu\text{A}$
Tensión en circuito abierto	1,6 V	2,75 V

## 6.4 Conclusiones

Se han estudiado, simulado e implementado cuatro topologías de rectificadores para obtener energía de RF del entorno. El estudio revela que sintonizar un circuito rectificador sobre una frecuencia conocida donde hay una señal de AM mejora los resultados. Mediante estos circuitos se puede conseguir hasta tres veces más tensión que con los rectificadores no sintonizados.

Los resultados experimentales y simulados no pueden ser comparados porque el modelo SPICE de los diodos que suministran los fabricantes no es completo. La dependencia de la capacidad del diodo respecto a la tensión y, por tanto con respecto a la tensión de entrada, no está considerada en el modelo. Esto repercute en la frecuencia de resonancia obtenida modificándola y es la causa que ha hecho que los valores de algunos componentes (como las bobinas) tengan que ser recalculados. Por tanto se puede concluir que la selección de los componentes es de vital importancia.

Se ha conseguido una corriente máxima de 8  $\mu\text{A}$  utilizando un rectificador Dickson resonante. Esta medida demuestra que este método de recolección de energía podría ser viable si se aplica a nodos de redes de sensores con ciclos de trabajo bajos, muy bajo consumo y bajo voltaje.

A pesar de ser una corriente relativamente baja, hay que tener en cuenta que su flujo es ininterrumpido, por lo que puede servir de apoyo a otras fuentes de energía más puntuales, como las que provienen de energías renovables.

La solución propuesta no requiere un generador de RF porque aprovecha la energía que se gastaría por absorción del medio de estaciones de radio y televisión comerciales, lo cual supone una gran ventaja respecto a los métodos tradicionales. Además el rango de cobertura es más amplio que en otros sistemas, llegando a ser de varios kilómetros.

## 6.5 Bibliografía

- [1] Mateu, L., & Moll, F. (2005, June). Review of energy harvesting techniques and applications for microelectronics (Keynote Address). In *Microtechnologies for the New Millennium 2005* (pp. 359-373). International Society for Optics and Photonics.
- [2] Popović, Z. B. (2006). Wireless powering for low-power distributed sensors. *Serbian Journal of Electrical Engineering*, 3(2), 149-162.
- [3] McSpadden, J. O., & Chang, K. (1994, May). A dual polarized circular patch rectifying antenna at 2.45 GHz for microwave power conversion and detection. In *Microwave Symposium Digest, 1994., IEEE MTT-S International* (pp. 1749-1752). IEEE.
- [4] Jamali, B., Ranasinghe, D. C., & Cole, P. H. (2005, December). Analysis of UHF RFID CMOS rectifier structures and input impedance characteristics. In *Microelectronics, MEMS, and Nanotechnology* (pp. 60350D-60350D). International Society for Optics and Photonics.
- [5] Yan, H., Montero, J. M., Akhnouk, A., De Vreede, L. C. N., & Burghartz, J. (2005, November). An integration scheme for RF power harvesting. In *Proc. STW Annual Workshop on Semiconductor Advances for Future Electronics and Sensors* (pp. 64-66).
- [6] Jiang, B., Smith, J. R., Philipose, M., Roy, S., Sundara-Rajan, K., & Mamishev, A. V. (2005, May). Energy scavenging for inductively coupled passive RFID systems. In *Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE* (Vol. 2, pp. 984-989). IEEE.
- [7] Mikami, S., Matsuno, T., Miyama, M., Kawaguchi, H., Yoshimoto, M., & Ono, H. (2006). An Energy-Harvesting Wireless-Interface SoC for Short-Range Data Communication. *IEEJ Transactions on Electronics, Information and Systems*, 126, 565-570.
- [8] Latrach, M., Zbitou, J., & Toutain, S. (2002). Low cost microwave rectifier for low and high powers. In *Proc. URSI*.
- [9] Zbitou, J., Latrach, M., & Toutain, S. (2006). Hybrid rectenna and monolithic integrated zero-bias microwave rectifier. *Microwave Theory and Techniques, IEEE Transactions on*, 54(1), 147-152.
- [10] Mi, M., Mickle, M. H., Capelli, C., & Swift, H. (2005). RF energy harvesting with multiple antennas in the same space. *Antennas and Propagation Magazine, IEEE*, 47(5), 100-106.
- [11] Penella-López, M. T., Gasulla-Forner, M (2011). Powering Autonomous Sensors. An integral approach with focus on solar and RF energy harvesting. *Springer*. ISBN 978-94-007-1573-8.
- [12] Lamantia, A., Maranesi, P. G., & Radrizzani, L. (1994). Small-signal model of the Cockcroft-Walton voltage multiplier. *Power Electronics, IEEE Transactions on*, 9(1), 18-25.
- [13] Lamantia, A., Maranesi, P., & Radrizzani, L. (1990, June). The dynamics of the Cockcroft-Walton voltage multiplier. In *Power Electronics Specialists Conference, 1990. PESC'90 Record., 21st Annual IEEE* (pp. 485-490). IEEE.
- [14] Jullian, M. (2005, August). Cockcroft Walton Optimum Design Guide. In Blazelabs.

# 7. NODO ALIMENTADO POR ENERGÍA SOLAR PARA SISTEMAS DE IRRIGACIÓN

---

## 7.1 Introducción

Los sistemas de irrigación requieren utilizar solenoides de tipo *latch* para controlar automáticamente las zonas a irrigar. Actualmente (año 2008) el control de estos sistemas se realiza mediante redes cableadas con dos o tres cables para realizar la comunicación, sobre la cual se enviará una serie de órdenes de tipo activa/desactiva el solenoide, como por ejemplo Piccolo RTU de Motorola [1]. Algunos sistemas más recientes están basados en enlaces de radio a 2,4 GHz, como [2-5], pero estos sistemas no implementan el control de una válvula de solenoide. Muchos de estos sistemas se limitan a monitorizar las condiciones del entorno. Otros sistemas más avanzados como Piccolo-XR [1] sí realizan un control, pero para ello utilizan baterías, las cuales deben ser reemplazadas cada cierto tiempo debido al desgaste para asegurarse un correcto funcionamiento.

Las redes de sensores inalámbricas tienen algunas peculiaridades como que, en general, no requieren grandes tasas de transferencias de datos y que suelen funcionar de forma periódica con transmisiones puntuales. Esto permite optimizar los protocolos de comunicación para reducir el consumo, lo que a su vez permite alimentar los nodos utilizando pequeñas baterías y/o supercondensadores. El motivo es que las comunicaciones son, en muchos casos, el mecanismo que más consumo presenta.

En el diseño propuesto se ha realizado una optimización basada en hardware de los protocolos de comunicación para reducir el consumo de energía de las comunicaciones. Así la alimentación desde pequeños paneles de energía solar se convierte en viable, además de poder realizar una actuación suficientemente rápida sobre los solenoides. Para realizar esta optimización se necesita identificar los mecanismos de gasto de energía, detallados en [6], donde se puede comprobar la importancia de la temporización en estos tipos de nodos y aplicaciones.

Los protocolos de nivel MAC tradicionales mantienen los *transceivers* en un estado de recepción durante todo el tiempo que no utilizan el canal porque la eficiencia energética no es su prioridad [7], pero esto no es práctico para redes de sensores y, en general, cualquier dispositivo que requiera bajo consumo. De esta forma, los nodos de las redes de sensores deben apagar los *transceivers* —y en general todos los dispositivos— cuando no sean necesarios [8]. Esta razón y otras como las limitaciones en el hardware o la necesidad de reconfiguración de la red hacen que los esquemas tradicionales de sincronización no sean

válidos en redes de bajo consumo. Por tanto se necesitan nuevas aproximaciones específicamente diseñadas para esta tarea, algunas de las propuestas son [7-8, 10-12].

El caso más simple son dos nodos sensores comunicándose entre ellos. Para que la comunicación se lleve a cabo, los *transceivers* de ambos deben estar activos (sincronizados) en el mismo instante de tiempo, uno en modo transmisión y el otro en modo recepción. Este problema debe resolverse en el protocolo de nivel MAC para determinar cuándo habilitar el *transceiver* implementando un mecanismo de sincronización, además de evitando colisiones para que no haya dos nodos transmitiendo a la vez que puedan generarse interferencias mutuamente.

Como se ha comentado anteriormente, el principal —o uno de los más importantes— mecanismo para ahorrar energía es conseguir deshabilitar el *transceiver* de comunicación cuando no es necesario. Para cuantificar el ahorro que puede suponer se pueden estudiar las características de uno en particular, el CC2420 el cual consume 19,7 mA en modo transmisión, 17,4 mA en recepción y 20  $\mu$ A en modo *power-down* [13].

Por supuesto, el microcontrolador también debe apagarse o ir a un modo de bajo consumo cuando no sea necesario. Desde este modo, el microcontrolador debe despertarse antes para habilitar el *transceiver* y el resto de elementos. El consumo de los microcontroladores en estado activo varía mucho de unos tipos a otros y también en función de sus tensiones de alimentación y frecuencia de reloj, pero en los modos más profundos de dormido suelen estar en el rango de los microamperios.

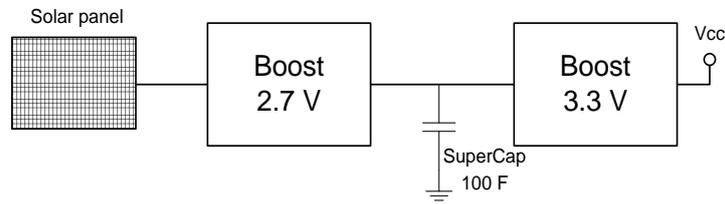
El activado y desactivado de cada bloque únicamente cuando son necesarios provoca los típicos esquemas consistentes en ciclos *listen/sleep* periódicos, típicos de los nodos de las redes de sensores. En modo activo, los nodos realizan tareas como la transmisión o el sensado y, en modo *sleep*, todo lo prescindible se apaga. Por simplicidad se puede considerar que el procesador y los sensores tienen el mismo periodo de activación que el *transceiver*. De este modo, cuando menor sea el ciclo de trabajo (menor tiempo activo), mayor será el ahorro de energía.

## 7.2 Desarrollo hardware wEcoValve

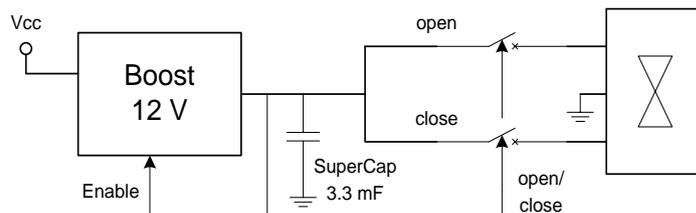
El diagrama de bloques del mote wEcoValve se muestra en la Figura 7-1, donde cada sección es:

- a) Circuito de alimentación. Está formado por un panel solar, un supercondensador y dos convertidores DC/DC de alta eficiencia. La fuente de alimentación es una simplificación de la presentada en el capítulo 5 eliminando la batería a costa de aumentar el supercondensador. Con este cambio se simplifica el circuito, se reduce el tamaño y se ahorra en costes, aunque no se puede garantizar la perpetuidad del funcionamiento si hubiera varios días sin energía solar; pero en esta aplicación no eso no es crítico, además, en el caso de haber varios días sin suficiente radiación solar, probablemente se deba a que ha llovido, por lo que no sería necesario el riego.

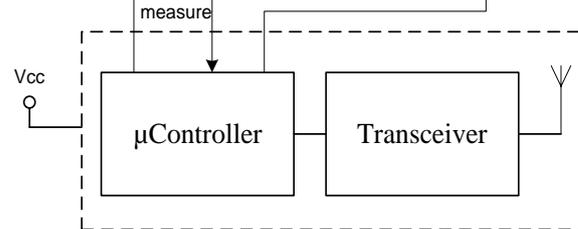
- b) Circuito de control del solenoide. Formado por un convertidor DC/DC, un supercondensador secundario y un circuito de control para conmutar el solenoide con un pulso de corriente suficiente a 12 V.
- c) Diagrama del microcontrolador y el *transceiver* de comunicación.



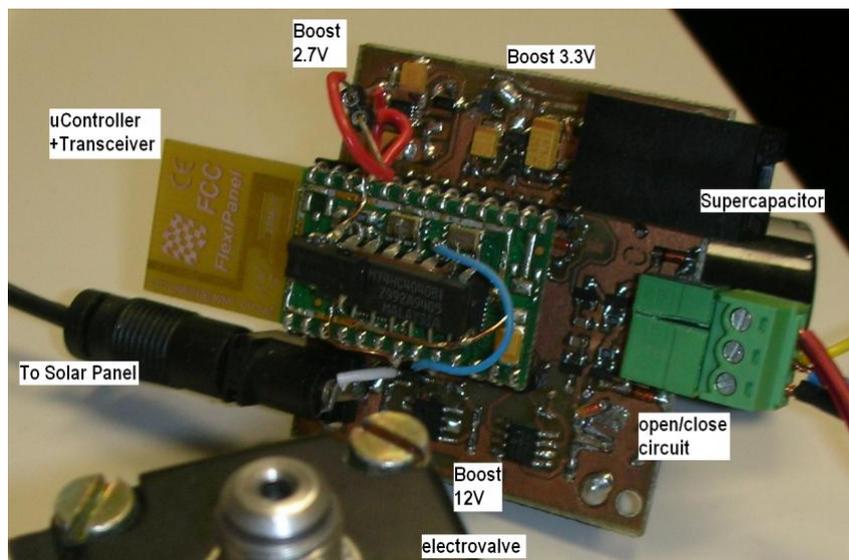
(a)



(b)



(c)



(d)

Figura 7-1: Diagrama de bloques del nodo terminal.

El nodo actuador es similar al diseñado en [15], pero adaptado a las necesidades de actuación sobre una electroválvula y basado en el uso de un panel solar que recarga un único supercondensador de 100 F, el cual reemplaza por completo a las baterías. El esquemático del nodo se muestra en la sección 10.2.3 de los anexos.

Se ha utilizado un oscilador SG-3030JC [16] y un contador 74HC40103 para generar una señal de reloj a partir del oscilador, pero dividida, para permitir un modo de *sleep* profundo en el microcontrolador como se explicará más adelante.

Los pulsos de corriente típicamente son de 700 mA y 12 V de pico durante 20 ms [15], tanto para abrir como para cerrar la electroválvula. Los pulsos generados por este sistema son conseguidos a través de un convertidor Boost de 12 V que carga un segundo supercondensador de 3,3 mF. La energía es acumulada en el supercondensador hasta que llega a una tensión de 10,5 V, en el momento que el microcontrolador lee esa tensión, se activa el circuito de disparo de la electroválvula durante 10 ms. Se ha comprobado experimentalmente que esta configuración es válida para la electroválvula utilizada.

Además se han incluido otros circuitos para realizar una medida de corriente proporcionada por el panel solar y para medir la tensión en el supercondensador principal. Esta información es útil para las tareas de validación y modelización del consumo.

### 7.3 Desarrollo de los protocolos de sincronización

En [7] se clasifican los diferentes tipos de protocolos, poniendo especial interés en:

- De división de tiempo o *slotted*: TDMA, 802.15.4, SMAC, DSMAC, TMAC, FRAME.
- De muestreo: ALOHA, BMAC, WiseMAC, CSMA (y variantes como CSMA/CA para redes de sensores).
- Híbridos: ZMAC, SCP-MAC.
- Otros: STEM (que utiliza dos *transceivers*, uno para enviar datos y otro permanente activo para despertar el sistema).

Los protocolos de división en el tiempo mantienen los nodos permanentemente sincronizados, por lo tanto son particularmente útiles en aplicaciones donde hay un intercambio periódico de información. En ellos la activación y desactivación es en general bastante determinística, lo que permite ajustar de forma más precisa los periodos en modo activo.

Por otro lado, los protocolos de muestreo son más convenientes, desde el punto de vista de ahorro energético, en aplicaciones con transmisiones esporádicas porque no existe la penalización de mantener el sincronismo, simplemente se busca cuando es necesario.

#### 7.3.1 Protocolos de división de tiempo

Los protocolos de división en el tiempo requieren que todos los nodos trabajen con una señal de reloj común, la cual servirá para mantener la sincronización. Existen diversas propuestas, pero al final, en el nivel más bajo, el reloj de todos los nodos depende del uso del oscilador principal del sistema para mantener la sincronización. Si todos los osciladores fueran ideales, los nodos estarían siempre perfectamente sincronizados después de una calibración inicial; sin

embargo ocurren variaciones en la frecuencia de los relojes de un nodo respecto a los demás, incluso aunque el mismo modelo de oscilador, por ejemplo debido a cambios en la temperatura, envejecimiento o estabilidad [16].

Por ejemplo, el reloj usado en la placa PICDEM.Z, el HC-49US de ECS International tiene una estabilidad de  $\pm 15$  ppm a  $\pm 100$  ppm a 25 °C. Esto significa que, en el peor caso, en un minuto dos nodos pueden haberse desincronizado 12 ms. Para evitar este desajuste, los protocolos envían periódicamente unas tramas especiales de señalización (*beacons*) a todos los nodos para mantener el sincronismo; o bien utilizan el preámbulo de otras tramas para enviar información sobre la sincronización. En ambos casos, se establecen unos márgenes de deriva para incluir todas las desviaciones posibles.

Los protocolos MAC se enfocan en activar y desactivar el *transceiver*, y esto requiere que el microprocesador mantenga la sincronización. Esto implica un problema si se quiere poner el microprocesador en el modo de consumo más bajo posible: si tanto el microprocesador como todos sus temporizadores se apagan, el sincronismo se perderá.

En el caso de la placa mencionada antes, se utiliza un PIC18F4620 con una frecuencia de reloj de 4 MHz y 3 V. Esto implica un consumo de 1,3 mA en modo activo, 430  $\mu$ A en modo PRI\_IDLE (CPU y periféricos apagados) y 0,1  $\mu$ A en modo *sleep* profundo [17]. El modo PRI\_IDLE permite mantener un contador activo en el microcontrolador para medir el paso del tiempo y mantener así el sincronismo. Para un ciclo de trabajo del 1% y 21 mA de consumo en modo escucha (microcontrolador + *transceiver* activos), y un consumo de 450  $\mu$ A en modo *sleep* (microcontrolador en PRI\_IDLE y *transceiver* en *power down*), el consumo medio del sistema sería de 655,5  $\mu$ A.

Si el microcontrolador pudiera utilizar el modo *sleep*, se podría conseguir un consumo medio de 210,1  $\mu$ A. Por tanto, el poder utilizar el modo *sleep* mejora en tres veces el consumo medio, lo que justifica su importancia.

### 7.3.2 Propuesta

Los protocolos de división del tiempo con bajos ciclos de trabajo tienden a igualar el consumo medio al consumo en estado *sleep*, tanto más cuanto menor sea el ciclo de trabajo. Por lo tanto, reducir el consumo a modo *sleep* tiene un factor decisivo en el consumo medio. Y ya se ha visto anteriormente que un modo muy efectivo de reducir el consumo en modo dormido es llevando el microcontrolador a un modo de *sleep* profundo y estar en este estado todo el tiempo que sea posible.

Para despertar el microcontrolador del modo *sleep* profundo hay varias opciones. La primera aproximación es usar los osciladores internos que suelen incorporar los microcontroladores para despertarse. Habitualmente los circuitos de *watchdog* incorporan un oscilador independiente que siempre está funcionando en cualquier modo. En este caso, el microcontrolador PIC15F4620 incorpora un oscilador interno de 32 kHz [17] conectado al temporizador del *watchdog*. Pero hay dos inconvenientes: 32 kHz es una frecuencia demasiado alta para aplicaciones de bajo ciclo de trabajo, esto produciría activaciones innecesarias del microcontrolador, la solución sería utilizar algún mecanismo de pre o postcaler; el segundo —y principal— inconveniente es que el oscilador es de tipo RC, los cuales suelen presentar peor

estabilidad que los osciladores basados en cristal de cuarzo. En el caso del microcontrolador bajo estudio, este oscilador puede variar de 26,562 kHz a 35,938 kHz, lo que hace que en un periodo de 1 segundo pueda haber una desincronización de 0,3 s en el peor caso. Esta deriva es demasiado grande para los propósitos de la aplicación, por lo que habrá que descartar el uso de osciladores RC.

La segunda opción es utilizar un pequeño hardware externo como el mostrado en la Figura 7-2. Este hardware consiste en usar un oscilador de cristal de cuarzo muy preciso y baja frecuencia de trabajo. La señal de este cristal pasa a través de un registro, el cual funciona como un divisor de frecuencia. La salida de este registro se conecta a una entrada digital del microcontrolador que señala una interrupción externa que despierta al microcontrolador. Así, además de precisión, no es necesario tener activo ningún bloque interno del microcontrolador y se podrá llevar a un modo de *sleep* profundo, añadiendo únicamente el consumo del nuevo cristal y el registro.

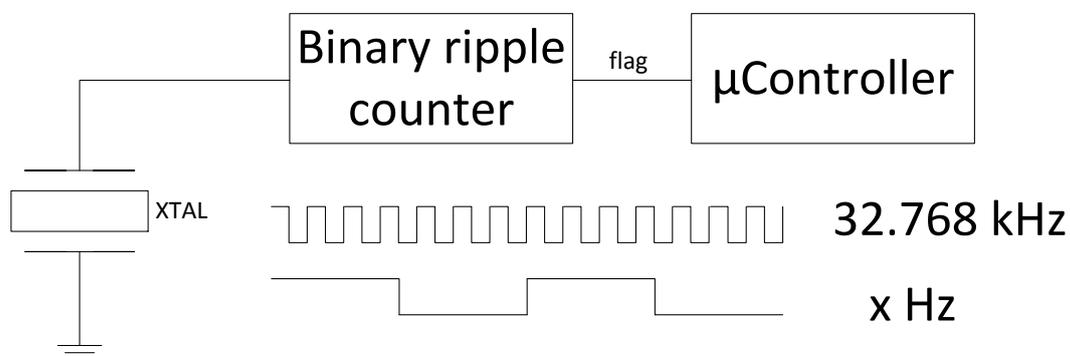


Figura 7-2: Cristal y registro como divisor de frecuencia para generar señal que despierta al microcontrolador.

Al utilizar un oscilador de cristal externo de 32,768 kHz, el consumo y la precisión temporal aumentan enormemente. El oscilador utilizado es un 90SMX de IQD [18], el cual tiene una tolerancia máxima de  $\pm 100$  ppm (típica de  $\pm 20$  ppm). El peor caso es cuando dos nodos están desincronizados el mismo tiempo que en el ejemplo precedente, donde había un reloj de 4 MHz con una tolerancia de 100 ppm, que en este caso es de 12 ms sobre 1 minuto.

La precisión es suficientemente buena para aplicaciones típicas de redes de sensores de bajo consumo, como el control de regadíos. No es necesaria más precisión porque ya entrarían en juego otros factores como el tiempo de *wake-up* que tarda el microcontrolador en pasar de modo *sleep* a modo activo.

Respecto al consumo, el cristal tiene un valor máximo de  $1 \mu\text{W}$ . Por otra parte, el microcontrolador en su modo de *sleep* profundo tiene  $0,3 \mu\text{W}$  ( $0,1 \mu\text{A}$  a 3V).

Añadiendo un registro de división de frecuencia como el 74HC40103 al oscilador, se tiene un consumo de corriente máximo de  $8 \mu\text{A}$  a 6V según sus especificaciones.

Por tanto, el consumo total del nuevo sistema en modo *sleep* es de  $1 \mu\text{W}$  (oscilador) +  $0,3 \mu\text{W}$  (microcontrolador en *sleep*) +  $12 \mu\text{W}$  (divisor de frecuencia), un total de  $13,3 \mu\text{W}$ . En este proyecto se utilizará la opción de despertar el microcontrolador por un temporizador externo por las ventajas que se acaban de exponer.

### 7.3.3 Desarrollo del firmware

La solución propuesta en la sección anterior ha sido implementada junto con una versión del protocolo SMAC [20]. Este protocolo se introdujo en el capítulo 2 y se representa su diagrama temporal en la Figura 7-3. SMAC permite que dos o más nodos se puedan sincronizar y comunicar durante sus periodos en activo. Es un protocolo de división del tiempo en el que cada periodo se divide en cuatro intervalos llamados SYNC (*Synchronization*), CTS (*Clear To Send*), RTS (*Request To Send*) y data. La sincronización se consigue con una transmisión periódica por *broadcast* de tramas SYNC desde un nodo a los demás. También se dispone de un esquema de tramas CTS y RTS para evitar colisiones. Las tramas RTS son una solicitud de permiso para enviar datos y las tramas CTS son la confirmación de la solicitud. Si hay que enviar datos, éstos se envían en los instantes posteriores a RTS. Si los nodos no tienen datos que mandar, los nodos entran en modo *sleep*.

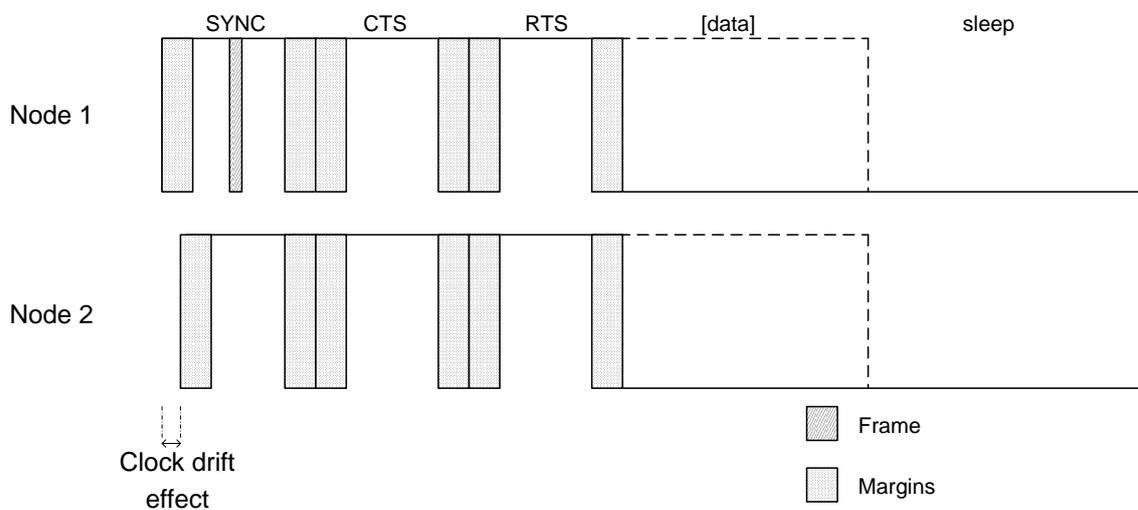


Figura 7-3: Esquema *listen/sleep* con dos nodos utilizando SMAC.

En este esquema, algunos nodos son llamados «nodos de control» y transmiten los comandos para abrir y cerrar remotamente las electroválvulas. El resto de nodos son llamados «nodos terminales», los cuales reciben los comandos del «nodo de control» y actúan sobre las electroválvulas. El hardware de estos nodos es el descrito en la sección «7.2 Desarrollo hardware wEcoValve». Además, los «nodos terminales» monitorizan el estado de las electroválvulas, así como también podrían hacerlo de las condiciones meteorológicas. Para ayudar a la depuración y validación del sistema, los «nodos terminales» también realizan un seguimiento de la corriente aportada por el panel solar y de la carga del supercondensador principal; esta información se envía al control y ayuda a conocer el estado de cada nodo en todo momento. Además, por ejemplo puede utilizarse esta información para saber si hay energía suficiente para conmutar el estado de la electroválvula o el disparo fallará.

Toda la información que llega al «nodo de control» pasa a un PC, actuando éste como coordinador de la red y como *gateway*, como se representa en la Figura 7-4. Desde una aplicación en el PC, el usuario puede ver el estado de la red y enviar comandos a cada nodo para actuar sobre su electroválvula.

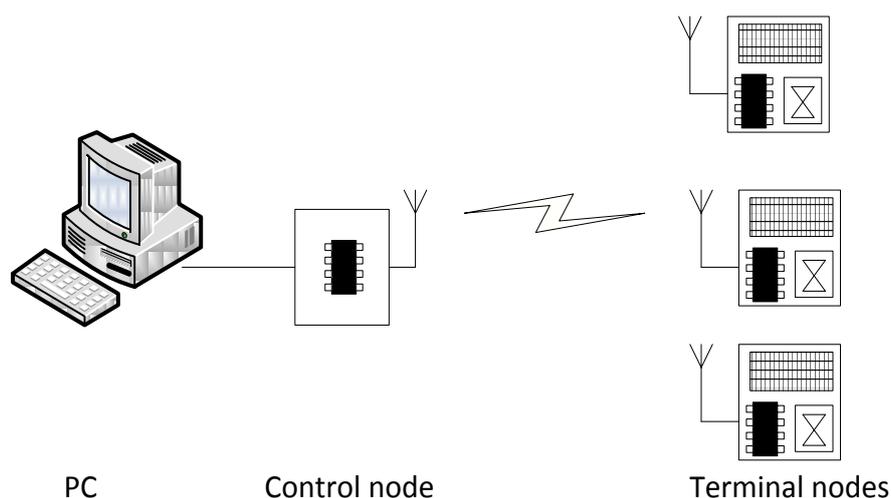


Figura 7-4: Arquitectura de la red.

## 7.4 Resultados y discusión

La red de los nodos se implementa mediante el protocolo SMAC sobre un *transceiver* compatible con 802.15.4. En la implementación se ha seleccionado un periodo de 4 segundos para resincronizar los nodos como se aprecia en la gráfica de consumo de corriente de la Figura 7-5. Cada 4 segundos un nodo difunde por *broadcast* una trama de señalización a todos sus vecinos. Existe un retraso aleatorio entre la transmisión de una trama SYNC de sincronización para reducir la probabilidad de colisiones, el primer nodo que transmita el SYNC hará que los demás se sincronicen con él.

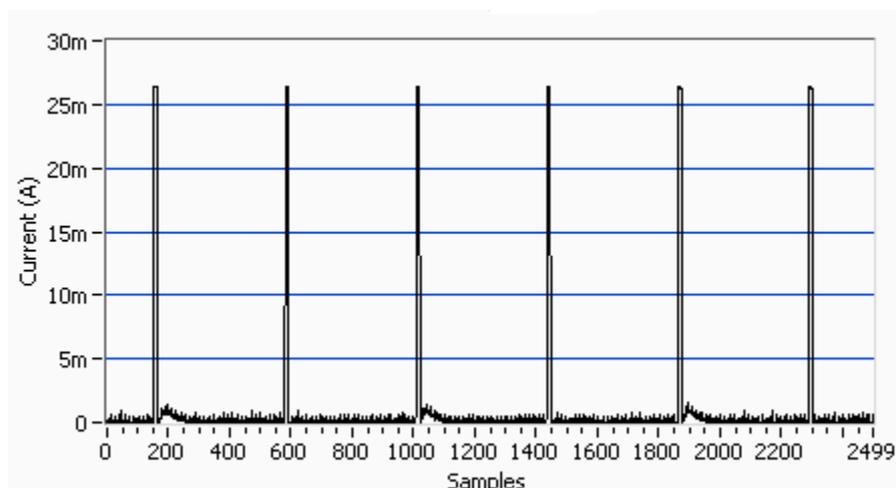
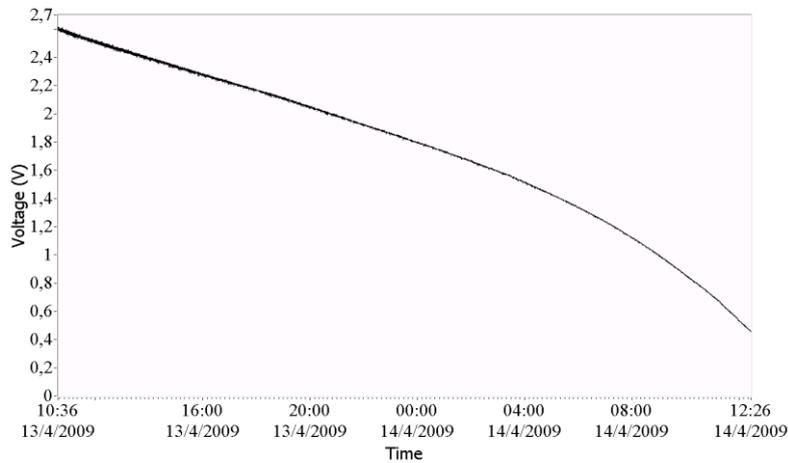


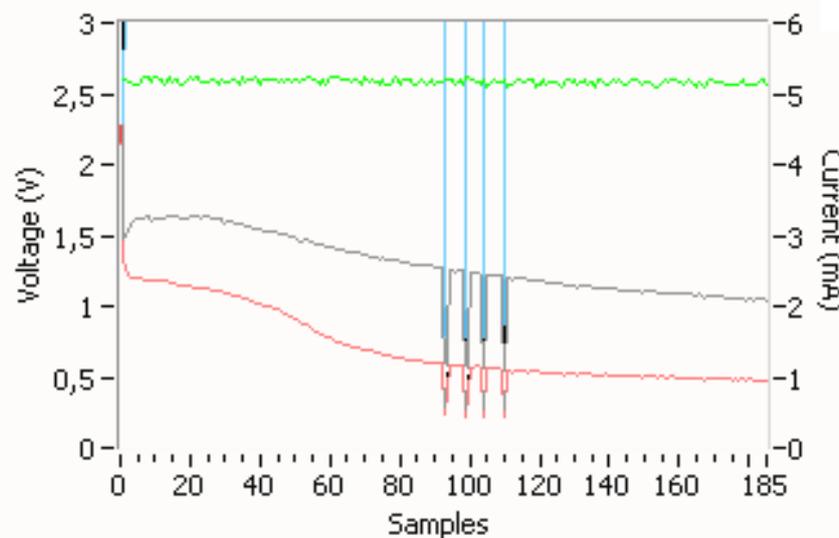
Figura 7-5: Consumo de corriente de un nodo terminal. Los picos de 27 mA son producidos cada 4 s debido a la resincronización. El consumo medio es de 836,76  $\mu$ A.

El tiempo de resincronización puede aumentarse hasta los límites de la precisión del reloj para mejorar el consumo, o reducirse para mejorar el tiempo de latencia que hay entre el envío de la orden y su recepción. Con un tiempo de resincronización de 4 segundos se ha comprobado experimentalmente que se tiene una autonomía de 26 horas sin aporte de energía del panel solar (y sin disparos de la electroválvula), una autonomía de más de un día es suficiente pues en la práctica habrá aporte energético por parte del panel solar. Si se aumentara el tiempo de

resincronización a 8 segundos, el tiempo de vida sin recarga solar es de 50 horas. Se puede ver la descarga del supercondensador principal en la Figura 7-6. El efecto de las conmutaciones de la electroválvula en la alimentación se pueden ver en la Figura 7-7.



**Figura 7-6:** Curva de descarga del supercondensador principal. La autonomía es de aproximadamente 26 horas sin aporte del panel solar con un tiempo de sincronización de 4 segundos. Con un tiempo de 8 segundos la autonomía se extiende a 50 horas.



**Figura 7-7:** Descarga del supercondensador principal, en verde, durante una secuencia abrir-cerrar-abrir-cerrar la electroválvula de un nodo. Señal de disparo de la electroválvula en azul y tensión del panel solar en los extremos de una resistencia shunt (gris y rojo). La tensión en el supercondensador no se ve afectada por los disparos.

Por otro lado, el hardware mostrado en la Figura 7-1 (b) para abrir y cerrar la electroválvula se ha optimizado para cargar y descargar el supercondensador secundario de 3,3 mF. La carga sólo ocurre cuando se hace una petición de conmutación para usar la mínima cantidad de energía posible, si estuviera continuamente cargado habría fugas de energía por las resistencias parásitas del mismo. En la Figura 7-8 se puede ver la carga y descarga de este condensador. El mecanismo, como se ha comentado antes, consiste en iniciar una carga desde el condensador principal de 100 F a este condensador secundario de 3,3 mF a través de un convertidor DC/DC de 12 V; mientras se carga, el microcontrolador monitoriza la tensión en el mismo; cuando la tensión ha llegado a 10,5 V el microcontrolador abrirá el interruptor

correspondiente a la apertura o cierre de la electroválvula, por lo que el supercondensador se descargará rápidamente a través de la electroválvula, además también se apaga el convertidor DC/DC; el tiempo que este interruptor permanece activado es de 10 ms.

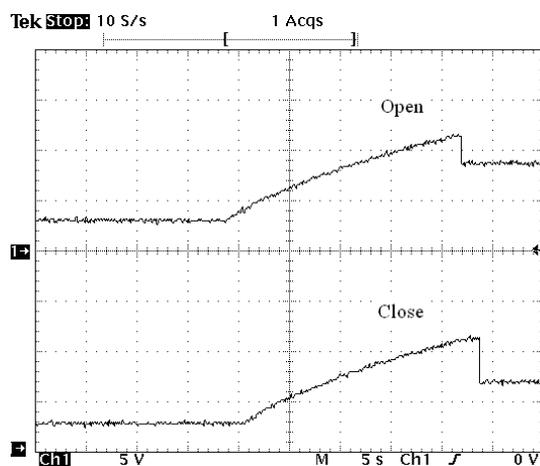


Figura 7-8: Abertura de la electroválvula (superior): un pulso de corta duración (microsegundos) inicia la carga del supercondensador de 3,3 mF hasta alcanzar el umbral de disparo, cuando se alcanza, se dispara interruptor de apertura durante 15 ms. Cierre de la electroválvula (inferior): se carga el supercondensador hasta el umbral y se provoca un pulso de disparo de 20 ms.

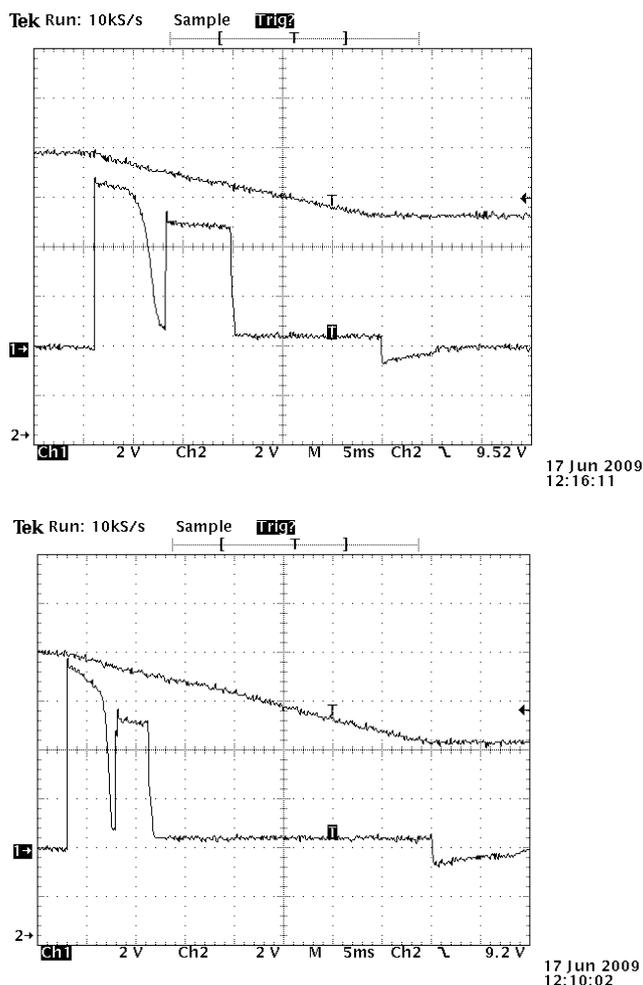


Figura 7-9: Canal 1: corriente para conmutar la electroválvula. Canal 2: caída de voltaje en el supercondensador de 3,3 mF. Gráfica superior: apertura, gráfica inferior: cierre.

Como se puede ver en las gráfica de la Figura 7-8, queda una energía residual en el supercondensador de 3,3 mF después del disparo. Esta tensión puede permitir una secuencia rápida de conmutaciones consecutivas puesto que la recarga desde esta tensión será más rápida. En este caso, una carga a partir de la tensión residual dura aproximadamente 3 segundos, mientras que una carga completa necesita 23,1 segundos para alcanzar la tensión de disparo.

## 7.5 Conclusiones

Se ha implementado un nodo para redes de sensores con un actuador, el cual funciona como una electroválvula autónoma activada por RF que puede ser usada para regadío de jardines, plantas de interior, invernaderos o cultivos extensivos. También se puede utilizar para cualquier sistema que requiera la activación de válvulas con solenoide. Todo ello sin necesidad de cableado, reduciendo por tanto la necesidad de infraestructuras costosas.

Esta propuesta se ha implementado dentro de una aplicación de redes de sensores que se utiliza para controlar regadíos en jardines por medio de electroválvulas, las cuales pueden abrir o cerrar el flujo de agua hacia las plantas. Cada electroválvula está conectada a un nodo que se podrá controlar de forma inalámbrica e independiente.

Se ha desarrollado e implementado una nueva estrategia para reducir el consumo que permite mantener varios nodos sincronizados mediante unos mínimos componentes externos. El nodo tiene una sincronización tan precisa como con el reloj principal del microprocesador pero con menos periféricos activados, lo que implica un consumo menor. Este mecanismo se puede usar en otros protocolos de sincronización para reducir el tráfico de mensajes o mejorar la precisión.

Una ampliación sería utilizar una entrada digital para el contador de agua y enviar esta información y otras provenientes de sensores a la central de forma muy sencilla.

## 7.6 Bibliografía

- [1] Irrinet for a greener world. Irrinet system description. Product brief, Motorola: Schaumburg, IL, USA, 2009; Available online [www.mottech.com](http://www.mottech.com).
- [2] Lopez, J.A.; Soto, F.; Sanchez, P.; Iborra, A; Suardiaz, J; Vera, J.A. Development of a sensor node for precision horticulture. *Sensors* 2009, 9, 3240-3255.
- [3] eKo Outdoor Wireless System for Environmental Monitoring; Crossbow Technology Inc.: Milpitas, CA, USA, 2009; Available online:
- [4] Kim, Y; Evans, R.G.; Iversen, W.M. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEE Trans. Instrum. Meas.* 2008, 57, 1379-1387.
- [5] Chavez, J.L.; Pierce, F.J.; Elliott, T.V. A remote irrigation monitoring and control system for continuous move systems. Part A: Description and development. *Precis. Agric.* 2010, 11, 1-10.
- [6] Demirkol, I.; Ersoy, C.; Alagoz, F., MAC protocols for wireless sensor networks: a survey, *Communications Magazine, IEEE* , vol.44, no.4, pp. 115-121, April 2006.

- [7] Kuorilehto, M.; Suhonen J. et al. *Ultra-Low Energy Wireless Sensor Networks in Practice*, Wiley, ISBN 978-0-470-05786-5, 2007
- [8] Sichitiu, M.L. Veerarittiphan, C., Simple, accurate time synchronization for wireless sensor networks. *Wireless Communications and Networking*, 2003. 20 (March 2003), 1266-1273 vol.2.
- [9] Elson, J. and Römer, K. 2003. Wireless sensor networks: a new regime for time synchronization. *SIGCOMM Comput. Commun. Rev.* 33, 1 (Jan. 2003), 149-154.
- [10] Faizulkhakov, Y.R. Time synchronization methods for wireless sensor networks: A survey. *Program. Comput. Soft.* 2007, 33, 214-226.
- [11] Sommer, P.; Wattenhofer, R. Gradient clock synchronization in wireless sensor networks. In *Proceeding of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, San Francisco, CA, USA, 13-16 April 2009.
- [12] Sundararaman, B.; Buy, U.; Kshemkalyani, A.D. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Netw.* 2005, 3, 281-323.
- [13] CC2420, 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver DataSheet; Texas Instruments: Dallas, TX, USA, 2009.
- [14] Alberola, J.; Pelegri, J.; Lajara, R.; Perez, J.J., Solar Inexhaustible Power Source for Wireless Sensor Node, *Instrumentation and Measurement Technology Conference Proceedings*, 2008. IMTC 2008. IEEE , vol., no., pp.657-662, 12-15 May 2008
- [15] Pelegri J.; Ramírez D., Circuit latches solenoid at a distance. *EDN Magazine*, 14/10/1999.
- [16] HC-49US Quartz Crystal DataSheet; ECS International Inc.: Olathe, KS, USA, Available online: [http://www.ecsxtal.com/store/pdf/hc\\_49us](http://www.ecsxtal.com/store/pdf/hc_49us)
- [17] Enhanced Flash Microcontroller with 10-Bit A/D and nanoWatt Technology DataSheet; PIC18F4620.
- [18] 90SMX & 91 SMX Crystals DataSheet IQD Frequency Products: Crewkerne, UK, 2009.
- [19] Lajara, R.; Pelegri-Sebastia, J.; Perez-Solano, J.J. Power consumption analysis of operating systems for wireless sensor networks. *Sensors* 2010, 10, 5809-5826.
- [20] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1567-1576). IEEE.

# 8. ESTACIÓN METEOROLÓGICA DE ULTRA-BAJO CONSUMO

---

La aparición de *transceivers* inalámbricos de muy bajo consumo a precios competitivos es uno de los principales motivos del interés en realizar aplicaciones de redes de sensores inalámbricos. En este capítulo se presenta el diseño de un nodo sensor inalámbrico de pequeño tamaño y bajo coste que implementa una estación meteorológica. Esta estación meteorológica mide varias variables con alta precisión: temperatura ( $\pm 0,1$  °C), humedad relativa ( $\pm 3$  %), intensidad luminosa y presión atmosférica ( $\pm 0,8$  hPa). De estas variables se pueden derivar otras como el punto de rocío, la sensación térmica, etc. y pueden ser accedidas a través de una página web. El nodo sensor consigue un muy bajo consumo (40  $\mu$ A de media), permitiendo que un simple supercondensador lo alimente durante 52 días.

## 8.1 Introducción

El estudio meteorológico y climático es de vital importancia en muchas áreas de investigación, como las que estudian los desplazamientos de animales, los que estudian el cambio climático, etc. Su análisis requiere conocer cómo varían varios parámetros, tales como la temperatura, humedad, presión atmosférica y otros como la concentración de CO<sub>2</sub>. Por otra parte, algunos sectores económicos como el turismo y la agricultura dependen fuertemente del clima, por tanto un conocimiento previo de éste les ayudará a planificarse mejor.

Existen varias estaciones climatológicas comerciales, como por ejemplo las de las empresas Oregon, Davis o Vaisala. Algunas de ellas son inalámbricas, con mejor o peor precisión y relativamente autónomas, pero usualmente las que tienen todas estas características son muy caras, de gran tamaño y de alto consumo energético.

Las redes de sensores inalámbricos basados en el estándar IEEE 802.15.4 [1] están construidas por nodos de muy bajo consumo, bajo coste y bajas tasas de transmisión. Se han utilizado en muchos ámbitos de aplicación, como monitorización de animales [2][3], monitorización de entornos ambientales [4] e investigación clínica y médica [5], entre otras muchas. Por lo tanto, una aplicación como una estación climatológica inalámbrica se adapta muy bien a las características de las redes de sensores inalámbricas.

## 8.2 Descripción del sistema

En la Figura 8-1 se puede ver un esquema conceptual del sistema. Un nodo sensor de pequeño tamaño y autónomo transmite la información de los parámetros sensados. Estos parámetros son la temperatura, humedad relativa, presión atmosférica e intensidad lumínica. Estos datos son recogidos por una estación base, la cual está continuamente conectada a una alimentación ininterrumpida y a un ordenador. Los datos de la estación base son retransmitidos por cable al ordenador, el cual los almacena y genera varios ficheros, que pueden ser accedidos a través de un servidor web.

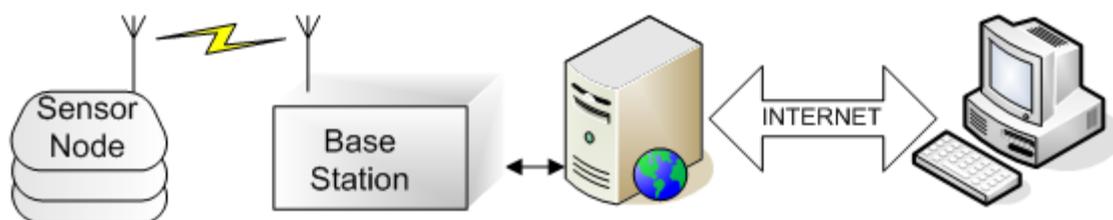


Figura 8-1: Esquema general del sistema.

Las siguientes secciones discutirán en detalle la arquitectura del sistema y el interfaz web.

### 8.2.1 Hardware

En esta sección se estudiará específicamente la arquitectura hardware del nodo sensor que implementa la estación meteorológica porque es la que presenta mayor interés. La estación base simplemente actúa como un *gateway* y su hardware es básico. En la Figura 8-2 se muestran todos los bloques internos del nodo sensor, que se describirán a continuación.

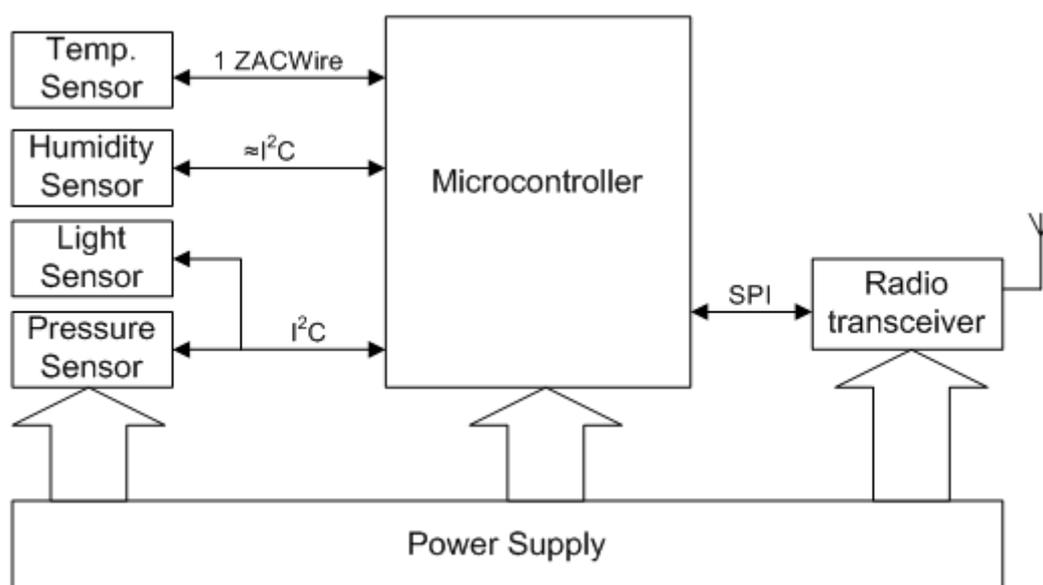


Figura 8-2: Diagrama de bloques del hardware del nodo que implementa la estación meteorológica inalámbrica.

El nodo sensor utiliza una fuente de energía ambiental a través de un sistema electrónico diseñado para esta aplicación específica basado en múltiples caminos de corriente [6]. Esta fuente utiliza energía solar para cargar un supercondensador, el cual tiene una rápida recarga y mínimo desgaste en comparación con otros sistemas como las baterías de litio. El esquema es el correspondiente al diseño del capítulo 5.

El consumo energético ha sido uno de los puntos clave al diseñar la estación meteorológica, ya que es lo que permite que sea autónoma. Con esta premisa se han seleccionado los componentes hardware.

El *transceiver* inalámbrico para implementar las comunicaciones ha sido el CC2420 [7] de Chipcon, posteriormente Texas Instruments. Este *transceiver* es compatible con el estándar IEEE 802.15.4, el cual define las capas física y MAC del modelo OSI. Utiliza una modulación de anchura de espectro (DSSS) con 0-QPSK en 2,4 GHz, permitiendo un ratio de transferencia de 250 kbps, una topología en estrella o *peer-to-peer* y un rango de cobertura alrededor de 120 m transmitiendo a 0 dBm en espacio abierto [8]. El *transceivers* puede ser manejado directamente por un microcontrolador a través del puerto SPI.

La unidad de procesamiento del nodo sensor es un microcontrolador PIC18LF4620 [9] de Microchip. Este microcontrolador utiliza la tecnología nanoWatt de Microchip que permite un muy bajo consumo a través de varios modos de funcionamiento.

Los sensores han sido seleccionados también primando el bajo consumo, y además buscando una alta precisión y un coste relativamente bajo. Se han primado por interfaces digitales para minimizar el número de componentes necesarios y, por tanto, el tamaño final del nodo implementado; además esto permite simplificar los mecanismos de calibración. La estación meteorológica inalámbrica mide temperatura, humedad, intensidad lumínica y presión atmosférica. Cada sensor tiene un interfaz diferente, como se puede ver en la Figura 8-2. A continuación se comentará cada sensor en particular.

El sensor de temperatura es un TSIC506F [10], el cual tiene una precisión de  $\pm 0,1$  °C y utiliza un sistema de comunicación propietario basado en el protocolo de un hilo 1-ZACWire.

El sensor de humedad es un SHT11 de Sensirion [11], que tiene una precisión de  $\pm 3$  % en un amplio rango. Los valores medidos deben ser compensados para temperaturas significativamente diferentes de 25 °C utilizando la ecuación (8.1), donde  $RH_{true}$  es el valor de humedad relativa compensado;  $T_c$  es la temperatura;  $t_1$  y  $t_2$  son coeficientes constantes que dependen del sensor;  $SO_{RH}$  es el valor digital dado por el convertidor ADC del sensor; y  $RH_{linear}$  es el valor de humedad relativa sin la compensación. Además, es posible calcular el punto de rocío [12] usando la temperatura y la humedad relativa, tal y como se describe en las ecuaciones (8.2) y (8.3), donde  $a$ ,  $b$  y  $c$  son constantes. Este sensor utiliza un protocolo propietario basado en el I<sup>2</sup>C. También se puede comentar que integra la medida de temperatura, pero no se utilizará ya que el sensor mencionado antes tiene características superiores.

$$RH_{true} = (T_c - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linear} \quad (8.1)$$

$$H = \left( \frac{\log(\text{RH}_{\text{true}}) - 2}{a} + \frac{T_o C \cdot b}{T_o C + c} \right) \quad (8.2)$$

$$\text{DP} = \frac{H \cdot c}{b - H} \quad (8.3)$$

El sensor de luz es el ISL29001 [13], que consiste en dos fotodiodos, uno de ellos sensible a la luz visible e infrarroja (IR), mientras que el otro diodo es usado para realizar una compensación interna de la temperatura (cancelación de la corriente de *leakage*) y rechazo de IR. Tiene una resolución efectiva de 15 bits y da una medida directamente en lux, en el rango de 0,3 a 10.000. Utiliza un interfaz I<sup>2</sup>C.

El sensor de presión es el SMD500 [14], que tiene una precisión de ±0,8 hPa y compensación de temperatura. La altitud también se puede calcular usando la ecuación (8.4), donde *p* es la presión real; y *p<sub>0</sub>* es la presión a nivel del mar. Este sensor ha sido diseñado para conectarse directamente a un microcontrolador a través del interfaz I<sup>2</sup>C.

$$\text{Altitud} = 44330 \cdot \left( 1 - \left( \frac{p}{p_o} \right)^{\frac{1}{5.255}} \right) \quad (8.4)$$

Las características más importantes de los sensores se muestran resumidas en la Tabla 8-1.

**Tabla 8-1: Características principales de los sensores utilizados.**

Sensor	TSIC506F	SHT11		ISL29001	SMD500		
Parámetros	Temp. (°C)	Humedad relativa (%)	Temp. (°C)	Luminancia (lux)	Presión atmosférica (hPa)	Temp. (°C)	Altura (m)
<b>Precisión@25°C</b>	± 0,1 <sup>1</sup>	± 3	± 0.4	?	± 0,8..2,5	± 1,5	?
<b>Rango</b>	-10 ...+60	0..100%	-40..123.8	0.3..10000	300..1100	-40..+85 -20..+60	+9000.. -500
<b>Resolución</b>	0,034	0,03%	0,01	3-15 cuentas	0,06 0,03		0,5 m, 0,25 m
<b>Bits efectivos</b>	11 bits	8,12 bits	12,14 bits	15,16 bits	16 bits		
<b>Repetitividad</b>	--	± 0.1%	± 0.1°C	--	--		
<b>Alimentación DC</b>	2,97 - 5,5 V	2,4 - 5,5 V		2,5 - 3,3 V	2,2 - 3,6 V		
<b>Corriente consumida @25°C</b>	30,45,80µA <sup>2</sup>	0.3,28,550µA <sup>3</sup>		1 µA	5 µA,10 µA(1sample/s),0.1µA(sleep)		
<b>Tiempo de medida</b>	0,1 s	55 ms		200 ms	34 ms @ 32.768 kHz ± 3%		
<b>Tiempo de respuesta</b>	--	4 s	5-30 s	100 ms	--		
<b>Interfaz</b>	1 ZACWire	Similar a I2C		I2C	I2C		

El esquemático del nodo se puede consultar en la sección 10.2.4 de los anexos de esta tesis.

<sup>1</sup> En un rango de 40 °C y Vcc = [4.5V,5.5V].

<sup>2</sup> Mínimo, típico y máximo @ 25°C y Vcc = 5V.

<sup>3</sup> Dormido, consumo medio, midiendo.

## 8.2.2 Software

En los siguientes apartados se describen las principales estrategias seguidas para reducir el consumo de energía a través del software y cómo se muestra la información recogida.

### 8.2.2.1 *Nodo sensor*

El programa que se ejecuta en el nodo sensor se puede resumir en: despertar, leer los sensores, construir una trama de datos, enviar los datos y volver a dormir. Se intentará que esté dormido la mayor parte del tiempo y en el modo más profundo posible. La red utiliza un mecanismo de planificación dormido-despierto [15]. De esta forma, el consumo de energía puede ser reducido si se maximiza el tiempo en el estado dormido. Por tanto este es un factor clave en el diseño que se ha tenido en cuenta en el programa del nodo sensor.

Como cada sensor tiene un tiempo de respuesta diferente, por tanto es necesario establecer un orden de activación de los sensores que reduzca el tiempo activados y, como consecuencia, el consumo de energía. Utilizando la información de la Tabla 8-1 se puede planificar la activación que se muestran en la Figura 8-3.

Inicialmente todos los sensores están apagados y los puertos del microcontrolador configurados como alta impedancia, la cual reduce el consumo. Llegado el momento, el microcontrolador se despierta y los puertos son configurados convenientemente para permitir la comunicación con los sensores. Cuando el microcontrolador se despierta, los sensores comienzan a realizar sus lecturas, el microcontrolador las lee y graba los datos en su memoria RAM. Una vez que los valores han sido obtenidos, los sensores se van apagando, se configuran las líneas nuevamente como alta impedancia y se vuelve al estado dormido.

```

Loop{
    Config.Ports ( in-out )
    Init.Sensors ( )
    Delay 34ms
    Measure.Pressure ( )
    Delay 21ms
    Measure.Humidity ( )
    Delay 10ms
    Measure.Temperature ( )
    Delay 55ms
    Measure.Light ( )
    Config.Ports ( Hi-Z )
    If measures = 10 then {
        BuildPacket ( )
        Transceiver ( on )
        SendPacket ( )
        Transceiver ( off )
    }
    Sleep ( 32,768s )
}

```

**Figura 8-3: Planificación de la lectura de los sensores en el nodo sensor. El orden optimiza el consumo energético.**

El nodo sensor y la estación base forman una red con topología de estrella. El flujo de datos es unidireccional, del nodo sensor a la estación base y, como se ha mencionado al principio, ésta tiene una conexión asegurada a la alimentación, por lo que puede estar continuamente a la escucha. El hecho de que la estación base pueda estar siempre activada hace que no sea

necesario un mecanismo de sincronización y únicamente será necesario un mecanismo de acceso al medio como CSMA/CA, el cual es utilizado con los mismos condicionantes en el protocolo WiseMAC [16]. De esta forma, cuando se envíe el último valor de los sensores del nodo éste volverá a dormirse; posteriormente se podrá despertar a través de un mecanismo de *watchdog*. Este mecanismo tiene un reloj de baja precisión, pero no es ningún inconveniente para el protocolo de comunicación en este caso y hace que el microcontrolador tenga un consumo en estado dormido inferior a 2  $\mu\text{A}$ .

Cuando se han adquirido diez muestras de cada uno de los sensores, se construye una trama con un campo de identificación, la tensión de la batería y los parámetros meteorológicos. A continuación el *transceiver* es activado y la trama se envía a la estación base. Cuando se ha enviado la trama el *transceiver* se vuelve a apagar. De esta forma se completa un ciclo de lectura y envío de información. El *transceiver* es el elemento que más energía consume, y lo hace a través de dos mecanismos: la propia transmisión por RF y la comunicación con el microcontrolador. El primer factor es obvio, y depende fundamentalmente de la potencia de transmisión y, en último término, de la distancia entre el emisor y el receptor. El segundo mecanismo es el menos obvio y se debe a las resistencias de *pull-up* internas que tiene el microcontrolador en las líneas SPI. Activando únicamente estas resistencias en la transmisión se ahorran unos 500  $\mu\text{A}$  de corriente de media.

### 8.2.2.2 Interfaz web

Cuando la estación base recibe una trama del nodo sensor, los datos se retransmiten inmediatamente a través de un cable serie a un ordenador. El ordenador actúa como un servidor web, permitiendo la visualización remota de los datos recolectados. El ordenador tiene dos piezas de software: un programa en LabVIEW que recibe, interpreta y almacena los datos y un servidor web Apache que los sirve.

El programa en LabVIEW maneja el puerto serie para recibir las tramas retransmitidas por la estación base. Después el programa divide la trama en sus respectivos valores de temperatura, humedad, luz y presión. Estos datos son procesados y se realizan sus correspondientes compensaciones. Además, también se calculan los parámetros indirectos mencionados antes, tales como el punto de rocío —ecuación (8.2) y (8.3)— y el índice de calor —ecuación (8.5)—.

$$\begin{aligned} HI = c_1 + c_2 T + c_3 H + c_4 TH + c_5 T^2 + c_6 H^2 + \\ + c_7 T^2 H + c_8 TH^2 + c_9 T^2 H^2 \end{aligned} \quad (8.5)$$

Donde  $HI$  es el índice de calor;  $c_1$  a  $c_9$  son constantes;  $T$  es la temperatura; y  $H$  es la humedad relativa.

Además, el programa de LabVIEW genera varios ficheros que son almacenados en el ordenador, que son a los que se accederá a través de internet mediante el servidor Apache. Se han implementado diferentes formas de acceder a la información. La primera de ellas es a través de un simple fichero PNG (*Portable Network Graphics*), el cual contiene la imagen más reciente de la evolución de los parámetros y los últimos valores. El segundo método utiliza un fichero javascript, el cual se puede referenciar desde páginas de terceros y su funcionamiento

consiste en 'embeber' en la página una herramienta que muestre los últimos valores. El tercer método es un fichero RSS (*Really Simple Syndication*), que también proporciona los valores instantáneos. Y el último método es una base de datos que contiene todo el histórico de valores de todas las variables medidas. De esta forma no sólo se proporcionan valores, sino que se ofrecen servicios a páginas web o aplicaciones de terceros.

### 8.3 Resultados

La estación meteorológica ha estado funcionando de forma autónoma durante meses en la azotea de un edificio de la universidad enviando datos al servidor web. Puede verse el aspecto de una página web servida en la Figura 8-4.

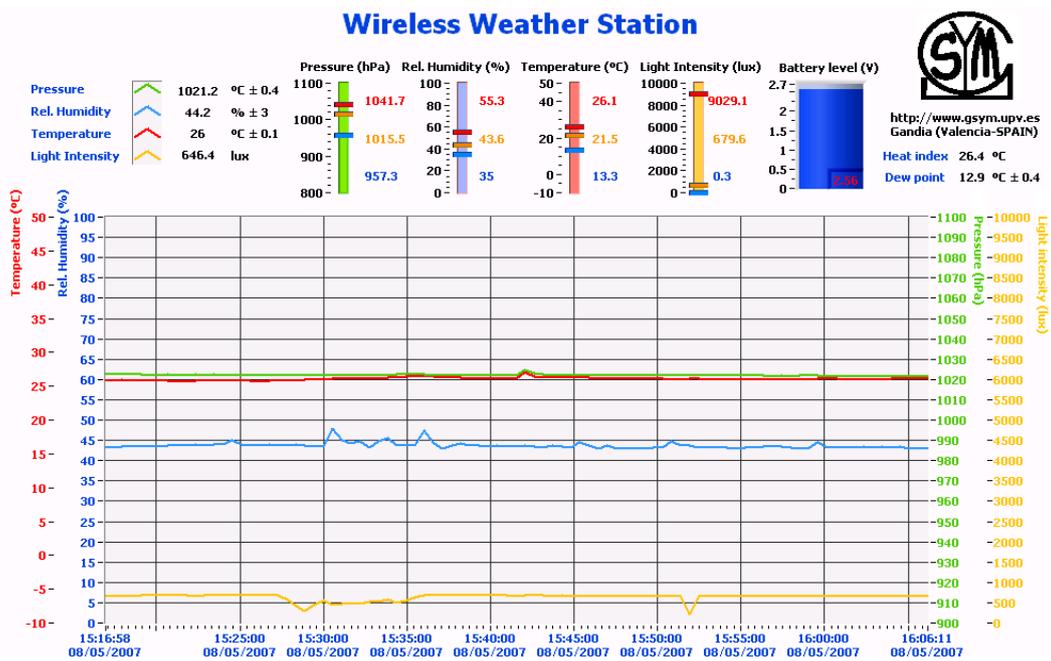


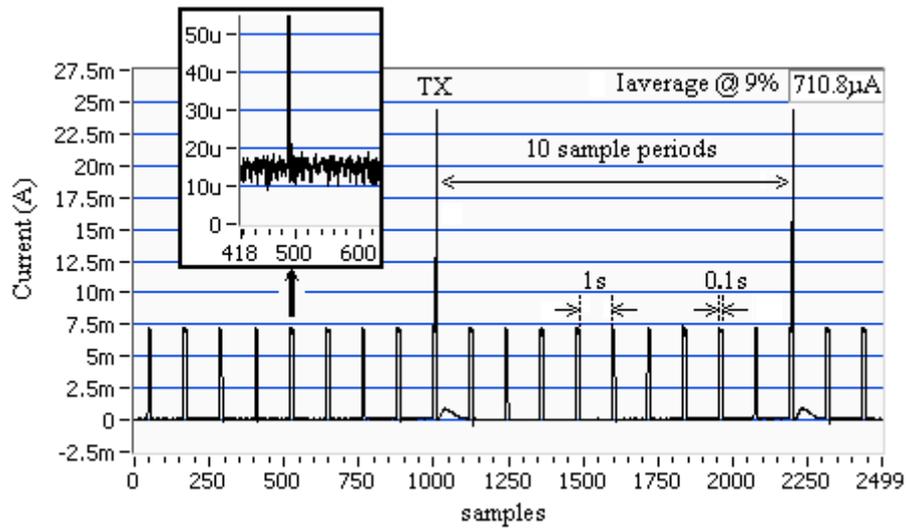
Figura 8-4: Interfaz gráfico que muestra la variación de las variables meteorológicas. Es el primero de los métodos de acceso a los datos.

La aplicación no requiere altas frecuencias de muestreo, por lo que se ha escogido un ciclo de trabajo del 1% con una frecuencia de muestreo de 30 segundos, la cual da una frecuencia de refresco de datos de 5 minutos (10 muestras por trama). Con esta configuración, el consumo medio del nodo sensor es extremadamente bajo ( $I_{av} = 40 \mu A$ ), un valor más cercano al consumo en el modo dormido ( $I_{off} = 15 \mu A @ t_{off} = 32,768 s$ ) que al modo despierto ( $I_{on} = 7,4 mA @ t_{on} = 0.29s$ ) o al modo transmitiendo ( $I_{tx} = 25mA @ t_{tx} = 0.012s$ ). El consumo medio se puede aproximar mediante la ecuación (8.6).

$$I_{av} = \frac{10(t_{on} \cdot I_{on} + t_{off} \cdot I_{off}) + t_{tx} \cdot I_{tx}}{10(t_{on} + t_{off}) + t_{tx}} \quad (8.6)$$

En la Figura 8-5 se muestra una captura de una medida del consumo de corriente del nodo funcionando con un ciclo de trabajo del 9% y 1 segundo de frecuencia de muestreo, el cual se cambió respecto al del funcionamiento normal para facilitar la medida ya que los picos de

consumo con un ciclo del 1% eran demasiado cortos para la resolución temporal del sistema de medida y no podían capturarse.



**Figura 8-5: Consumo de corriente en un nodo sensor de la estación meteorológica inalámbrica alimentado a 3,3 V. El microcontrolador se despierta cada segundo y obtiene una muestra de cada sensor. Después de 10 muestras se envía una trama de datos.**

En condiciones normales de funcionamiento, es decir, con un ciclo de trabajo del 1%, el consumo medio es de 40 µA. Esto posibilita que el sistema esté alimentado a través de un simple supercondensador durante 13 días sin necesidad de ninguna recarga solar, de acuerdo con la estimación de la ecuación (4.4) [17].

$$t = \frac{C \cdot (V_{\max}^2 - V_{\min}^2)}{I_{\text{load}} \cdot (V_{\max} + V_{\min})} \quad (8.7)$$

Donde:

$C$  es la capacidad nominal del supercondensador en Faradios

$I_{\text{load}}$  es el consumo medio de la carga

$V_{\max}$  y  $V_{\min}$  son los umbrales de tensión en los cuales el nodo puede funcionar

$t$  es el tiempo de autonomía.

Los valores numéricos han sido 25 F para el supercondensador (dos condensadores en serie de 50 F cada uno), 40 µA, 2,7 V y 0,9 V. En esta estimación se han considerado todos los componentes ideales y sin pérdidas. Se pueden ver más detalles de esta estimación en el capítulo 5.

Por otro lado, el tamaño de la placa del nodo sensor ha sido también muy pequeño, de únicamente 4,4 x 6,6 x 1 cm, como se puede ver en la Figura 8-6.

El rango de cobertura es de alrededor de 120 metros por nodo en campo abierto, aunque con obstáculos en el camino puede disminuir notablemente.

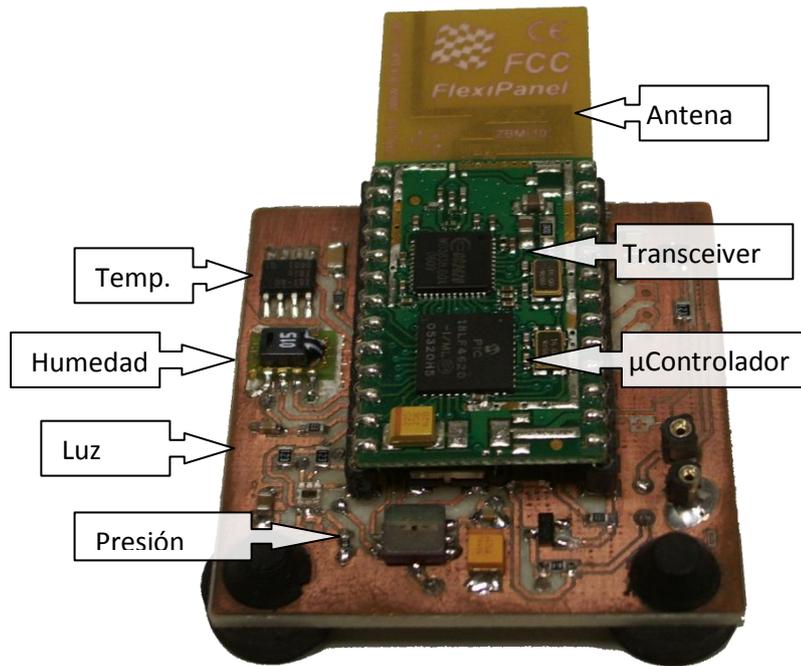


Figura 8-6: Fotografía del nodo sensor. Se muestra la localización de los cuatro sensores, el microcontrolador y el *transceiver*. El conector de programación está en la parte inferior.



Figura 8-7: Fotografía del nodo sensor (izquierda) conectado a la fuente de alimentación del capítulo 5 (derecha), al panel solar (arriba) y dentro de una carcasa protectora.

## 8.4 Conclusiones

Se ha diseñado y probado un prototipo de estación meteorológica inalámbrica. Ha funcionado proporcionando precisa información de temperatura, humedad relativa, presión atmosférica e intensidad luminosa durante meses.

La aplicación no requiere mecanismos de sincronización. De esta forma todos los componentes del nodo sensor, excepto el temporizador del *watchdog* pueden desactivarse durante el estado dormido. Esto ahorra una gran cantidad de energía y posibilita un consumo medio

extremadamente bajo. Normalmente el oscilador del *watchdog*, basado en un oscilador RC, no es adecuado para la medida precisa de tiempos, pero en el caso de la aplicación y la arquitectura utilizadas no supone inconveniente.

El tamaño de la electrónica resultante ha sido muy pequeño, pero todavía puede reducirse más ubicando los sensores bajo el módulo PICZEE, el cual es de 2 x 5,4 cm.

La presión barométrica puede resultar de especial interés para la investigación de modelos regionales de predicción climatológica. Además, se puede ampliar con la introducción de nuevos sensores, por ejemplo para medición de lluvia utilizando sensores piezoeléctricos [18], los cuales no incrementarán el consumo ni el tamaño. También se puede añadir un sistema de medición de la velocidad del viento mediante el estudio del enfriamiento empleando el sensor de temperatura del SHT11; así se puede utilizar un sistema interno de calentamiento en este sensor para que lo caliente por encima de la temperatura ambiente y se mida el tiempo necesario para enfriarse 1 °C y encontrar la relación entre la velocidad del viento y del enfriamiento como se describe en [19]. Al mismo tiempo, esto permitiría derivar también el parámetro de sensación térmica. Finalmente, teniendo en cuenta que la estación meteorológica se instaló en la playa de Gandía, podría ser interesante añadir un sensor de radiación ultravioleta.

### 8.5 Bibliografía

- [1] IEEE Standard for Information technology-Telecommunications and information exchange between systems-PART 15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks. October 2003. ISBN: 0-7381-3677-5.
- [2] Noda, C., Fernández, J., Pérez-Penichet, C., & Altshuler, E. (2006). Measuring activity in ant colonies. *Review of scientific instruments*, 77(12), 126102.
- [3] Zhang, P., Sadler, C. M., Lyon, S. A., & Martonosi, M. (2004, November). Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 227-238). ACM.
- [4] Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., & Welsh, M. (2005, January). Monitoring volcanic eruptions with a wireless sensor network. In *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on* (pp. 108-120). IEEE.
- [5] Park, C., Liu, J., & Chou, P. H. (2005, April). Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* (pp. 398-403). IEEE.
- [6] Jiang, X., Polastre, J., & Culler, D. (2005, April). Perpetual environmentally powered sensor networks. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* (pp. 463-468). IEEE.
- [7] Texas Instruments. Datasheet CC2420. (online document, available in <http://www.ti.com/lit/ds/symlink/cc2420.pdf>)
- [8] Flexipanel. Pixie DS481-11 Datasheet. (online document, available in <http://www.FlexiPanel.com>)

- [9] Microchip Inc. PIC18LF4620 datasheet. (online document, available in <http://www.microchip.com>)
- [10] Innovative Sensor Technology (ISL). ISL TSic506F datasheet. (online document, available in <http://www.ist-ag.com>)
- [11] Sensirion. SHT1x datasheet. (online document, available in <http://www.sensirion.com>)
- [12] Sensirion. Application Note Dew-point calculation. (online document, available in <http://www.sensirion.com>)
- [13] Intersil. ISL 29001 datasheet. (online document, available in <http://www.intersil.com>)
- [14] Bosh Sensortec GmbH. SMD500 datasheet. (online document, available in <http://www.bosch-sensortec.com>)
- [15] Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1567-1576). IEEE.
- [16] El-Hoiydi, A., & Decotignie, J. D. (2004, July). WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on* (Vol. 1, pp. 244-251). IEEE.
- [17] Power Stor, Cooper Bussmann. Application Note: "Design Considerations In Selecting Aerogel Supercapacitors"
- [18] Salmi, A., Ikonen, J., & Oyj, V. (2005, May). Piezoelectric precipitation sensor from Vaisala. In *WMO Technical Conference on Instruments and Methods of Observation (TECO-2005), Bucharest, Romania* (pp. 4-7).
- [19] Makinwa, K. A., & Huijsing, J. H. (2002, February). A smart CMOS wind sensor. In *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International* (Vol. 1, pp. 432-479). IEEE.



# 9. CONCLUSIONES

---

## 9.1 Conclusiones generales

En esta tesis se ha visto la importancia y necesidad de la gestión y mejora del consumo de energía en los nodos que componen las redes de sensores inalámbricas.

Se ha investigado en lo relativo al consumo energético en los nodos finales de las redes de sensores. Se ha llevado a cabo una búsqueda de las plataformas, tanto hardware como software, y se han comparado para ver su respuesta en diferentes escenarios. En este análisis se ha visto que el sistema operativo que en general está más optimizado en lo relativo al consumo es TinyOS, que además es el más extendido. Esto lo consigue mediante su paradigma de programación orientado a eventos y a la poca carga extra que supone. La plataforma hardware más optimizada ha resultado ser Tmote Sky; la principal razón no es que sus modos de bajo consumo tengan un nivel extraordinariamente bajo, sino el poco tiempo de *wakeup* cuando sale de ellos. Combinando Tmote Sky y TinyOS se pueden realizar aplicaciones en las que el sistema operativo lleve al microcontrolador inmediatamente a dormir en cuanto tenga oportunidad, por ejemplo entre el inicio de una adquisición analógica-digital y su finalización, lo que ahorra una importante cantidad de energía en comparación con la realización de esperas activas.

Después se ha propuesto un método para modelar el consumo de nodos alimentados por baterías recargables de litio y se han propuesto varios modelos con diferentes prestaciones en cuanto a respuesta y cómputo.

Respecto al modelado del consumo energético en baterías se ha visto que uno de los sistemas más habituales: leer simplemente la tensión de la batería, es un método que puede no proporcionar la información correcta, ya que se asume que la caída de tensión es lineal en todo el rango y esto no es así. En aplicaciones donde es tan crítico el consumo, como es el caso de las redes de sensores, es necesario tener un conocimiento más preciso del estado energético del nodo. Para ello los modelos propuestos mediante la metodología explicada permiten determinar mejor el estado del nodo. Algunos de los modelos propuestos llegan a presentar una correlación de 0,99 respecto al estado real de carga. Estos modelos, o bien no necesitan hardware adicional, o necesitan una red trivial de resistencias entre la batería y el nodo.

Un parámetro que habitualmente no se tiene en cuenta al diseñar redes de sensores es el desgaste de la batería. Si los nodos sensores tienen baterías recargables, éstas se desgastarán en cada ciclo de carga y descarga. Se ha visto la necesidad de prever este hecho en el capítulo

3 y en el 5 se ha propuesto un diseño de fuente de alimentación que lo minimiza. Para ello utiliza un sistema mixto de batería como *backup* y supercondensador como elemento principal de almacenamiento y un sistema de conmutación entre ellos totalmente automático. Además la fuente también evita recarga esporádicas de la batería para no realizar ciclos innecesarios.

Después se ha explorado el campo del *energy harvesting* utilizando la energía RF en el ambiente procedente de emisoras comerciales de radio y televisión. En los resultados de este capítulo se ha visto que es posible extraer energía que, si bien es de escasa cantidad —8  $\mu\text{A}$ —, ésta puede ser mantenida en el tiempo; por ello esta fuente de energía puede servir de refuerzo a otra fuente de energía más eficiente y puntual.

Además se han explorado diversas configuraciones de circuitos rectificadores y dobladores de tensión, proponiendo modificaciones para permitir la sintonización y lograr aumentar la eficiencia.

Podemos decir que la mejora del consumo posibilita la creación de nuevas o más optimizadas aplicaciones, por ello los siguientes capítulos se han dedicado a realizar aplicaciones para la monitorización de parámetros ambientales y la gestión de regadíos.

En el capítulo 7 se ha realizado una aplicación en la que los nodos finales controlaban una electroválvula. Esta aplicación no es tan habitual en las redes de sensores, ya que normalmente sólo monitorizan. En esta ocasión los nodos tienen capacidad de actuar sobre el sistema. Esta actuación consistía en abrir o cerrar una electroválvula, algo para lo que se necesitan picos de corriente relativamente elevados. Se ha diseñado un sistema basado en condensadores y supercondensadores para realizar las activaciones de la electroválvula de forma eficiente.

Además, el sistema propuesto tiene la novedad de utilizar exclusivamente supercondensadores, ya que la alimentación a través de paneles solares permite prescindir totalmente de baterías de litio.

En el capítulo 7 también se ha visto el impacto energético que supone el mantenimiento de la sincronización entre nodos adyacentes en las redes de sensores. Para ello se ha implementado el protocolo de acceso al medio SMAC, pero con una modificación hardware para reducir el consumo. Esta modificación consiste en utilizar un reloj de baja frecuencia pero alta precisión, cuya frecuencia es dividida hasta el periodo de trabajo del nodo. Este mecanismo permite llevar al procesador a un modo *sleep* profundo ya que no necesita activar ningún contador internamente para despertarse, lo hará mediante una interrupción del reloj externo. Y a su vez es posible mantener la sincronización temporal entre nodos, porque el reloj es más preciso que los típicos osciladores RC que tienen los *watchdogs* típicos.

Finalmente en el capítulo 8 se ha propuesto una aplicación de monitorización ambiental. Para ello se ha propuesto un nodo final que implementa una estación meteorológica. El funcionamiento básico del sistema es sensor temperatura, humedad relativa, presión atmosférica y luminiscencia y enviarlas de forma inalámbrica. En esta aplicación se ha conseguido un consumo medio extremadamente bajo, de 40  $\mu\text{A}$  con un ciclo de trabajo del 1%, incluyendo la transmisión inalámbrica a 120 m en campo abierto. Además de este consumo

tan bajo, todavía se podría reducir más bajando el ciclo de trabajo y acercarlo más al consumo en su estado *sleep*, que es de unos 13  $\mu\text{A}$ .

En vista de los resultados se puede decir que este tipo de aplicaciones de tan bajo consumo pueden utilizar casi exclusivamente fuentes de *energy harvesting*. Además también se puede plantear la posibilidad, al igual que en el caso de la electroválvula, de prescindir de baterías para utilizar condensadores o supercondensadores. Hay que recordar que los resultados preliminares del sistema basado en energía RF eran de 8  $\mu\text{A}$ , lo que indica que en un futuro podría ser viable alimentar este tipo de aplicaciones con esta fuente de energía.

Además, a la fuente meteorológica se le ha habilitado el acceso a la web a través de su *gateway*, por lo que cualquier persona podría acceder a sus datos. Esto supone una primera toma de contacto con el concepto de internet de las cosas (IoT) y amplía sobremanera las posibilidades y servicios que las redes de sensores pueden proporcionar.

## 9.2 Trabajo futuro

El estudio y la optimización energética en las redes de sensores es un tema todavía abierto, y siempre podrá optimizarse más. En este aspecto, una primera línea de investigación sería estudiar el consumo, no ya a nivel de los nodos en particular, sino a nivel global, para toda la red.

En el capítulo del modelado de consumo se ha mencionado el uso de la medida de resistencia interna para estimar el desgaste de las baterías. Una línea de trabajo futuro sería estudiar cómo poder realizar esta medida en nodos de redes de sensores y estimar cuánta energía implicaría realizar dicha medida. Para hacer esta medida habría que conseguir una diferencia de tensiones y corrientes en la batería. Por ello la fuente de alimentación presentada en el capítulo 5 podría ser adecuada, ya que tiene dos elementos de almacenaje y se puede conmutar entre la batería y el supercondensador. También se puede aprovechar la distinta corriente consumida en los estados de los nodos para lograr esta diferencia, como se propone en la referencia 5 de la siguiente página.

También se pueden desarrollar otros modelos, parecidos a los de la bibliografía del capítulo 5, para cuantificar la energía consumida en cada subsistema del nodo sensor: *transceiver*, procesador, sensores, etc. Conociendo el estado energético y el coste de cada tarea se podría establecer una planificación de tareas que evalúe el coste y el beneficio de determinadas acciones.

Además otra vía abierta en este capítulo sería aplicar los modelos propuestos a simuladores de redes de sensores para que desde su aplicación puedan acceder al nivel de tensión real que devolverían sus baterías.

Desde el capítulo sobre *energy harvesting* se pueden iniciar muchas líneas de investigación ya que se trata de un estudio de viabilidad. Como esta viabilidad se ha demostrado positiva se puede trabajar en nuevas configuraciones de circuitos, otros anchos de banda, otras frecuencias, nuevas antenas, etc.

Por otro lado también se puede combinar la energía obtenida por el sistema de RF con otras como por ejemplo la energía solar o de sensores piezoeléctricos. Esta combinación daría lugar a una nueva fuente de alimentación que pueda aunar las corrientes de varias fuentes diferentes para cargar un único componente.

En la aplicación de nodos para irrigación se podría estudiar cómo extraer energía del flujo de agua que se deja pasar. Para ello se podrían utilizar o bien rotores o bien materiales piezoeléctricos que aprovechen el movimiento del agua por las tuberías.

En el capítulo de la estación climatológica se mencionó que los datos recopilados se servían a través de un servidor web y un servidor de bases de datos. Esta es una característica que nos introduce en la forma de trabajar del internet de las cosas. Este campo se puede explotar con nuevas arquitecturas y aplicaciones.

### 9.3 Contribuciones y publicaciones

Los resultados generados por el trabajo durante la realización de esta tesis han propiciado la generación de una serie de publicaciones presentadas en revistas, dentro del Q1 de su área, capítulos de libros y presentaciones en diferentes congresos.

Artículos:

1. **Lajara, R.**, & Solano, J. J. P. Pelegrí-Sebastiá, J. (2014). A method for modeling the battery state of charge in wireless sensor networks. *IEEE Sensors (submitted)*.
2. **Lajara, R.**, Pelegrí-Sebastiá, J., & Solano, J. J. P. (2010). Power consumption analysis of operating systems for wireless sensor networks. *Sensors*, 10(6), 5809-5826.
3. **Lajara, R.**, Alberola, J., & Pelegrí-Sebastiá, J. (2010). A solar energy powered autonomous wireless actuator node for irrigation systems. *Sensors*, 11(1), 329-340.

Capítulos de libros:

4. Pelegri-Sebastia J.; **Lajara, R.**; and Alberola, J. (2009). Solar Power Source for Autonomous Sensors, Renewable Energy, T J Hammons (Ed.), ISBN: 978-953-7619-52-7, InTech, DOI: 10.5772/7361.

Congresos:

5. **Lajara, R.**, Perez, J.J., Pelegri, J. (2014, April). Method for measuring internal resistance of batteries in WSN. *IEEE Sensors, 2014 (accepted)*.
6. Sogorb, T., Vicente Llario, J., Pelegri, J., **Lajara, R.**, & Alberola, J. (2008, May). Studying the feasibility of energy harvesting from broadcast RF station for WSN. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE* (pp. 1360-1363). IEEE.

7. Alberola, J., Pelegri, J., **Lajara, R.**, & Perez, J. J. (2008, May). Solar inexhaustible power source for wireless sensor node. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*(pp. 657-662). IEEE.
8. **Lajara, R.**, Alberola, J., Pelegri, J., Sogorb, T., & Vicente Llario, J. (2007, October). Ultra low power wireless weather station. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on* (pp. 469-474). IEEE.

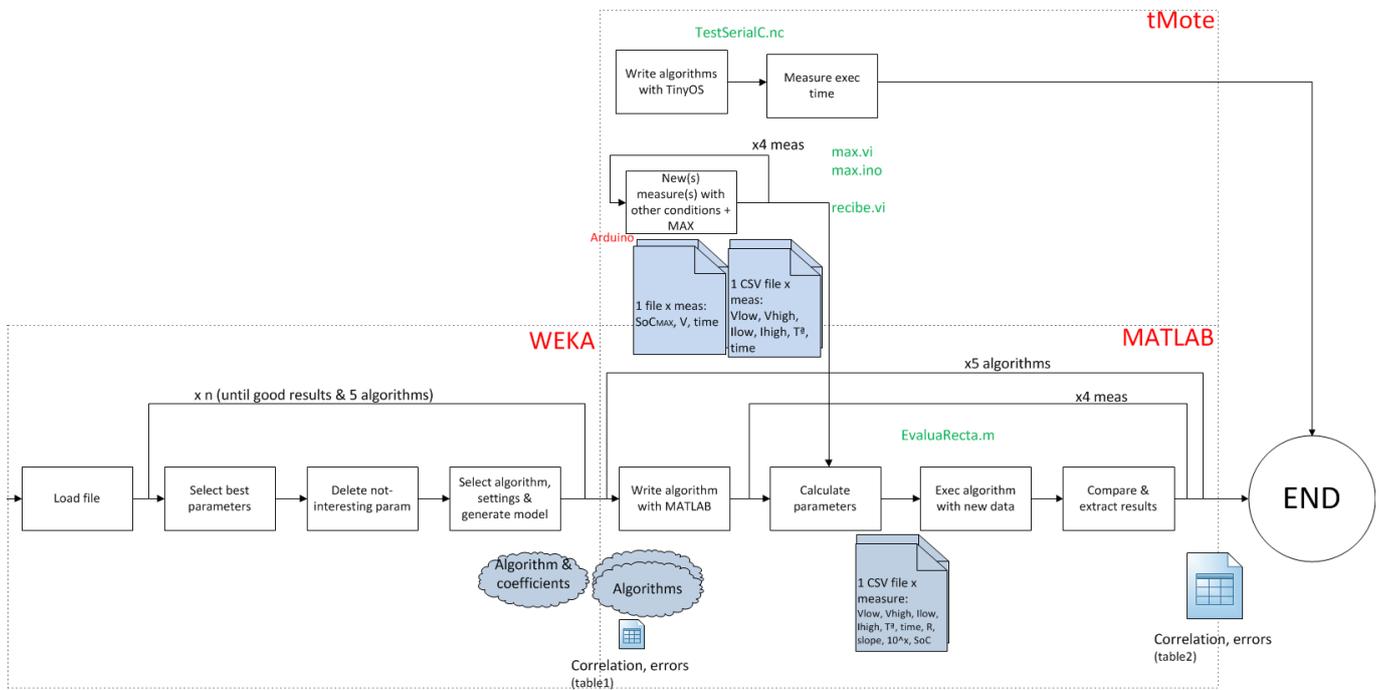
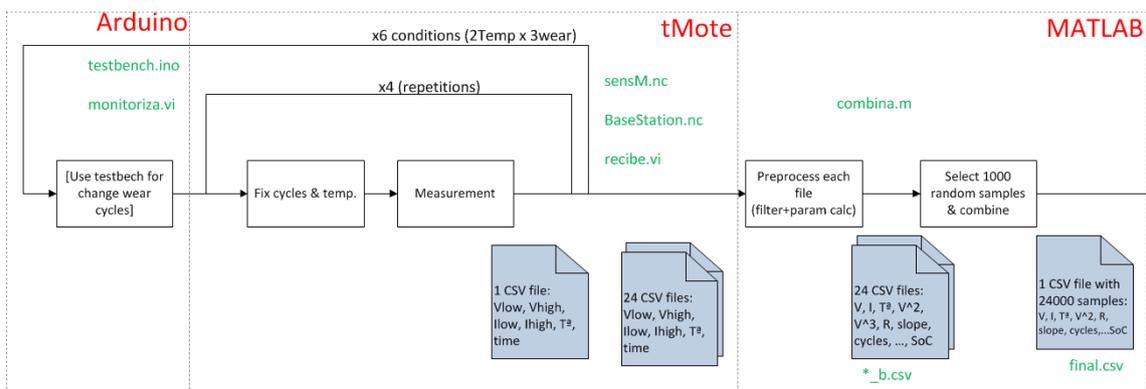
Proyectos de investigación:

1. Desarrollo de un sistema basado en una red de sensores con tecnología inalámbrica (PAID-06-06-002-61).
2. Utilización de energías renovables en redes de sensores inalámbricos (GVPRE/2008/348).
3. Redes de sensores en sistemas distribuidos: estudio, análisis y desarrollo (GV05/043)



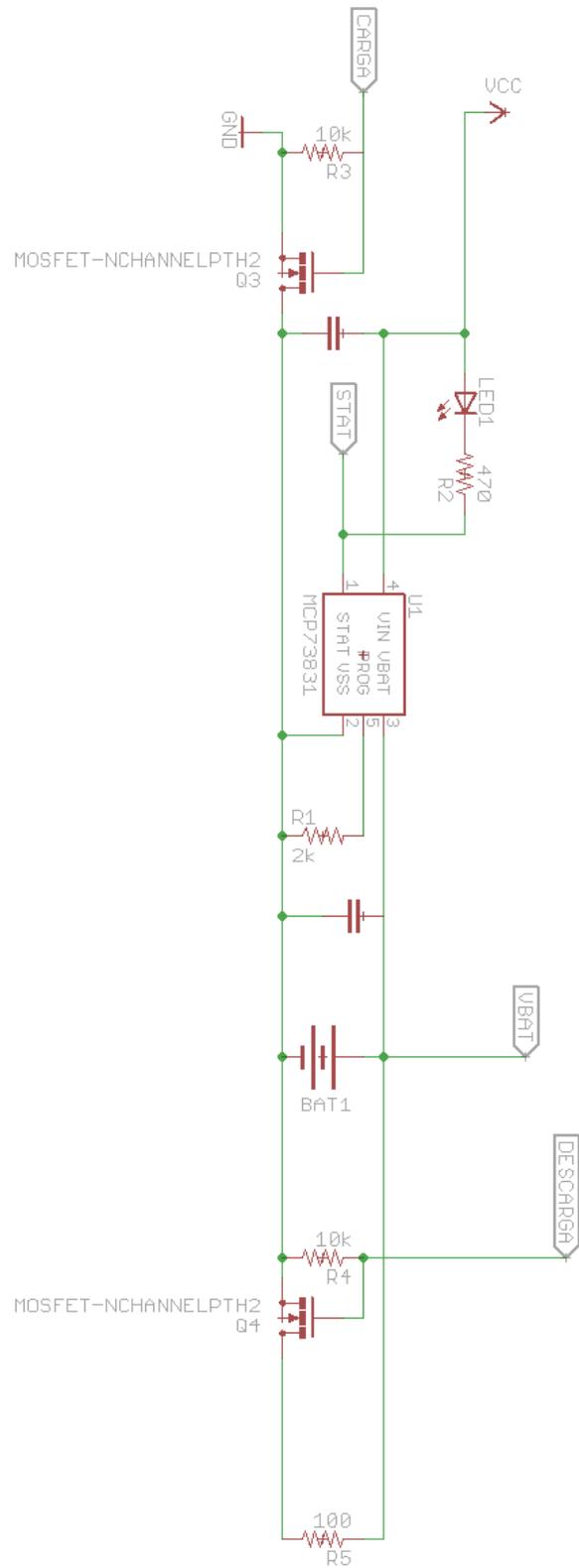
# 10. ANEXOS

## 10.1 Diagrama de operaciones para implementar la metodología de modelado

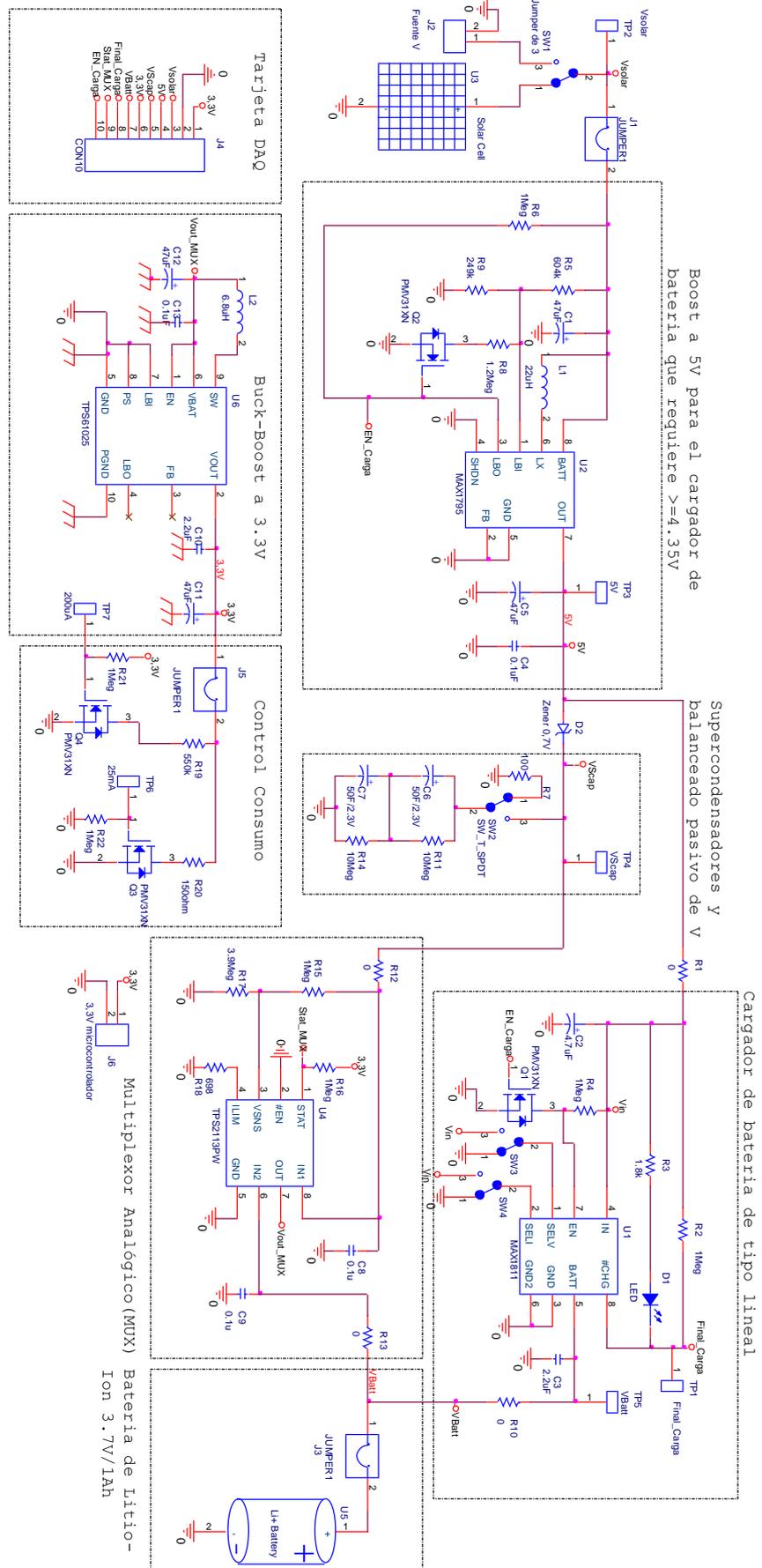


## 10.2 Circuitos

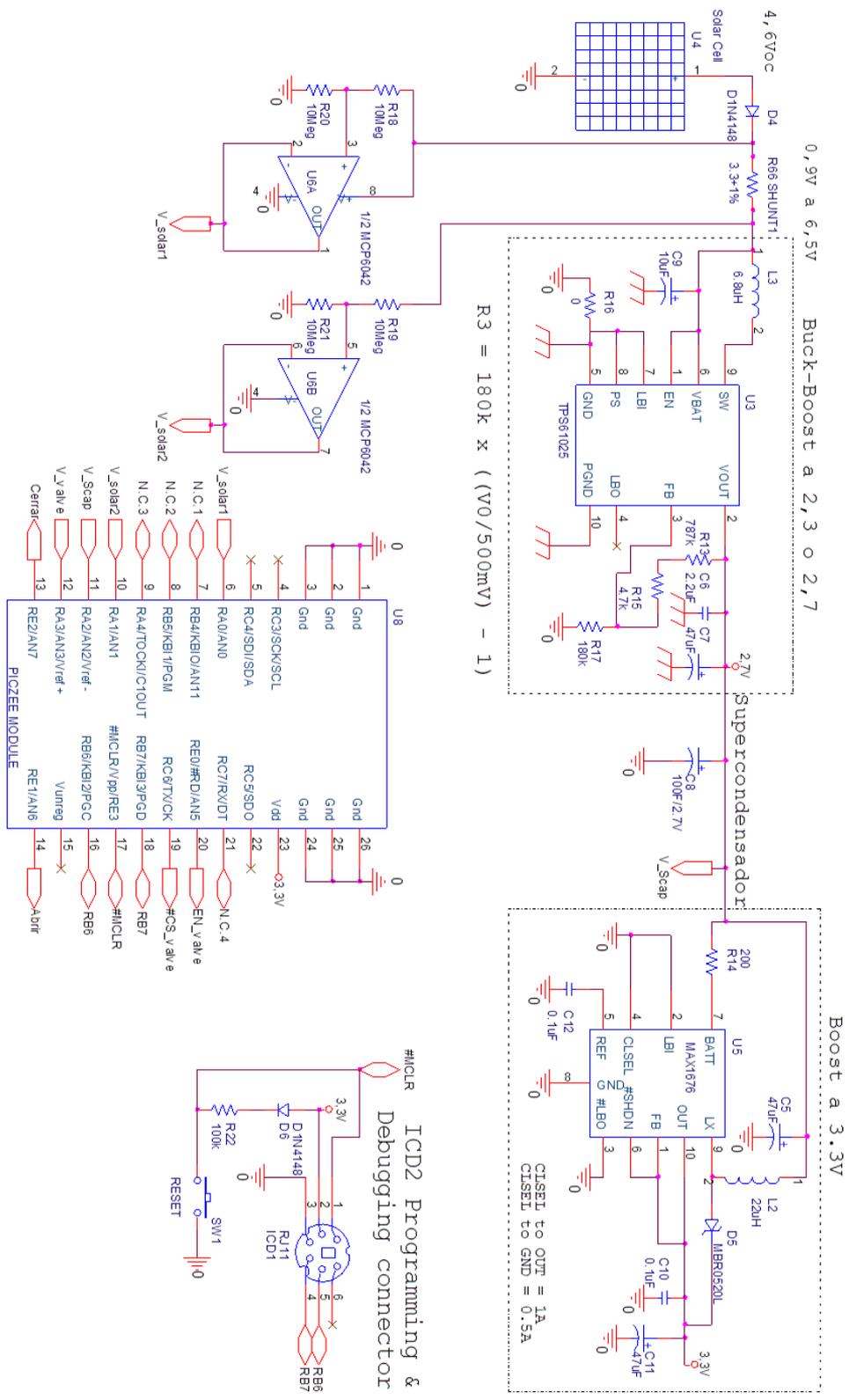
### 10.2.1 Circuito del banco de desgaste de baterías



## 10.2.2 Circuito de la fuente de alimentación

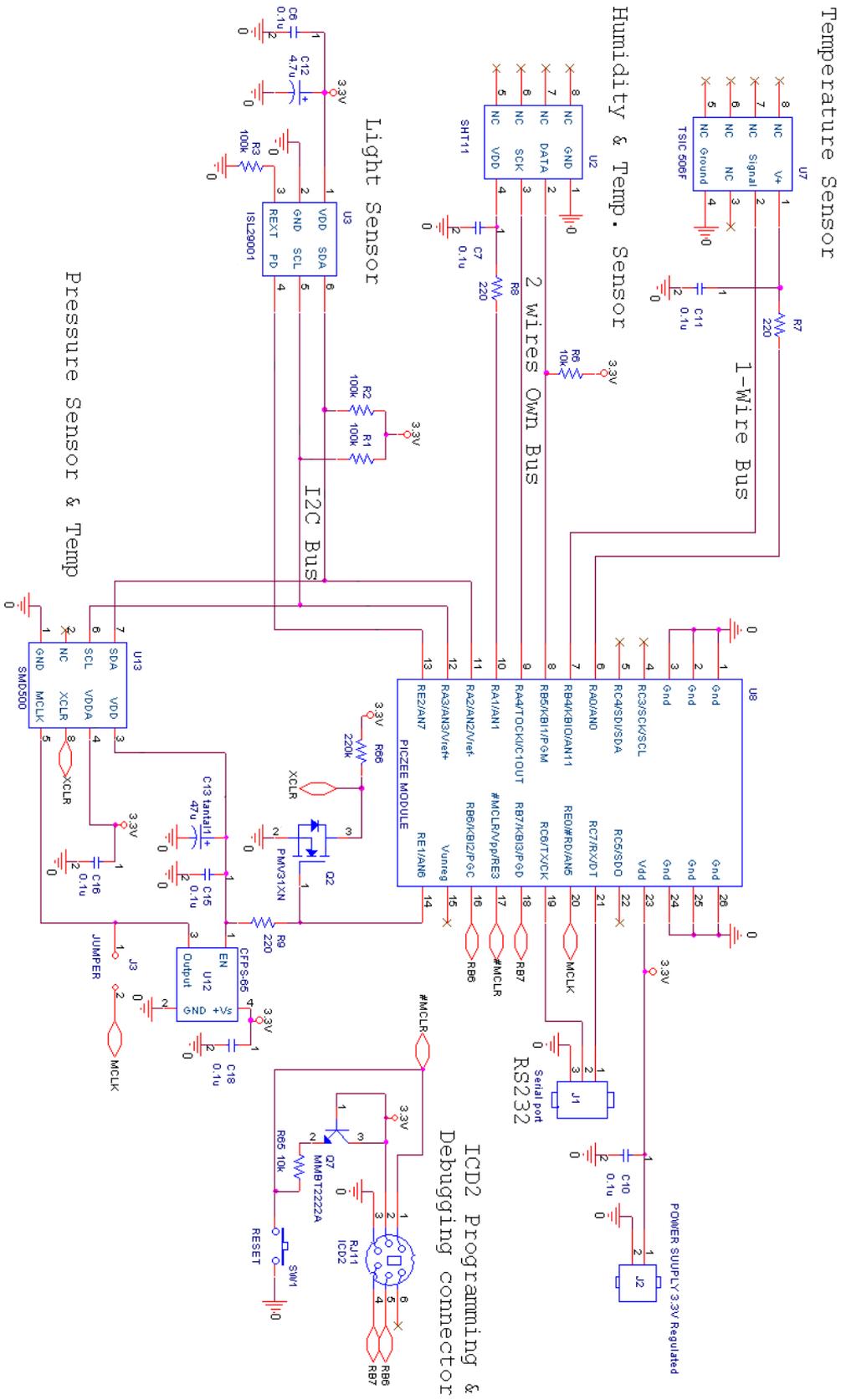


### 10.2.3 Circuito del nodo de irrigación





## 10.2.4 Circuito de la estación meteorológica



### 10.3 Tablas de resultados de los modelos sobre trazas

Sesión	Modelo	Correlación	Error absoluto (SoC)	Error cuadrático (SoC)	Error abs relativo (%)	Error cuadrático relativo (%)
212 ciclos 5°C	Eq 2	0,7499	32,38	35,22	208,56	185,99
	Eq 3 Reg1	0,7499	11,18	13,65	72,04	72,07
	Eq 4 segm	0,989	5,13	5,88	33,05	31,05
	Eq 5 Reg3	0,969	6,75	7,94	43,49	41,94
	Eq6 MLP 1	0,998	3,17	3,52	20,44	18,61
	Eq7 MLP 5	0,986	1,29	3,26	8,34	17,20
213 ciclos 5°C	Eq 2	0,8347	28,36	34,24	64,20	117,07
	Eq 3 Reg1	0,8347	16,81	18,88	42,24	64,57
	Eq 4 segm	0,9912	11,06	12,54	43,33	42,87
	Eq 5 Reg3	0,9555	11,34	12,87	33,12	44,00
	Eq6 MLP 1	0,9989	8,67	10,01	33,12	34,21
	Eq7 MLP 5	0,995	5,92	6,94	22,60	23,74
10 ciclos 5°C	Eq 2	0,83498	29,913	35,8884	116,5871	121,8989
	Eq 3 Reg1	0,83498	16,3317	18,6812	63,6536	63,45
	Eq 4 segm	0,99267	5,8306	6,8075	22,7248	23,12
	Eq 5 Reg3	0,96324	9,3367	10,6294	36,3899	36,10
	Eq6 MLP 1	0,99873	3,2924	4,236	12,8323	14,39
	Eq7 MLP 5	0,9955	3,975	5,4022	15,4927	18,35
11 ciclos 5°C	Eq 2	0,90538	30,592	36,2508	129,0401	131,2261
	Eq 3 Reg1	0,90538	15,3545	17,7693	64,7669	64,32
	Eq 4 segm	0,99491	5,9953	6,5769	25,2888	23,81
	Eq 5 Reg3	0,97885	7,4694	10,0772	31,5067	36,78
	Eq6 MLP 1	0,9862	7,3327	8,3024	30,9299	30,05
	Eq7 MLP 5	0,99258	4,9001	5,6417	20,669	20,42
4 ciclos 25°C	Eq 2	0,8703	30,6186	35,3151	127,8939	127,8246
	Eq 3 Reg1	0,8703	13,5134	15,8549	56,4455	57,39
	Eq 4 segm	0,98556	3,9339	5,25	16,4321	19,00
	Eq 5 Reg3	0,98113	5,3924	6,3734	22,524	23,07
	Eq6 MLP 1	0,98593	4,0693	4,9598	16,9973	17,95
	Eq7 MLP 5	0,99277	4,2269	5,2223	17,6559	18,90
5 ciclos 5°C	Eq 2	0,81914	34,9237	38,7388	189,7984	172,5964
	Eq 3 Reg1	0,81914	14,149	16,8292	76,895	74,98
	Eq 4 segm	0,99113	4,7377	5,5436	25,7477	24,70
	Eq 5 Reg3	0,97364	8,359	10,2464	45,4282	45,65
	Eq6 MLP 1	0,99025	7,4016	7,6992	40,225	34,30
	Eq7 MLP 5	0,99322	1,6017	2,9171	8,7048	13,00
8 ciclos 25°C	Eq 2	0,80664	28,8323	34,2869	127,0669	130,0086
	Eq 3 Reg1	0,80664	15,1519	17,5321	66,7762	66,48
	Eq 4 segm	0,99208	3,7963	4,5523	16,7307	17,26
	Eq 5 Reg3	0,96066	8,8792	10,2234	39,1315	38,77
	Eq6 MLP 1	0,99609	2,9062	3,5251	12,8081	13,37
	Eq7 MLP 5	0,9964	1,8091	2,6688	7,973	10,12
9 ciclos 5°C	Eq 2	0,82702	36,7293	40,3665	172,3028	157,3198
	Eq 3 Reg1	0,82702	14,9184	17,0936	69,9842	66,62
	Eq 4 segm	0,98867	3,8531	4,8103	18,0754	18,75

## 10. Anexos

	Eq 5 Reg3	0,95509	8,0512	9,188	37,7695	35,81
	Eq6 MLP 1	0,99782	1,647	2,6058	7,7262	10,16
	Eq7 MLP 5	0,99574	2,0162	3,0901	9,4582	12,04
1 ciclos	Eq 2	0,88057	32,3627	36,9918	139,7384	138,10
25°C	Eq 3 Reg1	0,88057	14,2342	16,6526	61,46	62,17
	Eq 4 segm	0,98732	9,2467	11,0656	39,92	41,31
	Eq 5 Reg3	0,95913	7,7158	9,7859	33,32	36,53
	Eq6 MLP 1	0,99202	8,1909	8,7529	35,37	32,68
	Eq7 MLP 5	0,99367	4,724	5,5838	20,40	20,85
4 ciclos	Eq 2	0,86222	28,8256	34,7154	127,8415	133,1864
25°C	Eq 3 Reg1	0,86222	16,0892	18,2779	71,3557	70,12
	Eq 4 segm	0,9898	5,3517	6,1146	23,7349	23,46
	Eq 5 Reg3	0,96678	10,7751	11,9338	47,7874	45,78
	Eq6 MLP 1	0,99884	3,5768	3,9664	15,8632	15,22
	Eq7 MLP 5	0,99641	2,6423	3,392	11,7185	13,01
40 ciclos	Eq 2	0,80341	29,2418	34,7583	132,2788	135,89
25°C	Eq 3 Reg1	0,80341	15,5537	17,7808	70,36	69,51
	Eq 4 segm	0,99224	3,7079	4,1302	16,77	16,15
	Eq 5 Reg3	0,97191	8,6732	9,6726	39,23	37,82
	Eq6 MLP 1	0,99411	3,6944	4,1928	16,71	16,39
	Eq7 MLP 5	0,99805	5,0999	5,4072	23,07	21,14
41 ciclos	Eq 2	0,85911	30,1962	35,3757	139,3379	141,049
25°C	Eq 3 Reg1	0,85911	14,8816	17,1231	68,67	68,27
	Eq 4 segm	0,99363	5,5605	6,2297	25,6584	24,84
	Eq 5 Reg3	0,97548	6,7187	8,6538	31,003	34,50
	Eq6 MLP 1	0,99266	7,3849	7,9515	34,0772	31,70
	Eq7 MLP 5	0,99746	1,3406	2,043	6,1859	8,15
44 ciclos	Eq 2	0,80269	32,1755	36,2815	145,7633	142,9424
5°C	Eq 3 Reg1	0,80269	13,3518	15,8703	60,4872	62,53
	Eq 4 segm	0,99394	5,2395	5,8499	23,7362	23,05
	Eq 5 Reg3	0,94831	7,884	9,2205	35,7165	36,33
	Eq6 MLP 1	0,99829	2,7151	3,1254	12,3002	12,31
	Eq7 MLP 5	0,99554	1,9814	2,6735	8,9761	10,53
45 ciclos	Eq 2	0,78202	28,0559	33,7728	127,8658	133,2756
25°C	Eq 3 Reg1	0,78202	15,7805	18,1291	71,9203	71,54
	Eq 4 segm	0,99125	4,7677	5,6985	21,7289	22,49
	Eq 5 Reg3	0,96606	9,5267	10,7261	43,4184	42,33
	Eq6 MLP 1	0,99875	2,7913	3,7247	12,7214	14,70
	Eq7 MLP 5	0,99503	3,1554	4,108	14,3808	16,21
49 ciclos	Eq 2	0,82177	23,6857	29,2443	109,0633	114,4945
5°C	Eq 3 Reg1	0,82177	15,5033	17,9865	71,3867	70,42
	Eq 4 segm	0,98914	10,7631	11,9631	49,5596	46,84
	Eq 5 Reg3	0,95826	10,876	12,5653	50,0797	49,20
	Eq6 MLP 1	0,99924	8,323	9,162	38,3241	35,87
	Eq7 MLP 5	0,99499	5,2065	6,2571	23,9739	24,50
50 ciclos	Eq 2	0,82596	27,656	32,223	124,9641	124,7596
5°C	Eq 3 Reg1	0,82596	13,3707	15,5517	60,4157	60,21
	Eq 4 segm	0,98762	8,6328	10,0879	39,0073	39,06
	Eq 5 Reg3	0,94893	9,072	10,6774	40,992	41,34
	Eq6 MLP 1	0,99738	6,1664	7,2641	27,8627	28,13

Modelado y optimización de energía en WSN para la medida de parámetros medioambientales

	Eq7 MLP 5	0,9909	3,129	4,4947	14,1386	17,40
53 ciclos 5°C	Eq 2	0,88963	31,0573	35,944	126,0391	127,564
	Eq 3 Reg1	0,88963	14,1271	16,1991	57,3316	57,49
	Eq 4 segm	0,99618	2,9039	3,4355	11,7849	12,19
	Eq 5 Reg3	0,97783	5,7384	8,1119	23,2882	28,79
	Eq6 MLP 1	0,9946	3,1621	3,773	12,8327	13,39
	Eq7 MLP 5	0,99289	4,1793	5,0814	16,9606	18,03
54 ciclos 25°C	Eq 2	0,86567	32,1944	37,13	144,8263	144,4693
	Eq 3 Reg1	0,86567	15,0389	17,4968	67,6526	68,08
	Eq 4 segm	0,99424	9,9848	10,3562	44,9164	40,30
	Eq 5 Reg3	0,9773	8,503	10,5648	38,2509	41,11
	Eq6 MLP 1	0,98592	10,4025	11,2371	46,7957	43,72
	Eq7 MLP 5	0,99623	4,9926	5,5847	22,4593	21,73
210 ciclos 25°C	Eq 2	0,78557	29,3425	34,5903	135,5701	137,9573
	Eq 3 Reg1	0,78557	14,8749	17,1635	68,7259	68,45
	Eq 4 segm	0,99128	3,094	4,361	14,2951	17,39
	Eq 5 Reg3	0,9692	6,9422	8,2157	32,0747	32,77
	Eq6 MLP 1	0,99332	4,9693	5,4544	22,9594	21,75
	Eq7 MLP 5	0,99541	1,787	2,8047	8,2564	11,19
211 ciclos 25°C	Eq 2	0,81896	28,459	33,8348	133,8035	137,2262
	Eq 3 Reg1	0,81896	14,9787	17,1541	70,424	69,57
	Eq 4 segm	0,99132	3,3598	3,9632	15,7966	16,08
	Eq 5 Reg3	0,96999	8,1885	9,2015	38,4992	37,32
	Eq6 MLP 1	0,99645	3,6457	4,1858	17,1406	16,98
	Eq7 MLP 5	0,9932	4,415	5,1639	20,7578	20,94
216 ciclos 25°C	Eq 2	0,8629	30,9087	36,2088	139,032	140,8073
	Eq 3 Reg1	0,8629	15,3269	17,6066	68,9426	68,47
	Eq 4 segm	0,99343	4,4309	5,429	19,9307	21,11
	Eq 5 Reg3	0,98092	7,2641	8,8177	32,6752	34,29
	Eq6 MLP 1	0,98942	6,6559	7,2217	29,9394	28,08
	Eq7 MLP 5	0,99503	1,2355	2,5628	5,5574	9,97
217 ciclos 5°C	Eq 2	0,83067	23,7358	29,6517	113,6669	118,1459
	Eq 3 Reg1	0,83067	15,5858	17,788	74,6377	70,88
	Eq 4 segm	0,99427	6,7739	7,6967	32,4392	30,67
	Eq 5 Reg3	0,97471	9,2967	10,2578	44,5204	40,87
	Eq6 MLP 1	0,99575	4,6739	5,7798	22,3824	23,03
	Eq7 MLP 5	0,98156	3,0516	4,9287	14,6135	19,64
222 ciclos 5°C	Eq 2	0,83161	30,1628	36,0231	119,4584	124,79
	Eq 3 Reg1	0,83161	15,8285	18,2553	62,69	63,21
	Eq 4 segm	0,99558	3,2547	3,8086	12,89	13,19
	Eq 5 Reg3	0,96583	8,029	10,0028	31,80	34,65
	Eq6 MLP 1	0,99341	3,8865	4,3736	15,39	15,15
	Eq7 MLP 5	0,98704	4,39	5,2705	17,38	18,26
223 ciclos 25°C	Eq 2	0,84218	31,7722	36,5918	145,7025	145,0591
	Eq 3 Reg1	0,84218	14,7505	17,1652	67,6437	68,05
	Eq 4 segm	0,99424	10,1757	10,5008	46,6643	41,63
	Eq 5 Reg3	0,98105	8,9736	10,6382	41,1518	42,17
	Eq6 MLP 1	0,98184	10,5815	11,4976	48,5249	45,78
	Eq7 MLP 5	0,99226	5,5748	6,4282	25,5649	25,48

## 10.4 Otras publicaciones no relacionadas con la tesis

Otros artículos, libros y proyectos de investigación y divulgación\* se enumeran a continuación:

Revistas:

1. Pelegrí Sebastiá, J., Alberola Lluch, J., & **Lajara Vizcaíno, J. R.** (2007). Signal conditioning for GMR magnetic sensors: applied to traffic speed monitoring GMR sensors. *Sensors and Actuators A: Physical*, 137(2), 230-235.
2. \* Pareja, M., **Lajara, J. R.**, & Pelegrí, J. (2012). Distribución Linux para electrónicos. *Revista española de electrónica*, (686), 44-48.

Congresos:

3. **Lajara, J. R.**, Pérez, E., Pelegrí-Sebastiá, J., & Solano, J. J. P. (2011, October). An integrated graphical development environment for the implementation of WSN. In *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on* (pp. 179-181). IEEE.
4. Sebastiá, J. P., Alberola Lluch, J., **Lajara Vizcaino, J. R.**, & Santiso Bellon, J. (2009). Vibration detector based on GMR sensors. *Instrumentation and Measurement, IEEE Transactions on*, 58(3), 707-712.
5. Sebastia, J. P., **Vizcaino, J. R. L.**, & Orti, M. T. (2004, December). Specific protocol (SENSLOG) over TCP/IP for sensors applications. In *Industrial Technology, 2004. IEEE ICIT'04. 2004 IEEE International Conference on* (Vol. 2, pp. 741-744). IEEE.

Libros:

1. \* **Vizcaíno, J. R. L.**, & Sebastiá, J. P. (2013). *Sistemas integrados con Arduino*. Marcombo.
2. \* **Vizcaíno, J. R. L.**, Llario, V. & Sebastiá, J. P. (2013). *Diseño de circuitos impresos con EAGLE*. Marcombo
3. \* **Vizcaíno, J. R. L.**, & Sebastiá, J. P. (2011). *LabVIEW: entorno gráfico de programación*. 2ª edición. Marcombo.
4. \* **Vizcaíno, J. R. L.**, & Sebastiá, J. P. (2007). *LabVIEW: entorno gráfico de programación*. Marcombo.

Proyectos final de carrera dirigidos:

1. Fuster, B. (2007). Diseño e implementación de un cliente/servidor DHCP para sistemas embebidos. Ingeniería electrónica, Universidad de Valencia.
2. Pérez, E. (2009). Implementación de redes de sensores mediante LabVIEW. Ingeniería técnica de telecomunicación, Universidad Politécnica de Valencia.

Participación en proyectos de investigación y desarrollo:

1. ASTUTE. Improve usability of embedded systems by using user state and context situation capture to provide pro-active decision support via multi-modal interfaces (ART Call 2010 269334).
2. INTEGNEO. Investigación y desarrollo de una plataforma para el seguimiento continuo de marcadores fisiológicos en recién nacidos asintomáticos. Papel en la predicción de cardiopatías congénitas, infección y muerte (IPT-2011-0824-900000).
3. SI4TE. Desarrollo de nuevos sistemas de generación eléctrica basados en la termoelectricidad para el aprovechamiento del calor residual generado por las personas, máquinas o aparatos eléctricos/electrónicos (TSI-020100-2011-0354).
4. NEWTON. Nueva generación de técnicas ópticas de transmisión OFDM para futuras redes WDM-PONS (TEC2011-26642).