TinyDomus

Ramón Beltrán Sánchez
“Ingeniería *Técnica Informática de Sistemas”*

Jordi Bécares Ferrés

8 de junio de 2011

Resumen

El proyecto realizado durante el TFC Sistemas Empotrados ha consistido, básicamente, en el desarrollo de un software para gestionar el despliegue de una red inalámbrica mediante el uso de dos dispositivos autónomos denominados motas. Cada módulo dispone de un interfaz USB para conectarlas a un equipo de escritorio, tres Leds con los que podremos interactuar, y una serie de sensores que nos permitirán monitorizar ciertos aspectos ambientales y físicos.

La funcionalidad que nuestro proyecto, TinyDomus, pretende conseguir es la de simular un hogar domótico. En función de los datos de temperatura y luminosidad, se activarán dos sensores emulando el comportamiento de un sistema de aire acondicionado o regulando el encendido de los sistemas de iluminación en la habitación donde nos encontremos. Por su parte, el sensor de efecto Hall, sería el responsable de gestionar el sistema de seguridad.

La parte de interfaz gráfica ha sido diseñada desde cero e integra todas estas funcionalidades. Desde dicho interfaz el usuario dispone de las funcionalidades necesarias para realizar las siguientes tareas:

- Visualizar las lecturas que la mota remota hace de los sensores
- Gestionar la frecuencia con que el dispositivo remoto las realiza de forma autónoma
- Hacer peticiones a la mota remota, de forma interactiva, sobre el estado de los sensores de temperatura y luminosidad
- Detener la lectura de los sensores por parte del nodo remoto
- Consultar a la mota remota su configuración actual
- Modificar y enviarle al sistema remoto los valores mínimo y máximo a partir de los que deberá alterar el estado del Led de temperatura para simular la puesta en funcionamiento del sistema de calefacción y aire acondicionado
- Cambiar el valor mínimo a partir del cual, el dispositivo mota remota enciende el Led asociado al fotosensor, regulando la intensidad del mismo en función al grado de oscuridad (a mayor oscuridad más intensidad)

Índice de contenidos

ÍNDICE DE TABLAS 5
1. INTRODUCCIÓN 6
1.1. JUSTIFICACIÓN 7
1.2. DESCRIPCIÓN DEL PROYECTO 9
1.3. OBJETIVOS DEL TFC 10
1.4. ENFOQUE 11
1.5. PLANIFICACIÓN 13
1.6. RECURSOS EMPLEADOS 14
1.7. PRODUCTOS OBTENIDOS 16
1.8. BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA 17
2. ANTECEDENTES 18
2.1. ESTADO DEL ARTE 18
2.2. ESTUDIO DE MERCADO 20
3. DESCRIPCIÓN FUNCIONAL 25
3.1. SISTEMA TOTAL 25
3.2. PC 25
3.3. MOTA 26
4. DESCRIPCIÓN DETALLADA 28
5. VIABILIDAD TÉCNICA 35
6. VALORACIÓN ECONÓMICA 37
7. CONCLUSIONES 39
7.1. CONCLUSIONES 39
7.2. PROPUESTA DE MEJORAS 40
7.3. AUTOEVALUACIÓN 40
8. GLOSARIO 42
9. BIBLIOGRAFÍA 44
10. ANEXOS 45
10.1. MANUAL DE USUARIO 45
10.2. EJECUCIÓN Y COMPILACIÓN 51
10.3. SERIALFORWARDER 53

Índice de figuras

FIGURA 1. ESQUEMA EJEMPLO DE SISTEMA DOMÓTICO 12
FIGURA 2. PARTES DE UNA MOTA COU_1_1 15
FIGURA 3. COMPONENTES FUNCIONALES Y ESTRUCTURA DE RED TINYDOMUS 27
FIGURA 4. EJEMPLO DE COMUNICACIÓN ENTRE PC Y NODO REMOTO 30
FIGURA 5. DIAGRAMA GESTIÓN SENSOR DE LUMINOSIDAD 32
FIGURA 6. DIAGRAMA GESTIÓN SENSOR DE TEMPERATURA 33
FIGURA 7. DIAGRAMA GESTIÓN SENSOR DE EFECTO HALL 34
FIGURA 8. ASPECTO PANEL DE CONTROL TINYDOMUS 46

Índice de tablas

TABLA 1. CUADRO COMPARATIVO FECHAS PREVISTAS VS FECHAS REALES DE ENTREGA 13
TABLA 2. PREVISIÓN CRECIMIENTO MERCADO WSN 21
TABLA 3. PREVISIÓN POR ÁREAS GEOGRÁFICAS 22

1. Introducción

Durante el desarrollo de nuestro TFC, hemos ido implementando un sistema de comunicaciones formado por una red de dispositivos inalámbricos de bajo consumo, en adelante motas, dotados de sensores que nos permiten obtener información, tanto ambiental como física, del entorno que nos rodea. Concretamente los sensores que disponen nuestro juego de motas, nos proporcionan información acerca de la temperatura, el índice de luminosidad ambiental, la detección de variaciones magnéticas (sensor efecto Hall), y la posibilidad de obtener el voltaje de las baterías que las alimentan.

Como complemento al software creado para nuestras motas, se ha desarrollado completamente una aplicación de PC para mostrar al usuario la información obtenida de los sensores, y por otra parte, permitirle interactuar con el sistema de motas, modificando ciertos aspectos del funcionamiento de nuestra red de sensores inalámbricos (WSN).

A la fase de instalación y configuración del entorno de desarrollo, ha seguido la de formación, desarrollo y testeo de nuestro software.

El alcance principal de este TFC es la realización práctica de todas las etapas que se requiere efectuar para llevar a cabo un proyecto y conseguir un resultado que cumpla con los requerimientos indicados durante las fases iniciales:

- Análisis de los requisitos del TFC y elaboración de una Propuesta Inicial

- A partir de la propuesta inicial, se procede a la elaboración de un Plan de Trabajo que haga una descripción detallada de nuestro proyecto:

    - Descripción de los objetivos y de las tareas necesarias para llevarlos a cabo
    - Requisitos funcionales que deberá cumplir
    - Requisitos técnicos necesarios para llevarlos a cabo
    - Planificación: Es una parte fundamente del Plan de Trabajo y es donde se indican claramente los puntos clave de nuestro proyecto y su organización temporal

- Desarrollo del software: Es la parte principal del proyecto y debe estar íntimamente ligada a los requerimientos especificados en el punto anterior

- Elaboración de la Memoria y Presentación Virtual del TFC

1.1. Justificación

Las Redes de Sensores Inalámbricas (WSN), facilitan el despliegue de una red descentralizada de sensores autónomos que pueden trabajar cooperativamente para monitorizar diferentes tipos de magnitudes físicas o ambientales. Su utilidad abarca un amplio abanico de posibilidades que pueden ir desde la medición de factores ambientales en diferentes localizaciones, el control de procesos industriales (a través del funcionamiento de sensores y actuadores, el control de tareas y la supervisión de los procesos), o la monitorización de ciertos parámetros físicos en medicina. La función principal de este tipo de redes es la de obtener información relevante a través de los sensores que puede ser recogida, interpretada y almacenada.

Las características intrínsecas a este tipo de redes, las hacen perfectamente compatibles a las necesidades de nuestro proyecto:

- Facilidad de instalación dado que no requieren de una infraestructura de cableado de red o suministro eléctrico
- Por su tamaño, pueden adaptarse fácilmente a cualquier entorno incluso aquellos lugares de difícil acceso
- En caso de avería, son fácilmente reemplazables
- Disponen de sensores que se adaptan perfectamente a nuestras necesidades
- Los avances tecnológicos de la plataforma, dejan la puerta abierta a nuevas funcionalidades

Cada nodo de red, dispone de una serie de sensores, un emisor/receptor (transceptor) de radio que permite la comunicación entre cada uno de los nodos y un microcontrolador (uC), que les proporciona memoria y capacidad de procesar información. La fuente de energía suele ser normalmente una pila aunque, en algunos casos, pueden ir dotados de sistemas de captación de energía (como fotocélulas o conversores termoeléctricos que incrementan la vida de estos dispositivos y minimizan su mantenimiento). La posibilidad de equipar a los nodos de diferentes tipos de sensores (como por ejemplo, detectores de fuego o seísmos, sensores para aplicaciones industriales, sensores ambientales de temperatura, humedad o iluminación, dispositivos con aplicaciones médicas, como, electrocardiogramas o encefalogramas, sensores de movimiento para el control de tráfico, etc.), hace que las WSN tengan un número ilimitado de aplicaciones.

Otros aspectos fundamentales son su facilidad de implantación, permitiendo incluso, acceder a lugares inaccesibles para redes cableadas, la posibilidad de que los nodos puedan estar operativos durante mucho tiempo sin necesidad de reemplazar su fuente de alimentación (redes desatendidas), y por otra parte, su versatilidad, permitiendo modificar dinámicamente las aplicaciones que ejecutan para adaptarlas a nuestras necesidades. También es importante destacar que los rápidos avances tecnológicos que se están produciendo en esta disciplina y que está dirigida a la fabricación de dispositivos menos costosos, de dimensiones más reducidas y, por otra parte, más eficientes energéticamente. Todos y cada uno de los apartados detallados anteriormente justifican la elección de este TFC. Prueba de ello, es el interés que despierta en las grandes empresas tecnológicas.

El TFC desarrollado trata de aplicar esta tecnología al mundo de la domótica mejorando las condiciones de habitabilidad de los hogares: Control optimizado de temperatura, variaciones de la intensidad de luz en función del grado de luminosidad y gestión de sistemas de seguridad. Estos son los puntos centrales de nuestro proyecto y que tienen como finalidad el uso del potencial que nos ofrece el hardware de que disponemos.

Haciendo una breve descripción de las funcionalidades de cada dispositivo, diremos que uno de estos módulos tendrá el rol de dispositivo remoto. Será el responsable de gestionar y responder a las peticiones que le llegan desde la estación base, obtener información de los diferentes sensores, modificar el estado de los Leds en función de las lecturas realizadas y, enviar dicha información a la mota base que, a su vez, las reenviará al equipo de sobremesa a través del puerto USB. Para el ordenador que hace las veces de sistema de control, se ha desarrollado un completo interfaz de usuario que mostrará la información recibida y que, además, permitirá gestionar el funcionamiento de la mota remota mediante el envío de mensajes a través de la estación base. Esta última, será la encargada de gestionar todas las comunicaciones en ambos sentidos, bien, convirtiendo en mensajes de radio, las peticiones que nuestra aplicación de escritorio le envía a través del puerto USB, o viceversa.

Para la programación de los microcontroladores que gestionan las motas, en adelante uC, hemos necesitado de un sistema operativo que nos permita abstraernos, en la medida de lo posible, de la complejidad del hardware, de su modelo de ejecución, y de la gestión de procesos que realiza. En nuestro caso se ha optado por la instalación, sobre un entorno Linux (Ubuntu 10.04.2 Lucid Lynx), de un sistema operativo desarrollado por la Universidad de Berkeley, TinyOS, expresamente dedicado a la gestión de redes inalámbricas de dispositivos. TinyOS dispone de una serie de librerías y un compilador, nesC, que nos permitirán llevar a cabo la programación de nuestras motas.

Para poder adaptar TinyOS a las características del uC y el sistema de radio (Atmel ATZB-24-A2), que hemos utilizado, ha sido necesario el uso de un software desarrollado por el Instituto Tecnológico de

Zurich (ETHZ), que adapta la estructura de árbol de TinyOS, con las funcionalidades de nuestro hardware.

La aplicación de escritorio que hace las veces de interfaz de usuario y sistema de control de las operaciones de lectura y gestión de Leds, ha sido diseñada completamente en Java. Dicho software hace uso de las herramientas Java disponibles en TinyOS y del software de gestión de mensajes, MainTester.java, diseñado previamente.

Para el desarrollo de este proyecto han sido fundamentales las aportaciones de Tutoría, de los compañeros del Foro y, por su puesto, de toda la documentación disponible en Internet.

1.2. Descripción del proyecto

Los requisitos funcionales del TFC se dividen en tres grandes bloques:

- Aplicación de escritorio: Incluye el desarrollo de la interfaz gráfica de usuario y la gestión de mensajes de entrada/salida desde y hacia la mota remota. Sus funcionalidades son estas:
    - Mostrar en pantalla las lecturas realizadas por el nodo remoto. Además de mostrar los valores recibidos desde el nodo remoto, simula el comportamiento de cada uno de los Leds, es decir:
        - Para cuadro de texto asociado al Led de temperatura, el color blanco del fondo cambiará a amarillo si su valor se encuentra por debajo del Mínimo. En caso de superar el Máximo, el color del fondo irá cambiando de amarillo a blanco simulando un efecto de parpadeo
        - El cuadro de texto asociado al Led de luminosidad, cambiará de blanco a rojo en caso de que el valor de lectura se encuentre por debajo del Mínimo establecido
        - En caso del sensor de efecto Hall, el cuadro de texto cambiará de blanco a verde cada vez que dicho sensor detecte un pico magnético
    - Controlar la frecuencia con que se realizan dichas lecturas. El usuario puede modificar a voluntad la frecuencia de muestreo. Dicho periodo puede indicarse tanto en segundos o minutos. Así mismo, dicha interfaz, permite detener temporalmente las lecturas que la mota remota obtiene de los sensores

  - Modificar el valor Mínimo que regula el funcionamiento del Led asociado al sensor de luminosidad
  - Regular los valores Mínimo/Máximo del sensor de temperatura. Dichos valores pueden ser introducidos tanto en grados Celsius como en grado Fahrenheit
  - Consultar los parámetros de configuración almacenados en la mota remota. Una vez recibida la respuesta del nodo, dichos valores serán mostrados en pantalla

- Mota Base. Gestor de comunicaciones entre la mota remota y el sistema de control:

  - Los mensajes de radio recibidos desde la mota remota son redirigido al PC a través del puerto USB
  - Por el contrario, los mensajes enviados desde la aplicación de escritorio al puerto USB son reenviados a la mota remota en forma de mensajes de radio

- Mota Remota. Juega un papel fundamental dentro de nuestro proyecto y es el responsable de las siguientes funcionalidades:

  - Realizar las lecturas de cada uno de los sensores y modificar el estado del Led asociado
  - Enviar la información de las lecturas, mediante mensajes de radio, a la mota base
  - Atender a todas las peticiones que llegan desde la mota base, a saber:

    - Consulta o modificación de los valores de configuración: temporizador, valores mínimo/máximo de los sensores de temperatura y luminosidad
    - Petición de lectura de los sensores de temperatura y luminosidad. Estas lecturas se realizan al margen de la frecuencia de muestreo programada
    - Detención del muestreo automático de los sensores

1.3. Objetivos del TFC

En líneas generales, los objetivos que se han perseguido durante el TFC han sido los de llevar a cabo el ciclo completo de desarrollo de un proyecto, pasando por cada una de sus fases:

- Analizar los objetivos propuestos para realizar un borrador inicial que cumpla con los objetivos planteados

- A partir de la propuesta consensuada con el tutor, realizar un plan de trabajo donde se detallen los requisitos funcionales y técnicos que tendrá nuestro proyecto
- Elaborar una planificación, lo más realista posible, de todas y cada una de las tareas a realizar
- Configurar un entorno de trabajo que nos permite programar e interactuar con nuestras motas
- Adquirir los conocimientos necesarios sobre TinyOS, su lenguaje de programación, nesC, y el hardware de nuestras motas
- Implementar el software que cumpla con los todas las funcionalidades definidas para cada una de las motas (tanto para la mota que cumple con el rol de mota base como para la mota remota)
- Diseñar un interfaz de usuario totalmente integrada con la red de sensores inalámbricos
- Elaborar una memoria que recoja, de una manera detallada, todos y cada uno de los aspectos del trabajo realizado y los resultados obtenidos
- Realizar una presentación virtual que describa de una manera clara y concisa nuestro proyecto

1.4. Enfoque

El enfoque que se ha pretendido dar a nuestro proyecto ha sido el de aplicar la tecnología que nos ofrecen las WSN a la automatización y control de determinados sistemas eléctricos (como puede ser el control de iluminación), y electrónicos (sistemas de aire acondicionado, alarmas), dentro del hogar. Es decir, la visión del proyecto está claramente orientada a la Domótica. Citando textualmente [3]:

"El término Domótica proviene de la unión de las palabras domus (que significa casa en latín), y tica (de automática, palabra en griego, 'que funciona por sí sola'). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado"

En definitiva, la finalidad del Proyecto TinyDomus no es otra que la de automatizar la gestión de los sistemas de iluminación y climatización del hogar y, por otra parte, establecer los mecanismos para activar un posible sistema de alertar para la detección de intrusos. Todos estos mecanismos son gestionados por el usuario a través de un interfaz gráfico diseñado a tal efecto.

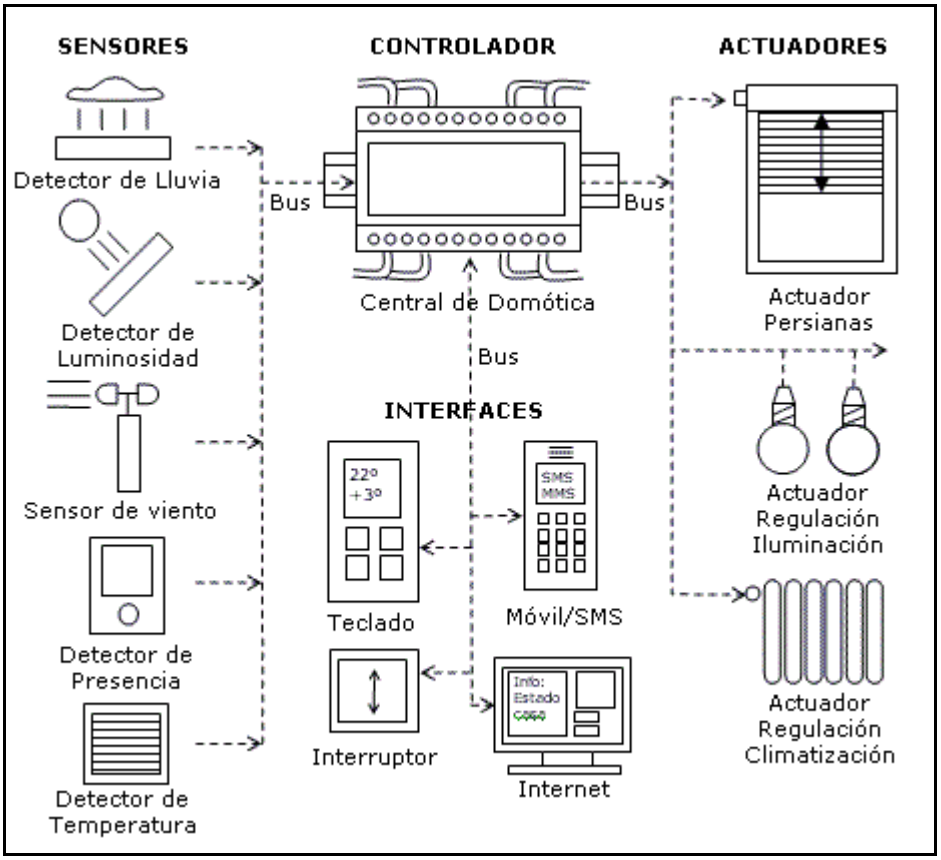

Figura 1. Esquema ejemplo de SISTEMA DOMÓTICO

Este esquema es un claro ejemplo de un entorno domótico en el hogar. Este diseño va más allá de las funcionalidades de nuestro proyecto. Además de los sensores de temperatura, luminosidad y detector de presencia que se incluyen en nuestra solución, el esquema añade detectores de lluvia y sensores de viento. Además, ofrece diferentes interfaces de usuario: un teclado a modo de mando de control (sería el equivalente a nuestra interfaz de usuario), la recepción de información directamente al móvil y la interacción con el sistema a través de Internet. Esta última opción, fue una de las posibles mejoras que se plantearon durante el TFC.

1.5. Planificación

En los primeros estadios del TFC se elaboró un Plan de Trabajo que incluía una planificación temporal de los hitos del proyecto. Cada uno de los puntos clave se subdividía en tareas que se acotaron en el tiempo. Por la experiencia obtenida durante la elaboración del proyecto, la delimitación temporal de las tareas ha resultado de gran importancia para la finalización de cada una de las etapas dentro de los límites de entrega exigidos. En el siguiente diagrama, comparamos las fechas previstas de entrega y las reales:

TAREA	COMIENZO PLANIFICADO	FIN PLANIFICADO	ENTREGA REAL
Fase 1: Plan de Trabajo	**02/03/2011**	**25/03/2011**	24/03/2011
• **Análisis, elaboración y entrega Propuesta TFC (rbeltrans_propuesta.pdf)**	02/03/2011	11/03/2011	
• **Elaboración Plan de Trabajo (Descripción objetivos TFC, tareas de las que se compone, calendario, recursos, etc.)**	12/03/2011	24/03/2011	
• **Revisión y entrega final documento (rbeltrans_plan_de_trabajo.pdf)**	25/03/2011	25/0311	
Fase 2: Implementación (1/2)	**25/03/2011**	**26/04/2011**	**25/04/2011**
• **Desarrollo parcial código de nuestra aplicación (interfaz gráfica y comunicación PC-mota)**	25/03/2011	24/04/2011	
• **Revisión y entrega de documentos**	25/04/2011	26/04/2011	
Fase 3: Implementación (2/2)	**26/04/2011**	**31/05/2011**	**23/05/2011**
• **Desarrollo final del proyecto (comunicación mota-PC, análisis y procesamiento de los datos recogidos –muestra en pantalla de datos y gestión Leds)**	26/04/2011	29/05/2011	
• **Revisión y entrega Proyecto**	30/05/2011	31/05/2011	
Memoria	**17/04/2011**	**10/06/2011**	**08/06/2011**
• **Desarrollo Memoria**	17/04/2011	08/06/2011	
• **Revisión y entrega Memoria**	09/06/2011	10/06/2011	
Presentación Virtual	**10/06/2011**	**17/06/2011**	**----------**
• **Creación Presentación Virtual**	10/06/12011	16/06/2011	
• **Revisión y entrega Presentación Virtual**	17/06/12011	17/06/2011	

Tabla 1. Cuadro comparativo Fechas Previstas vs Fechas Reales de entrega

Como se observa en el cuadro adjunto las dos primeras fases cumplen casi con exactitud las fechas planificadas. Respecto a las Fases 3 y 4, hay una variación temporal debido a problemas ajenos al proyecto (viajes por motivos de trabajo), que han obligado a replantear el desarrollo del trabajo. Debido a la necesidad de reorganizar el tiempo disponible para el TFC, se optó por dar prioridad a la segunda entrega del código (Fase 3), y no solaparlo con la Memoria, que se comenzó a desarrollar una vez realizada esta entrega.

1.6. Recursos empleados

Los requisitos fundamentales para la realización del proyecto han sido, por un lado, el hardware necesario para poder desplegar una red inalámbrica y, por supuesto, un entorno de desarrollo completamente operativo. No podemos olvidar la fase de formación sin la cual, hubiera resultado imposible la realización del TFC.

El hardware que utilizaremos para desplegar nuestra WSN se compone de dos motas COU_1_1 en las que encontramos los siguientes componentes:

- Un núcleo Atmel ATZB-900-B0 compuesto por un uC atmega 1281 y el dispositivo de radio ZigBee at86rf212 RF Transceiver
- Sensores temperatura, luminosidad, picos magnéticos (efecto Hall), y voltaje de la batería
- Chip MAC para redes inalámbricas
- Conversor UART a USB
- Botón RST para reiniciar la mota
- Botón USR programable

La siguiente imagen muestra la distribución física, en la placa de circuito impreso, de cada uno de los componentes descritos. En la parte trasera de la mota, se ubican las dos pilas del tipo triple A que se utilizan como fuente de energía.

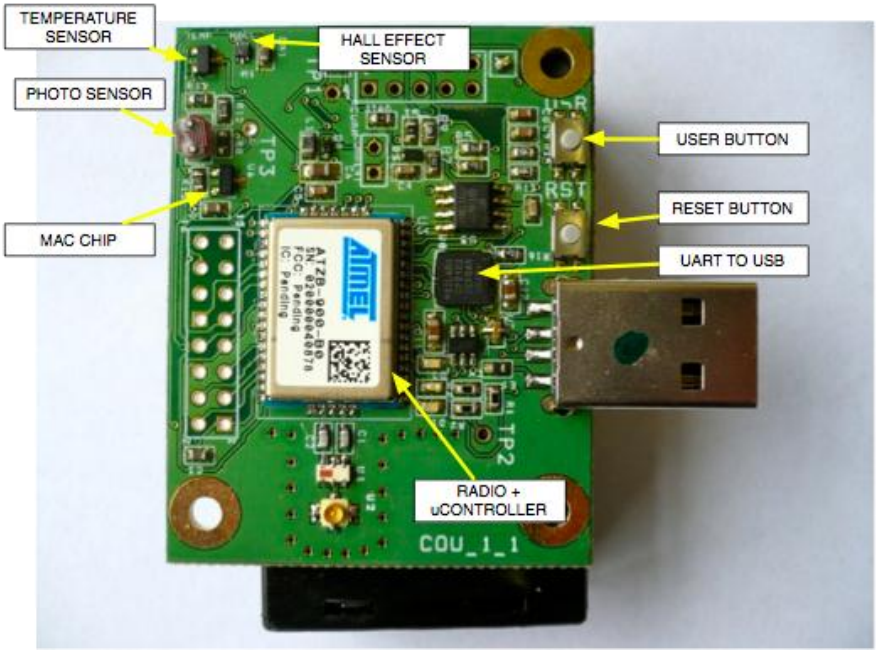

Figura 2. Partes de una Mota COU_1_1

Una condición necesaria para llevar a cabo el TFC, ha sido la configuración de un entorno que nos permitiera desarrollar un software compatible con el hardware Atmel que compone el núcleo central de la plataforma COU24 que utilizan nuestras motas. En nuestro caso, y una vez analizadas las diferentes alternativas propuestas, se ha optado por la configuración de una partición Linux (Ubuntu 10.04.2 Lucid Lynx), sobre un portátil MacBook Pro. En dicha partición se han configurado todas las herramientas necesarias para el desarrollo, instalación y puesta en funcionamiento de nuestro proyecto:

- Java 1.6 JDK
- Compiladores nativos (Atmel AVT Tools)
- Sistema operativo TinyOS (toolchain y source tree)
- Modificación del árbol de TinyOS para hacerlo compatible con nuestro hardware (software desarrollado por el ETHZ)
- Meshprog (desarrollado por Meshnetics). Permite programar las motas con nuestro software
- Instalación del IDE Eclipse Ganymede

- Instalación del plugin Yeti2 (ETHZ), para integrar TinyOS sobre Eclipse

- Instalación Graphviz (software open source para poder visualizar gráficamente la estructura de nuestro software)

- Finalmente, y como parte fundamental de nuestro desarrollo, descarga del software de ejemplo, a destacar:

    - BaseStation: Su función es la de realizan de transceptor entre la mota remota y el PC al que se conecta
    - blinkCou: Contiene todo el software necesario para poder hacer lecturas de todos los sensores y programar el botón USR de la mota
    - COUTester: La clase java MainTester, reúne todas las funcionalidades necesarias para poder recibir y enviar mensajes desde y hacia las motas

1.7. Productos obtenidos

A las fases de análisis, elaboración, entrega y aprobación de la propuesta de TFC, han seguido las fases de diseño e implementación del código. Como resultado de todo el trabajo realizado se ha obtenido un producto de software formado por las siguientes aplicaciones:

Para definir claramente nuestros objetivos, procederemos a detallar los tres módulos claramente diferenciados que componen nuestro proyecto:

- **BaseDomus**: Es el software que se instala en la mota base (desarrollada en nesC). Sus funcionalidades son:

    - Gestionar las comunicaciones entre la mota remota y el PC en ambos sentidos. BaseDomus reutiliza el software de BaseStation
    - Implemente el componente WatchDogC para resetear la mota en caso de que deje de estar operativa
    - Incluye el módulo BotonUser que gestiona el botón USR de la mota base. La utilidad de dicho botón es la de detener el muestreo en la mota remota

- **RemoteDomus**: Este módulo se instala en la mota remota (nesC). Sus elementos fundamentales son:

    - Parametros: Gestiona las peticiones para consultar/cambiar los valores de configuración de los sensores de luz (MIN), y temperatura (MIN/MAX)
    - GestorSensors: Lee la información de los sensores de luz y temperatura periódicamente y los envía por radio (mensajes AM_SENSORSMSG). También gestiona los mensajes recibidos de la mota base (AM_TIMERMSG), para parar/arrancar el temporizador o consultar/modificar su valor en milisegundos
    - EfectoHall: Es el responsable de gestionar el sensor de efecto Hall y del tratamiento de la mensajería asociada a dicho sensor (AM_HALLMSG)
    - GestorLeds: Gestiona el estado de los Leds. Es utilizado por los módulos EfectoHall y GestorSensors para modificar el estado de los Leds

- **TinyDomus**: Aplicación de escritorio desarrollada completamente en Java y que se divide en los siguiente componentes:

    - ControlRemoto: Define la interfaz de usuario
    - TinyDomus: Desarrollado a partir de MainTester, es la responsable de procesar todos los mensajes que llegan desde la mota remota. Utiliza ControlRemoto para gestionar la interfaz gráfica de usuario e implementa toda la funcionalidad que esta ofrece

1.8. Breve descripción de los otros capítulos de la memoria

En los siguientes apartados se procederá a realizar un breve análisis de la situación actual de las redes de sensores inalámbricas así como de su implantación en el mercado. Para ello citaremos algunos casos reales donde esta tecnología se ha implantado con éxito. Seguidamente realizaremos una descripción funcional de los distintos aspectos del proyecto. También trataremos de detallar su viabilidad técnica y su posible valoración económica. Este aparatado irá seguido de las conclusiones obtenidas, propuesta de mejora y autoevaluación del trabajo realizado.

Como punto final de la memoria se incluye un glosario de términos que aparecen en la memoria, bibliografía consultada y otros anexos como son, un manual de usuario, los pasos necesarios para su compilación y ejecución y, finalmente, un apartado dedicado a SerialForwarder. En este apartado enteremos las posibilidades que ofrece este software y que se adaptan perfectamente a los requerimientos de nuestro proyecto.

2. Antecedentes

2.1. Estado del arte

En las Redes de Sensores Inalámbricas convergen las tecnologías de computación, comunicaciones y sensores para cambiar nuestra forma de vida, la manera de interactuar con el medio y, por supuesto, la manera de llevar a cabo los negocios. Sólo tenemos que buscar en Internet para darnos cuenta de la gran cantidad de aplicaciones y proyectos que tiene esta tecnología.
Este tipo de redes se basan en el trabajo colaborativo de una serie de nodos de sensores (desde unos pocos hasta miles de ellos), normalmente denominados motas, capaces de comunicarse entre ellos para enviar la información registrada a un punto establecido de recogida de información. Las principales características de estos nodos son:

- Bajo costo de implantación y mantenimiento
- Capacidad de obtener determinada información del medio, procesarla y comunicarla a través de la red
- Bajo consumo de energía
- Multifunción: dada su versatilidad de uso al poder equiparse con una amplia gama de sensores (temperatura, humedad, luminosidad, vibración, temperatura del aire, concentración de gases, acelerómetros, sensores de carga, de inclinación, picos magnéticos, voltajes, etc.)

De estas características también se pueden deducir algunas de las principales limitaciones que tienen en la actualidad:

- Necesidad de mejorar el grado de precisión de los sensores soportados
- Mejoras en los tiempos de respuesta: algunos procesos industriales requieren que los datos obtenidos por los sensores sean enviados inmediatamente para la toma de decisiones acordes con las lecturas recibidas. En estos casos la demora en el tiempo puede ser decisiva
- Como consecuencia del punto anterior, necesidad de desarrollar protocolos especiales para aplicaciones de tiempo real

- Gestión energética de las motas: Ciclo de vida limitado. Nuevos protocolos que optimicen el consumo energético de los nodos que componen las redes WSN sin pérdida de la calidad del servicio

- Nuevas tecnologías que aumente el ciclo de vida de las motas: incorporación células fotosensibles, captación de energía a través de cambios térmicos

- Robustez y escalabilidad de las redes

- Mejora las herramientas de programación: Independencia entre la programación y los detalles a bajo-nivel de las diferentes configuraciones hardware de las motas, los sensores y las comunicaciones nodo a nodo

- Mejorar la cooperación, conocimiento y adaptación al medio

- Soporte de seguridad: protección frente a ataques de intrusos que puede sufrir la red. Especialmente si se trata de aplicaciones críticas que requieren de un gran nivel de fiabilidad

No obstante, a pesar de lo que parece al leer el apartado anterior, el panorama actual de la tecnología de redes inalámbricas es mucho más que optimista. Sólo hay que navegar por la red en busca de proyectos relacionados con redes de sensores para darse cuenta de la gran cantidad de aplicaciones reales que ya las utilizan y de la infinidad de proyectos que existen en la actualidad.

Como ejemplo de aplicaciones reales de las redes de sensores inalámbricas, podemos encontrar los siguientes:

- Agricultura de precisión con redes de sensores inalámbricas: Permiten gestionar parámetros ambientales como temperatura, humedad, radiación solar y ajustar el uso de fertilizantes y fitosanitarios a las necesidades de cada cultivo permitiendo realizar un control automatizado de plagas. Esta plataforma puede ser accesible a través de internet y permite gestionar las alertas a través de mensajes SMS o correo electrónico. Además, el control automatizado permite optimizar la gestión de la explotación agrícola y ahorrar en costes, además de mejorar la cantidad y la cantidad de la producción

- Domótica e Inmótica (básicamente domótica a gran escala, ver glosario): Obtienen medidas de consumo de agua, electricidad, etc. para optimizar su consumo. También se utiliza en alarmas y seguridad y, como en nuestra práctica aunque a mayor escala, crean entornos que se adaptan

dinámicamente a las actividades de los usuarios (p.e. gestión de los sistemas de climatización, encendido/apagado de iluminación, apertura/cierre de persianas)

- Medicina: Monitorización de los parámetros vitales de los pacientes. Supervisión de ancianos y personas con necesidades especiales. Esto implica un seguimiento 24x7 (24 horas, 7 días a la semana), de los pacientes sin necesidad de acudir al centro de salud

- Monitorización de procesos industriales: Toma de datos en puntos de difícil acceso y control automatizado de los procesos. Prevención de riesgos laborales y eficiencia en el uso de recursos

- Control de infraestructuras: Como distribuciones eléctricas o detección de pérdidas en canalizaciones y tuberías de agua. Monitorización de estructuras (p.e. control de grietas, vibraciones o de la velocidad del vientos en puentes)

- Monitorización de tráfico: Con el fin de aumentar la seguridad vial, y en particular advertir el conductor de cualquier tipo de evento que pueda pasar en su trayecto, como congestión de tráfico, accidentes o datos meteorológicos que dañen el estado de la carretera. Lo que redunda en menos accidentes que permitan salvar vidas humanas. También podrían permitir reducir la contaminación mediante el control del tráfico en determinadas zonas

Cada una de estas aplicaciones prácticas es un indicador del nivel de implantación que, en la actualidad, están adquiriendo las tecnologías de redes inalámbricas de sensores. Aunque, como veremos en el siguiente apartado, las posibilidades de futuro que ofrece pronostican un mayor nivel de implantación.

2.2. Estudio de mercado

El punto anterior nos ha permitido conocer la situación actual de las WSN, pero ¿cuáles son las previsiones de futuro?, ¿es una tecnología que ha llegado para quedarse o simplemente se trata de una tendencia pasajera? Como veremos en el siguiente análisis, los proyectos de investigación que se están desarrollando están centrados en buscar soluciones reales a sus limitaciones actuales y, por otro lado, a incrementar sus funcionalidades.

No en vano, las redes de sensores inalámbricas han sido consideradas, según el MIT, como una de las diez tecnologías emergentes de este siglo. Según ABIresearch, empresa americana de estudios de

mercados tecnológicos, el sector de las WSN creció durante 2010 un 300% sobre el año anterior y, lo que es más importante, sus previsiones para los próximos cinco años son absolutamente optimistas. Esto nos da una idea del crecimiento que va a sufrir esta tecnología durante los próximos años.

Si bien, durante 2009 hubo un retroceso con respecto al año anterior, ABIresearch indica que fue debido a que, al tratarse de una tecnología emergente, ciertos proyectos piloto que trataron de adoptar este tipo de soluciones en una fase quizás demasiado temprana, fueron pospuestos en espera de su evolución. La creación de estándares tecnológicos (citando algunos ejemplos, la alianza ZigBee, protocolos tales 802.15.4, IP/6LoPAN, Z-Wave, Wireless HART, ISA100 o EnOcean), está dejando de lado el uso soluciones propietarias y potenciando el uso de las WSN. Según Dolcera, firma internacional especializada en propiedad intelectual, investigación de mercados y estrategias de competitividad para Norte América, Europa y Asia, también señala unas estimaciones muy favorable para el mercado de las redes de sensores y transmisores durante los próximo años. Tal y como podemos comprobar en los gráficos siguientes, este mercado va a experimentar un importante crecimiento.

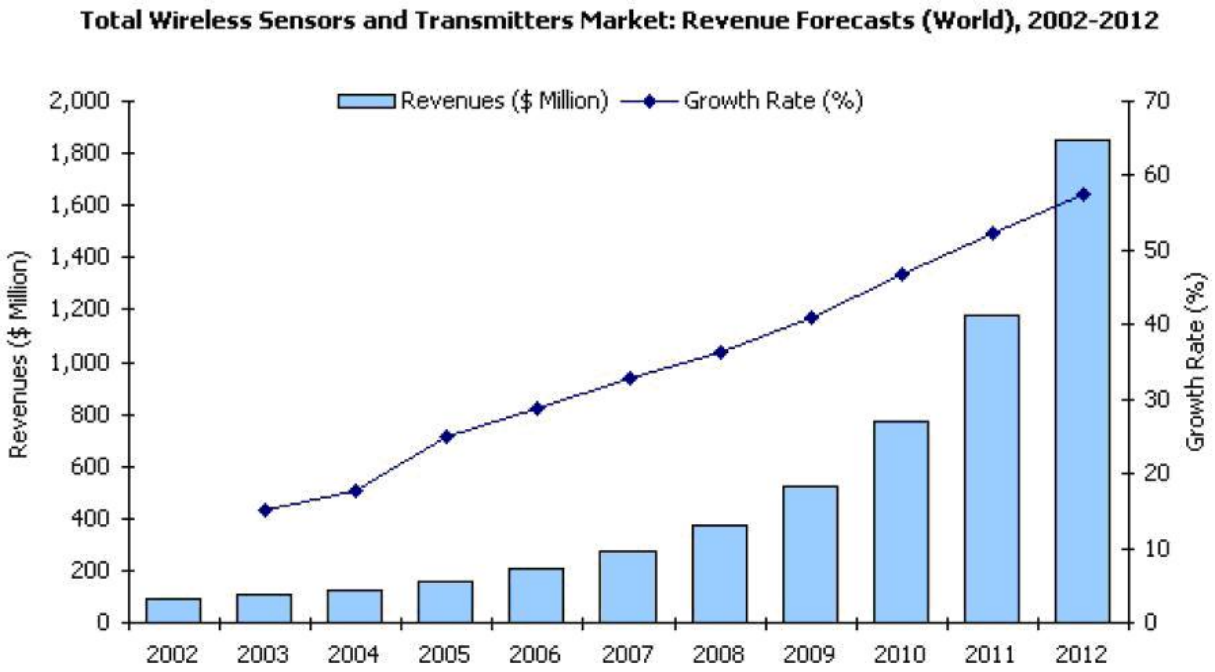

Tabla 2. Previsión crecimiento mercado WSN

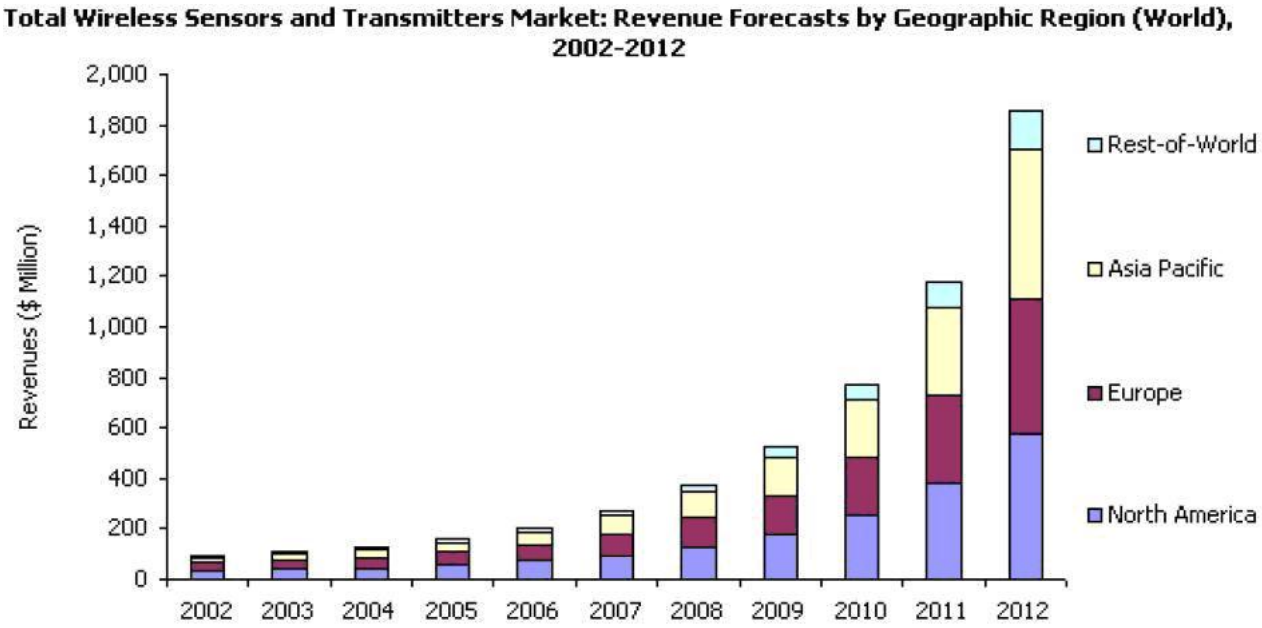

Tabla 3. Previsión por áreas geográficas

Así mismo la industria de las WSN está demostrando una clara orientación hacia el concepto de USN (Ubiquitous Sensors Networks). La filosofía de las USN es la tendencia a la interconexión entre personas y objetos, a cualquier hora y en cualquier lugar. Si las previsiones de futuro se cumplen, estaríamos hablando de un nivel de conectividad globalizado con una potencia de proceso difícilmente imaginable incluso en nuestros días: las USN proporcionarán nuevos servicios y aplicaciones que mejorarán la seguridad del entorno que nos rodea y, por consiguiente, nuestra calidad de vida.

Otro de los apartados fundamentales de las WSN en los que la industria está prestando especial atención es el diseño de nodos que tengan el mínimo consumo energético o, en el mejor de los casos, puedan ser autosuficientes: motas inalámbricas capaces de auto abastecerse sin necesidad de mantenimiento y otras técnicas que se están investigando se orientan a que el mantenimiento de las redes sea lo más sencillo posible. Así mismo, los nuevos dispositivos deben ser cuidadosos con el medio ambiente.

En el campo de la conectividad también se están desarrollando nuevos protocolos que permitan inter conectividad basada en IP, de modo que facilite la interconexión de las WSN con otras redes externas. Como ejemplo del gran interés que está suscitando esta tecnología citaremos algunos de los muchos proyectos orientados al desarrollo de una nueva generación de dispositivos y soluciones inalámbricas. Su finalidad es la de resolver algunas de los problemas actuales que presentan las WSN:

- Genesi (Green Sensor Networks for Structural Monitoring): El objetivo del proyecto GENESI es el desarrollo de las bases que formarán las siguientes generaciones de sistemas de monitorización de estructuras y que deberán cumplir con los requisitos de autonomía energética, durabilidad e integración con el entorno. Esto implica rediseñar los actuales sistemas para superar sus principales deficiencias, principalmente las relativas al almacenamiento y obtención de fuentes de energía respetuosas con el medio ambiente, tolerancia a fallos e interferencias. Así mismo, el proyecto contempla el diseño de nuevo software y protocolos que aprovechen al máximo el nuevo hardware, proporcionando mejoras en la interacción con los distintos componentes y mejorando la calidad del servicio.

- BSAC (Berkeley Sensor & Actuator Center): Se trata de una fundación, compuesta por sectores industriales y universitarios, para la investigación y cooperación en el ámbito de micro sensores y micro actuadores. Esta fundación dirige y agrupa todo tipo de investigaciones asociadas con la tecnología de sensores a micro y nano escala, movimiento de elementos mecánicos, micro fluidos y todos aquellos procesos y sistemas que hacen uso de los progresos realizados en el desarrollo de nuevos materiales para el diseño de circuitos integrados. Muchas son las empresas interesadas y los proyectos en desarrollo.

Obviamente, todos los avances obtenidos a raíz de estas iniciativas, tendrán una consecuencia directa sobre la evolución de las aplicaciones domóticas, que son el tema central de nuestro proyecto. Actualmente, a nivel nacional, podemos encontrar soluciones comerciales y proyectos de investigación orientados al hogar digital que hacen uso de redes inalámbricas de sensores:

- ASIMELEC (Asociación Multisectorial de Empresas de Tecnologías de la Información, Comunicaciones y Electrónica): Dispone de un proyecto, HOGAR DIGITAL, una vivienda real situada en Madrid, donde se exponen todos los productos, servicios y tecnologías (ocio digital, accesibilidad, eficiencia energética y sostenibilidad) y, donde los usuarios puede interactuar con cada producto y servicio. El hogar es accesible a través de Internet y cuenta con capacidad para interactuar en remoto con los dispositivos.

- SAYME: Empresa española que desarrolla soluciones en el ámbito de las nuevas tecnologías de comunicaciones. Su producto, **SENSbee**, es una solución completa de redes de sensores inalámbricos de bajo consumo que permite crear aplicaciones de control y monitorización de variables distribuidas en el entorno. Sus principales características son su tamaño reducido, su fácil instalación, el bajo consumo y las comunicaciones fiables y seguras. Los objetivos principales de la plataforma son:

- Recabar medidas de variables medioambientales y del entorno (incluso en ubicaciones de difícil acceso o lugares remotos)
- Agrupar medidas distribuidas en un punto central a través de comunicaciones inalámbricas en tiempo real
- Enviar señales y alarmas así como activar respuestas automáticas si las variables superan umbrales predeterminados

La solución se basa en el despliegue de una red de nodos inalámbricos que transmiten la información a un nodo central a través de una tecnología denominada SWAM que ha sido desarrollada por SAYME y se basa en el estándar 802.15.4. Además, SENSbee dispone de otros componentes, SENScada y SENSview, proporcionan una solución muy potente para gestionar y visualizar los datos capturados por los sensores. Son accesibles a través de cualquier navegador de Internet, y permiten operar con los datos en tiempo real, así como gestionar informes, configuración de alarmas, o visualización gráfica animada.

Tras analizar el estado de la tecnología y constatar el incremento en la demanda de este tipo de soluciones, sólo podemos concluir que la situación del mercado de redes inalámbricas de sensores y más concretamente, de su aplicación en hogares digitales, es un valor en alza.

3. Descripción funcional

3.1. Sistema total

Nuestro sistema se basa en el uso de la tecnología WSN para la automatización de ciertos procesos, contribuyendo a la eficiencia energética de instalaciones y dispositivos y, por otra parte, gestionando la seguridad en el hogar.

El alcance final del proyecto sería el de proporcionar una interconexión real, mediante el uso de actuadores, entre los resultados obtenidos a través de los sensores, los parámetros establecidos y los distintos dispositivos que tratan de gestionar:

- Control de sistemas de iluminación del hogar
- Regulación automática de los aparatos de climatización
- Gestión de un sistema de detección de intrusos y activación de alarmas

3.2. PC

La funcionalidad de la aplicación de escritorio es la ofrecer al usuario un interfaz gráfico donde monitorizar y configurar cada uno de los aspectos que controlan el comportamiento global del sistema. Desde aquí el usuario tendrá la posibilidad de controlar completamente cada una de las funcionalidades del sistema:

- Permite visualizar las lecturas que la mota remota obtiene de los distintos sensores

- Modificar la apariencia de la ventana principal así como los parámetros de configuración del sistema y, por consiguiente, su funcionamiento

- Consultar, completamente on-line, los distintos mensajes que se reciben y envían a la mota remota

En el anexo 10.1. Manual de usuario, encontraremos información más detallada acerca de la puesta en marcha de la aplicación y de cómo funcionan las distintas opciones de que dispone el interfaz de usuario.

3.3. Mota

En nuestro proyecto, cada una de las motas juega un rol específico y, por lo tanto, dispondrán de un software expresamente diseñado para realizar la labor que tienen asignada. En función del papel que ocupan en el sistema, se diferencian dos tipos de motas:

- Mota Base: Su papel principal es de hacer de enlace entre la mota remota y la aplicación de escritorio. El mal funcionamiento de esta mota hará imposible la comunicación, en cualquier sentido, entre PC y mota remota. Se conecta directamente a uno de los puertos USB del PC.

- Mota Remota: ha de ser capaz de interactuar con los sensores de que dispone, modificar el estado de los Leds, procesar las peticiones recibidas y enviar información al sistema central a través de la mota base. Así pues, dispone de las siguientes funcionalidades:

    - Obtener un muestreo de los sensores periódicamente o cuando se recibe una petición desde el panel de control
    - Gestionar el sensor de efecto Hall. A diferencia del resto de sensores, las lecturas de este dispositivo son completamente aleatorias y sólo las realiza cuando detecta picos magnéticos
    - Enviar toda la información obtenida al sistema central a través de mensajes de radio
    - Atender a todas las peticiones para realizar alguna de las siguientes tareas:
        - Detener el temporizador de lectura de sensores
        - Modificar el tiempo de muestreo
        - Cambiar los valores mínimo y máximo del sensor de temperatura
        - Modificar el valor mínimo para el foto sensor
        - Enviar al sistema central información acerca de los parámetros de configuración

    - Modificar el estado de los Leds (cada uno se asocia a un sensor), en función de los valores obtenidos de los sensores

En el siguiente esquema definiremos las distintas partes que componen el sistema. necesarias para que el sistema sea completamente funcional (para conocer la funcionalidad de SerialForwarder en el diagrama ver anexo 10.3. SerialForwarder):

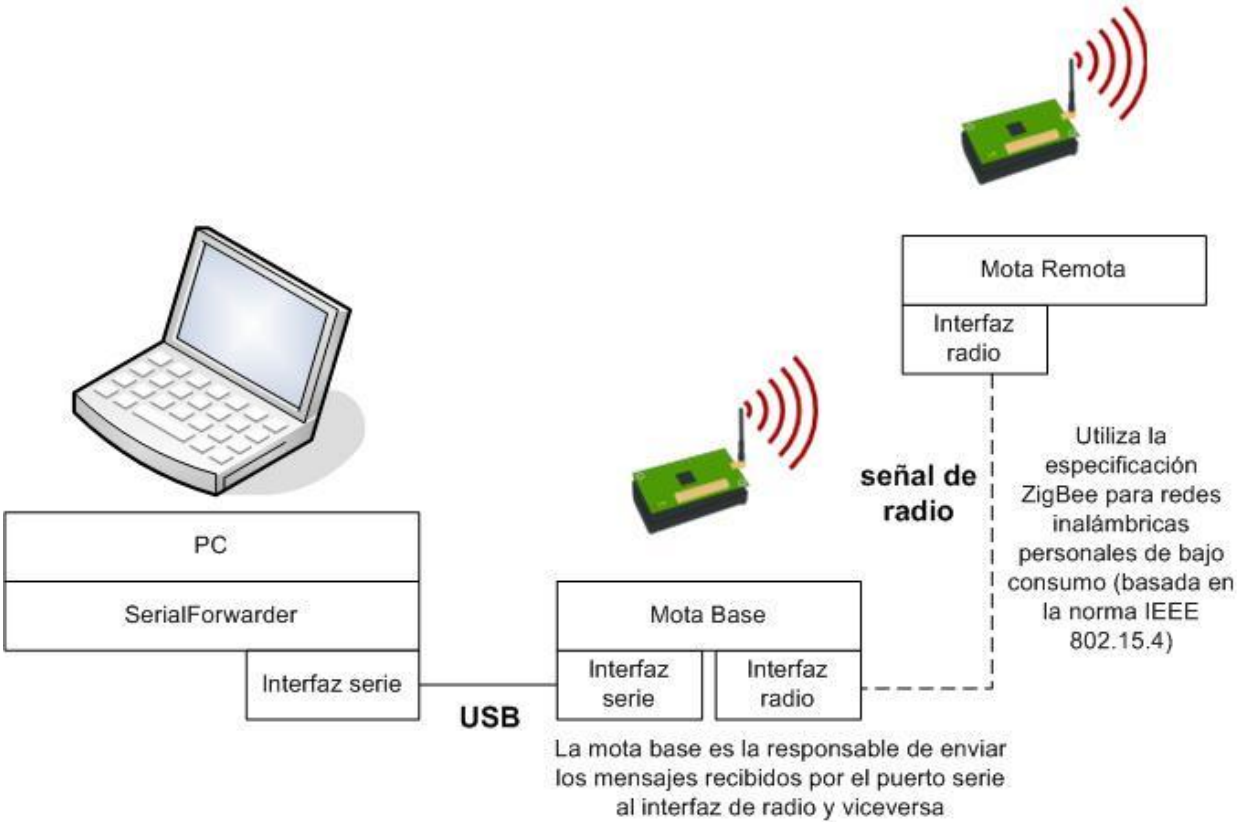

Figura 3. Componentes funcionales y estructura de red TinyDomus

Como podemos comprobar, cada parte del sistema, realiza una tarea determinada y no dispone de mecanismos redundante de tolerancia a fallos. El mal funcionamiento de alguno de los nodos, provocará que quede fuera de servicio. En el siguiente apartado veremos detalladamente el intercambio de mensajes que tiene lugar en cada una de las situaciones anteriores.

4. Descripción detallada

En esta sección analizaremos cada uno de los procesos que internamente se llevan a cabo cuando tienen lugar las operaciones explicadas anteriormente, esto es, desde que ponemos en marcha el sistema e interactuamos con él, hasta que lo detenemos completamente al finalizar la aplicación. Toda la lógica de comunicación entre el sistema central de control y la mota remota se lleva a cabo a través del intercambio de mensajes.

El modelo de mensajes que utiliza TinyOS está basado en Active Messages (AM). Se trata de un mecanismo de comunicación en el que, cada mensaje, contiene un identificador que permite al nodo destino reconocerlo y, una estructura de datos que define los argumentos que contiene.

Hecha esta aclaración, diremos que TinyDomus utiliza hasta cinco tipos de mensajes diferentes para poder implementar todas sus funcionalidades. Cada uno de estos mensajes realiza una función doble: realizar una petición a la mota remota y retornar la información solicitada. Esta es la estructura de los diferentes mensajes:

- **AM_SENSORSMSG**: Mediante esta estructura se solicita/retorno la información referente a los sensores de luz y temperatura así como del voltaje de las baterías. Contiene los siguientes campos:
    - **moteID**: Contiene el ID de la mota que envía el mensaje
    - **countsTemp**: Devuelve el valor obtenido del sensor de temperatura
    - **countsPhoto**: Al retorno, contiene el valor del fotosensor
    - **countsBat**: Voltaje de la batería
    - **vrefmilivolts**: '2560'
    - **tipo**: Indica el tipo de petición que estamos haciendo(1: consultar valor fotosensor, 2: consultar sensor de temperatura y 3: consultar ambos sensores)

- **AM_TIMERMSG**: Se utiliza para consultar o modificar el valor del temporizador:
    - **moteID**: ID de la mota emisora
    - **consulta**: Indica el tipo de petición (1: consultar el valor actual y 2: modificar dicho valor)
    - **tempor**: Si se tratar de una consulta de tipo 2, contendrá el valor que queremos asignar al temporizador (valor 0 si queremos detenerlo). En la respuesta siempre contendrá el valor actual

- **AM_TEMPMSG**: Permite consultar o modificar los valores Mínimo y Máximo de configuración para el sensor de temperatura:

    - **moteID**: ID de la mota emisora
    - **consulta**: Contiene el tipo de petición (1: consultar los valores actuales ó 2: modificarlos)
    - **tempMin**: Para consultas de tipo 2 contendrá el nuevo valor mínimo. En la respuesta siempre contendrá el valor actual en la mota remota
    - **tempMax**: Igual que tempMin pero para el valor máximo de temperatura

- **AM_PHOTOMSG**: Se emplea para consultar o modificar el valor Mínimo asignado al sensor de luminosidad:

    - **moteID**: ID de la mota emisora
    - **consulta**: Contiene el tipo de petición (1: consultar el valores actual y 2: modificar dicho valor)
    - **photoMin**: Para consultas de tipo 2 contendrá el nuevo valor mínimo. Como en el resto de casos, la respuesta siempre contendrá el valor del parámetro en la mota remota

- **AM_HALLMSG**: Se emplea para informar acerca del estado del sensor de efecto Hall:

    - **moteID**: ID de la mota emisora
    - **hallOn**: Cuando el nodo remoto detecta una lectura de este sensor, envía un mensaje al PC con valor 1 en este campo y enciende el Led verde. El sistema de control responde automáticamente con el mismo mensaje pero con valor 0 en este campo. Al recibirlo, la mota remota apaga de nuevo el Led.

Al enviar los mensajes es necesario indicar el ID de la mota destinataria. En TinyDomus, todos los mensajes se envían en modo BROADCAST, es decir, para todos los posibles receptores (ID=65535).

Todas y cada una de las funcionalidades que se implementan en el panel del control del sistema, se hacen a través de alguno de los mensajes definidos anteriormente. A modo de ejemplo, detallaremos el proceso que tiene lugar cuando iniciamos TinyDomus.

Durante la puesta en marcha del sistema de control: Al arrancar TinyDomus (ver apartado 10.2.

Ejecución y Compilación), la primera tarea que se realiza es la de inicializar tanto la interfaz gráfica (clase ControlRemoto), como las clases java que gestionarán la recepción de mensajes desde la mota remota y el envío de las peticiones a la misma. Para ello, TinyDomus implementa la interfaz Message Listener y registra los manejadores asociados a cada uno de los diferentes mensajes.

Una vez activada la interfaz gráfica, se solicitan tanto las lecturas de los sensores de luz y temperatura como la consulta de los parámetros de configuración enviando las siguientes solicitudes:

- Mensaje AM_SENSORSMSG (tipo=3): Lectura de los sensores de luz y temperatura
- Mensaje AM_TIMERMSG (consulta=1): Consulta valor temporizador
- Mensaje AM_TEMPMSG (consulta=1): Consulta valores MIN/MAX temperatura
- Mensaje AM_PHOTOMSG (consulta=1): Consulta valor MIN fotosensor

El siguiente diagrama muestra la secuencia de procesos que tiene lugar desde el momento en que se envían las peticiones al nodo remoto:

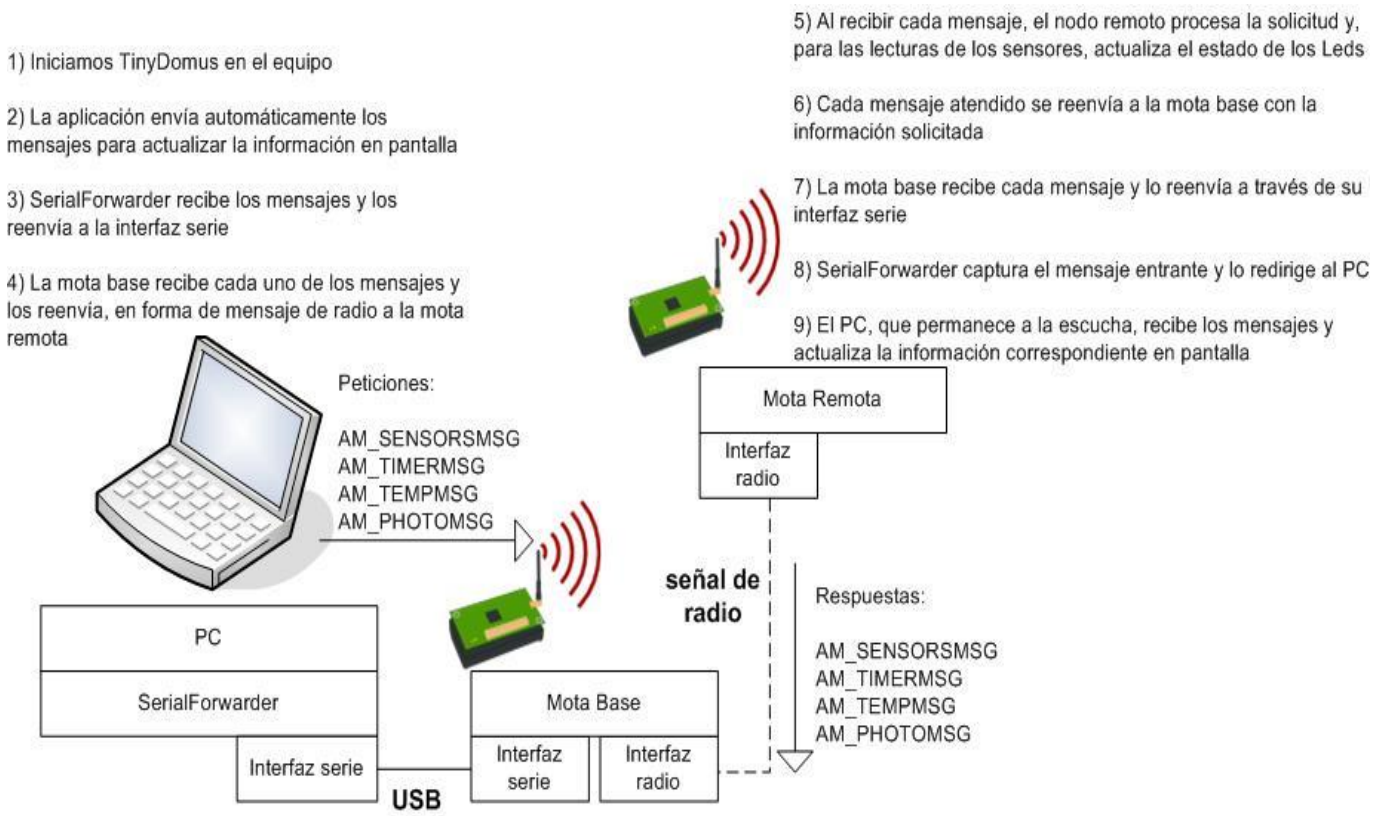

Figura 4. Ejemplo de comunicación entre PC y nodo remoto

Un apartado que merece especial atención para conocer en detalle el funcionamiento del sistema, es el relativo a mecanismos activados en la mota base para tratar de mantenerla en todo momento operativa.

Como hemos comentado en apartados anteriores, en la mota base hay un módulo de software, WatchDogC, que vigila continuamente el estado de funcionamiento de la misma. Su modo de funcionamiento es el de poner en marcha un temporizador establecido por nosotros en 8 segundos. Cada vez que lo mota base envía información a la mota remota este temporizador se inicia a cero. Si vence el tiempo establecido, WatchDogC fuerza un reseteo de la mota.

Este modo de funcionamiento tiene un efecto secundario no deseado: si la mota está en reposo también se resetea cada 8 segundos (que podría coincidir con la recepción de algún mensaje desde o hacia el PC). Para evitarlo, en el sistema de control hay programada una tarea que envía consultas a la mota remota cada 5 segundos: alternativamente, se van enviando mensajes del tipo AM_TIMERMSG (consulta valor temporizador), AM_TEMPMSG (consulta parámetros temperatura) y AM_PHOTOMSG (consulta parámetro fotosensor), que fuerzan a la mota base a seguir activa

Los siguientes diagramas de bloques muestran el tratamiento para cada sensor:

- Diagrama gestión sensor de luminosidad: A intervalos regulares o bajo petición, la mota remota realiza una lectura del sensor de luminosidad. Una vez obtenido el dato, gestiona el estado del Led asociado y envía el valor leído al sistema de control vía mensaje de radio.

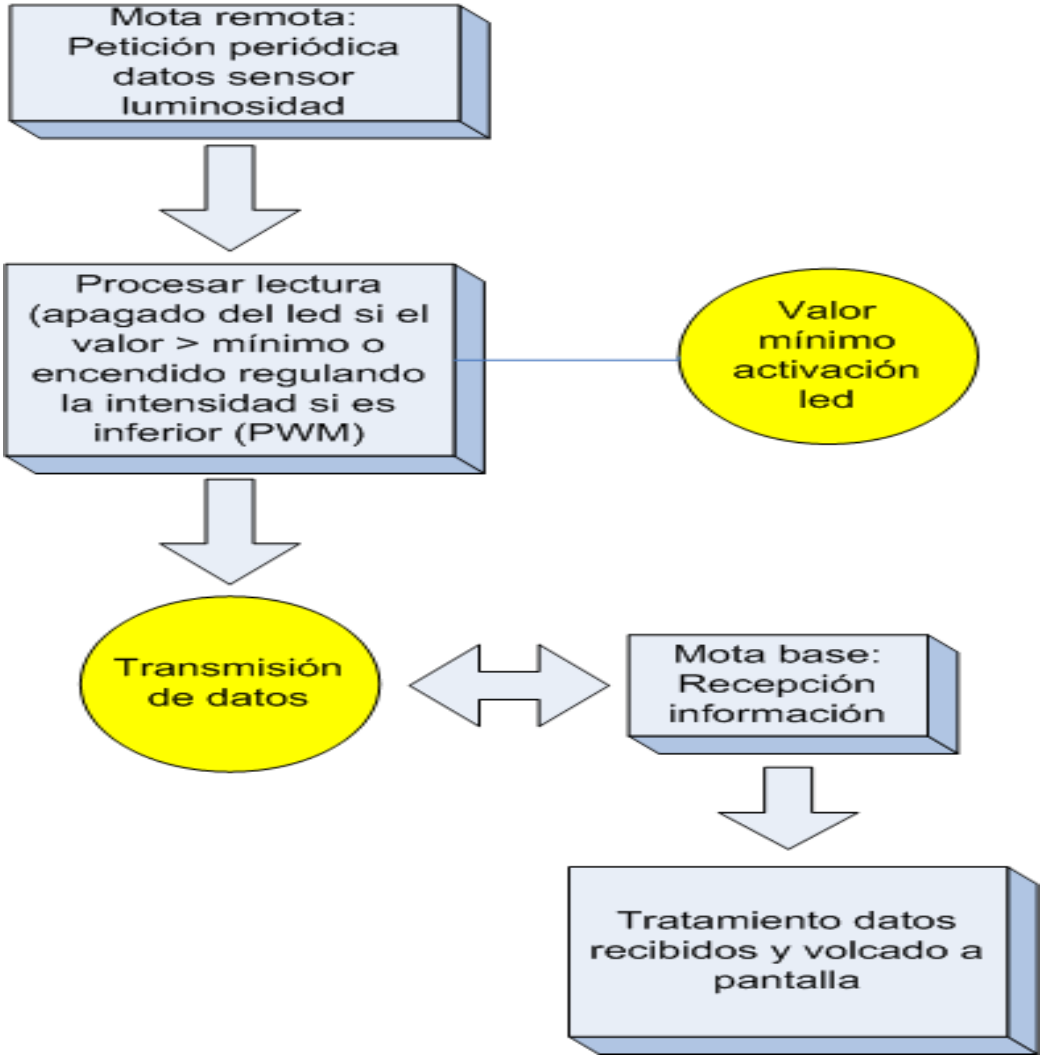

Figura 5. Diagrama gestión Sensor de Luminosidad

- Diagrama gestión sensor de temperatura: En cada petición cuando se activa el temporizador, la mota remota consulta el valor del sensor de temperatura, gestiona el estado del Leds de temperatura y envía un mensaje AM_SENSORSMSG que será procesado por la aplicación de escritorio.

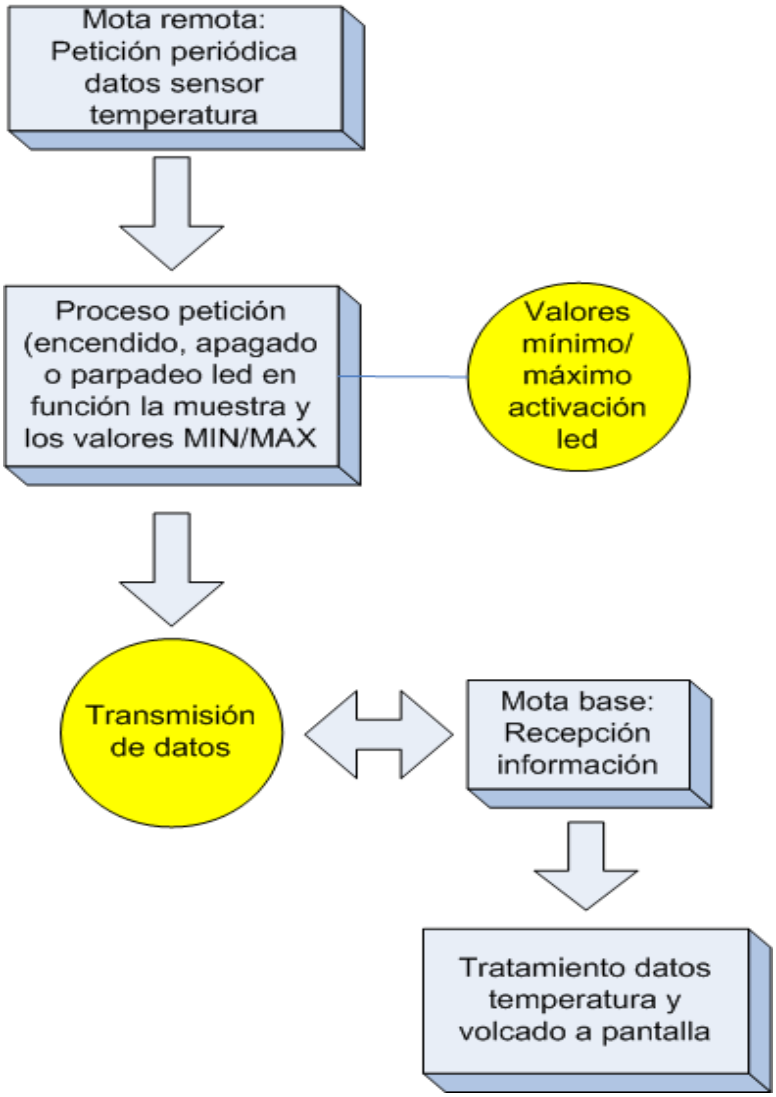

Figura 6. Diagrama gestión Sensor de Temperatura

- Diagrama gestión sensor efecto Hall: Cuando el sensor de efecto Hall detecta un pico magnético se activa una interrupción hardware que es capturada por la mota remota. Seguidamente, modifica el estado del Led asociado y envía un mensaje al sistema de control indicando su activación. La respuesta que reciba desde el sistema de control, será utilizada por el software de la mota para volver a apagar el Led.

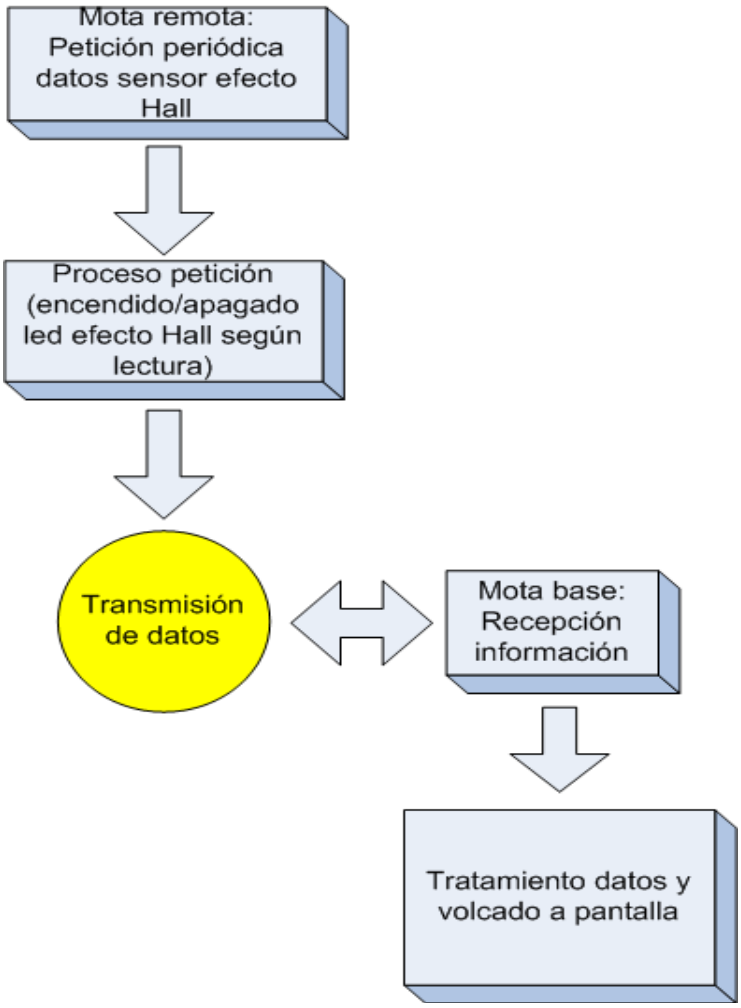

Figura 7. Diagrama gestión Sensor de efecto Hall

5. Viabilidad técnica

En la actualidad existen las soluciones técnicas, tanto a nivel de hardware como de software, requeridas por nuestro sistema para hacer viable la implementación del sistema desarrollado:

- Un juego de motas de la familiar COU24 compatibles con el software desarrollado

- Un equipo donde conectar la mota base y desde donde pueda ejecutar la aplicación de escritorio desarrollada

- Los actuadores (sistemas electromecánicos), que interconecten nuestra aplicación con el mundo real, es decir, que puedan subir/bajar persianas, controlar el sistema de iluminación o gestionar el sistema de climatización del hogar

El proyecto de hogar digital desarrollado por ASIMELEC o la solución comercial SENSbee desarrollada por SAYME (ver apartado 2.2. Estudio de mercado), son dos ejemplos reales que avalan la viabilidad técnica del proyecto.

Evidentemente, nuestro proyecto necesita de una segunda fase de investigación y desarrollo para convertirlo en producto que pueda competir, de alguna manera, con las soluciones comerciales existentes en la actualidad.

Entre los puntos fuertes a destacar y que facilitan la viabilidad de implantación del sistema, podemos destacar los siguientes:

- Facilidad de implantación: El proyecto en sí mismo no necesita de una infraestructura muy compleja para su despliegue (en este punto faltaría por analizar la complejidad de implantar actuadores que sean capaces de recibir peticiones de nuestro sistema para modificar la intensidad de encendido de las lámparas en función del grado de luminosidad en el ambiente, activar el sistema de calefacción si baja la temperatura, etc.)

- Bajo nivel de mantenimiento: A falta de un estudio serio acerca de la vida de los nodos remotos en un entorno real que funcione las 24 horas del día, el sistema en sí mismo, no necesitaría de un alto nivel de mantenimiento para permanecer operativo

- Facilidad de uso: El sistema de control diseñado permite a cualquier usuario interactuar con el sistema de una manera sencilla y directa

Por otra parte, el grado de desarrollo de la solución obtenida junto con las limitaciones tecnológicas del hardware utilizado, nos muestra claramente los puntos que deberemos resolver para hacer viable técnicamente su implantación en un entorno real:

- La tecnología actual de redes de sensores inalámbricos no resuelve el problema de la dependencia energética de las motas. Una asignatura pendiente de las WNS es la de desarrollar nodos que sean, en la medida de lo posible, autosuficientes energéticamente

- El producto desarrollado es una simulación de un hogar domótico. Para su implementación en un entorno real necesitaría de una segunda fase de formación y desarrollo para dotarlo de sistemas actuadores que, a partir de la información obtenida por el sistema, pudiera modificar el estado de los sistema que tratamos de controlar (subir o bajar persianas, controlar el encendido de lámparas en función del grado de iluminación ambiental, encender el sistema de calefacción cuando la temperatura sea inferior a un valor mínimo establecido por el usuario, etc.)

- La interfaz gráfica está basada en una aplicación de escritorio. Esta solución implica que el usuario sólo puede interactuar desde un punto del hogar con el sistema el sistema de control. En una implantación real probablemente la solución más acertada podría ser un entorno Web que permitiera al usuario, desde cualquier lugar, gestionar los parámetros del sistema y obtener información detallada del mismo

6. Valoración económica

Para llevar a cabo una estimación económica de la solución desarrollada, trataremos de analizar cada uno de los factores que han sido necesarios para poder llevar a cabo el proyecto en su totalidad:

- Fase de análisis y desarrollo de cada uno de los componentes de software. Durante la fase de planificación inicial del proyecto se estimaron un total de 63 días para llevar a cabo las etapas de análisis, diseño, implementación y pruebas. En total, el tiempo empleado para finalizar estas etapas ha sido de 54 días. De este dato obtendremos el costo de las horas de trabajo necesarias para llevarlo a cabo. No obstante, debemos tener muy claro que un dato a nivel orientativo. Hay que tener en cuenta que el tiempo empleado es la consecuencia de tener que compaginar el trabajo del TFC con las responsabilidades reales del día a día (en un entorno real de desarrollo en el que, además, participaría un equipo humano acorde con los objetivos que se persiguen, el resultado podría ser muy distinto)

- Requerimientos de hardware para poder implementar el proyecto. Son muy pocos los requisitos de equipamiento que la aplicación, en su estado actual, necesita para ser totalmente operativa:

    - Un juego de motas de la familia COU24 compatibles con el software desarrollado. No disponemos datos acerca del valor actual de las dos motas pero es la parte que menos influirá en el presupuesto total del proyecto
    - Y un equipo informático donde podamos desplegar la aplicación de escritorio desarrollada. Este equipo no requiere características especiales para su funcionamiento
- Fase de instalación y mantenimiento. Para la fase de instalación (ver apartado 10.2. Ejecución y compilación), necesitaremos disponer de la versión final de nuestro software instalada en cada uno de los componentes del sistema. Un punto a tener en cuenta durante la instalación será la elección de la mejor ubicación para la mota remota en el entorno

  En cuanto a la fase de mantenimiento, a falta de conocer datos reales acerca de la fiabilidad de los nodos inalámbricos y la duración real de las baterías, entendemos que el costo de mantener la instalación operativa no es un dato significativo para la implantación del mismo

En base a los puntos anteriormente analizados, podemos obtener una estimación bastante orientativa acerca de la posible valoración económica que supondrá la puesta en funcionamiento del proyecto.

A pesar de todo lo expuesto anteriormente, y dado el carácter académico del proyecto, debemos valorarlo en su justa medida. El nivel de desarrollo de la solución propuesta no puede valorarse económicamente como si se tratara de una aplicación comercial. Le queda mucho camino por recorrer para poder llegar a considerarse como tal. Algunas razones podrían ser estas:

- No interactuar realmente con el medio. Simplemente gestiona de Leds para simular acciones reales

- El interfaz gráfico podría cumplir con los requisitos funcionales necesarios pero, su uso está restringido al ámbito del hogar. En una solución comercial se debería considerar por optar por trasladar dicha funcionalidad a un mando a distancia convencional y/o dotarlo de soporte Web que nos permitiera interactuar con el entorno desde cualquier lugar

- Sería necesario analizar si el hardware de motas utilizado cumple con los requerimientos mínimos, o dicho de otro modo, resulta ser el más apto para el despliegue de una solución real en la que las exigencias de fiabilidad y bajo mantenimiento son una condición ineludible

No obstante, entendemos que el valor económico no debería medirse pensando solamente pensando en el proyecto diseñado para el TFC, sino por los conocimientos adquiridos:

- Familiarizarse con las tecnologías redes de inalámbricas

- Implementación de una plataforma completa de desarrollo

- Creación de un código totalmente funcional para las motas y diseño de una interfaz gráfica totalmente integrada con la red inalámbrica de sensores

- Conocimiento del estado actual de la tecnología y las previsiones de desarrollo a medio plazo

Si nos atenemos a estos argumentos, creemos que el verdadero valor del proyecto desarrollado son, principalmente, las posibilidades reales de futuro que nos brindan los conocimientos adquiridos.

Y aquí es donde, a nuestro entender, encontramos su valor económico real: el proyecto desarrollado nos ha permitido conocer, y poner en práctica, las grandes posibilidades que ofrece la tecnología. Supone un excelente punto de partida para continuar profundizando en la materia y nos abre las puertas para poder analizar la demanda real que ofrece el mercado laboral dentro de este sector.

7. Conclusiones

7.1. Conclusiones

Tras los datos obtenidos en el estudio mercado podemos concluir con toda seguridad que la tecnología de WSN está aquí para quedarse. Todas las previsiones auguran un crecimiento cuantitativo y cualitativo a nivel mundial. Estudios de consultoras tecnológicas e instituciones académicas de primer nivel (ABIresearch, Dolcera, MIT), respaldan estas afirmaciones.

Se trata de una tecnología relativamente nueva, que ha despertado un gran interés por sus múltiples aplicaciones y ventajas. Sólo tenemos que buscar información en Internet para darnos cuenta del despegue que está sufriendo esta tecnología. Además de las múltiples soluciones que ya existen en la actualidad, hay muchos proyectos de investigación abiertos (los citados BSAC y Genesi son sólo algunos ejemplos), para abordar los problemas pendientes de resolver (resumiendo lo descrito en apartados anteriores: mejorar el grado de precisión de los sensores soportados, mejoras en los tiempos de respuesta, necesidad de desarrollar protocolos especiales para aplicaciones de tiempo real, gestión energética de las motas, aumento de su ciclo de vida, robustez y escalabilidad de las redes, mejora de las herramientas de desarrollo o mejora de los dispositivos para su adaptación a cambios).

La evolución de la tecnología y de sus herramientas de desarrollo, implicarán un aumento de las soluciones y aplicaciones de esta nueva tecnología. El concepto de "Internet de las Cosas" (objetos de la vida cotidiana interconectados y compartiendo información), puede llegar a convertirse en una realidad si la evolución de las redes de sensores inalámbricas alcanzan los niveles de desarrollo estimados. Por el momento, todo apunta hacia un futuro más que prometedor.

TinyDomus ha supuesto una aproximación hacia las posibilidades reales de las redes de sensores inalámbricas. A pesar no de no estar familiarizados con este tipo de entornos, nos ha dado una idea nítida acerca de la multitud de puertas que se abren ante nosotros. Estamos seguros de que, tal y como auguran las compañías especializadas en nuevas tecnologías, en un futuro muy cercano estas tecnologías formarán parte de muchas de las actividades que realizamos a diario.

7.2. Propuesta de mejoras

Partiendo de las mejoras propuesta en la fase de planificación del proyecto y, dada la clara orientación del mismo a la automatización de procesos en la vivienda, algunas posibles mejoras a llevar a cabo sobre la plataforma **TinyDomus** podrían ser:

- Dotarla de mecanismos actuadores para interactuar realmente con los dispositivos del hogar (encender/apagar luces en función de la luminosidad, automatismos para bajar/subir persianas, gestionar sistemas de calefacción/aire acondicionado, enviar avisos si una ventana/puerta se abre cuando estamos fuera de casa y/o activar sistemas de alarma)

- Posibilidad de acceso a la plataforma vía Web para poder acceder al sistema en tiempo real. Para ello sólo necesitaríamos acceso a Internet y cualquier navegador de los existentes en el mercado. Este nuevo interfaz dispondría de todas las funcionalidades de la aplicación de PC.

- Envío de mensajes SMS a dispositivos móviles

- Ampliación de los sistemas de detección para incluir sistemas ignífugos y detectores de humedad

- Posibilidad de gestionar perfiles de usuario para poder adaptar las condiciones de una sala a la preferencias de cada individuo

En definitiva, utilizar toda la tecnología disponible con vistas a crear un hogar domótico que facilite la vida de los inquilinos y redunde en un mejor aprovechamiento de los recursos.

7.3. Autoevaluación

Para tratar de llevar a cabo una autoevaluación del trabajo realizado, que sea lo más objetiva posible, me he basado en estos cuatro apartados:

- Aprovechamiento de los materiales recibidos y de la documentación disponible

- Disponibilidad temporal para llevar a cabo las distintas fases del proyecto

- Cumplimiento de los objetivos temporales marcados a comienzos del TFC

- Análisis, desarrollo e implantación de un software que cumpla con todos los requisitos definidos y consensuados previamente con el Tutor

Atendiendo a estos puntos, y teniendo en cuenta todas las horas de esfuerzo personal dedicadas a la finalización del mismo, considero que la calificación final no debería quedar por debajo de un notable.

Tratando de analizar el trabajo de manera objetiva, entiendo que siempre hay puntos que se pueden mejorar pero, vuelvo a repetir, si tengo que atenerme a los puntos indicados, y teniendo en cuenta que no tenía ningún tipo de formación previa en esta materia, creo que sería una justa calificación.

8. Glosario

Atmel: Atmel Corporation, e slider mundial en el diseño y fabricación de microcontroladores, lógica avanzada sensores sensibles al tacto, memorias no volátiles y componentes de radio frecuencia (RF).

ETHZ: Instituto Tecnológico de Zurich.

IEEE: Institute of Electrical and Electronics Engineers. Entidad dedicada a la elaboración de estándares que faciliten los desarrollos tecnológicos sustentados en una plataforma común.

IEE 802: Se trata de una serie de estándares elaborado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) que actúa sobre Redes de ordenadores. Concretamente y, según su propia definición, sobre redes de área local (LAN) y redes de área metropolitana (MAN). También se usa el nombre **IEEE 802** para referirse a los estándares que proponen, algunos de los cuales son muy conocidos: Ethernet (IEEE 802.3), Wi-FI (IEEE 802.11) o Bluetooth (IEEE 802.15).

Inmótica: Consiste en incorporar, al equipamiento de edificios de uso terciario o industrial (oficinas, edificios corporativos, hoteleros, empresariales y similares), sistemas de gestión técnica automatizada de las instalaciones, con el objetivo de reducir el consumo de energía, aumentar el confort y la seguridad de los mismos.

Led: Diodo emisor de luz (Light-emitting diode).

Linux: Hace referencia a las computadoras basadas en el sistemas operativo libres similares a Unix y que utilizan el núcleo (kernel) libre, Linux. Su desarrollo se basa en software libre. El nombre *Linux* viene del núcleo Linux, inicialmente escrito por Linus Torvalds en 1991.

MAC: Medium Access Controller. Se emplean en la familia de estándares IEEE 802 definir la subcapa de control de acceso al medio.

Microcontrolador (uC): Un microcontrolador es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida.

MIT: Massachusetts Institute of Technology.

MOTE (mota): Mobile Transmission Elements.

Sistema embebido o empotrado (también denominados incrustados o embutidos): Un sistema embebido es una combinación de hardware y software de computación diseñado para realizar una o varias tareas concretas, a menudo dentro de una restricción de computación en tiempo real , y que puede contener un software preinstalado para realizar exclusivamente dichas funciones o disponer de un interfaz programable. Los diseños de estos sistemas suelen optimizarse para reducir su tamaño, costos de producción, fiabilidad y eficiencia energética. Sus aplicaciones abarcan un enorme abanico de sectores (procesos industriales, automóviles, equipos médicos, cámaras, pequeños electrodomésticos, aviones, máquinas de venta, juguetes, telefonía, etc.).

TinyOS: Es un sistema operativo orientado a eventos. Su filosofía es el uso de tareas, para realizar operaciones con mayor penalización de tiempo, y eventos. Los eventos se lanzan de manera asíncrona mediante interrupciones que posponen las operaciones utilizando tareas. FIFO es el modelo utilizado para procesar las tareas. Cuando no hay tareas pendientes, el uC queda en estado de reposo.

TFC: Trabajo Fin de Carrera.

UART: Universal Asynchronous receiver/transmitter.

Ubuntu: Es un sistema operativo un núcleo Linux y su origen está basado en Debian (sistema operativo libre). Ubuntu está orientado al usuario promedio, con un fuerte enfoque en mejorar su experiencia y la facilidad de uso.

USB: Universal Serial Bus. El Bus de Serie Universal es una interfaz que provee un estándar de bus serie para conectar dispositivos a un ordenador personal.

USN: Ubiquitous Sensors Networks. El término hace referencia al despliegue redes de sensores en cualquier lugar susceptible de ser monitorizado. Términos como 'Anywhere, Anytime, by Anyone, and Anything' o 'Internet de las cosas' están íntimamente ligados a este concepto.

WSN: Wireless Sensors Networks. Consiste en una red de nodos, equipados con sensores, que colaboran en una tarea común. Las redes de sensores están formadas por un grupo de sensores con ciertas capacidades sensitivas y de comunicación inalámbrica los cuales permiten formar redes ad hoc sin infraestructura física preestablecida ni administración central.

ZigBee: Es un estándar de transmisión de datos de baja velocidad, en dos direcciones, para redes de automatización industriales y domésticas. Este estándar soporta bandas de radio sin licencia de 2,4 GHz.

9. Bibliografía

[ABIresearch] http://www.abiresearch.com/home.jsp

[ASIMELEC] http://www.hogardigital.asimelec.es/

[Atmel] http://www2.atmel.com/

[BSAC] http://www-bsac.eecs.berkeley.edu/

[Dolcera] http://www.dolcera.com/

[ETHZ] http://www.ethz.ch/index_EN

[Genesi] http://genesi.di.uniroma1.it/

[Java] http://download.oracle.com/javase/tutorial/ (en http://www.oracle.com/technetwork/java/index.html)
[Java] Herbert Schildt. Java 2 Manual de Referencia (Cuarta Edición). Osborne McGraw-Hill.

[MIT] http://web.mit.edu/

[SAYME] http://www.sayme.es/

[TFC Sistemas Empotrados] http://cv.uoc.edu/app/mediawiki14/wiki/Inici24

[TinyOS] http://tinyos.net/

[Ubuntu] http://www.ubuntu.com/

[Wikipedia] http://es.wikipedia.org

[Yeti 2] http://tos-ide.ethz.ch/wiki/index.php

[ZigBee] http://www.zigbee.org/

10. Anexos

Este apartado comprende dos secciones fundamentales para la puesta en funcionamientos de nuestra aplicación: cómo compilar y ejecutar cada uno de los componentes de software desarrollados. Además contiene un manual de funcionamiento para el usuario final. Así mismo, incluye un apartado dedicado a SerialForwarder. Este software desarrollado por terceras partes, hace las veces de intermediario entre el puerto USB y nuestra aplicación.

10.1. Manual de usuario

En este apartado detallaremos paso a paso cómo poner en funcionamiento nuestra aplicación. Explicaremos cada una de las funcionalidades existentes en el panel del control y cómo puede utilizarlas cualquier usuario que tenga acceso al sistema.

Para la puesta en marcha de la aplicación deberemos seguir los siguientes pasos:

- Conectaremos la mota base al puerto USB. Atención: Deberemos conocer el número de puerto USB que vamos a utilizar. En el manual suponemos que se trata del puerto USB0. En caso de tratarse de otro puerto, deberemos sustituir este valor por el correcto

- Iniciaremos SerialForwader (ver apartado 10.3. SerialForwarder), en el equipo. Para arrancarlo simplemente deberemos abrir una sesión de terminal y ejecutar el siguiente comando:

  $java net.tinyos.sf.SerialForwarder –comm serial@/dev/ttyUSB0:19200

- Si la mota remota estaba parada, la pondremos en funcionamiento

- Iniciaremos TinyDomus en el equipo local. Para ello abriremos una nueva sesión de terminal e introduciremos el siguiente comando:

  /ProyectoDomus/TinyDomus/bin/java domus.TinyDomus

Si hemos seguido correctamente los puntos anteriores, no tardaremos en ver aparecen en la pantalla de nuestro ordenador, el panel de control de la aplicación (si tenemos algún tipo de problema y no se

muestra el formulario o, bien, retorna algún tipo de error, en primer lugar, deberemos revisar que todos los puntos anteriores se hayan realizado correctamente). Una vez arrancado, TinyDomus solicitará, automáticamente, información al nodo remoto mostrar en pantalla los valores iniciales de los sensores y los parámetros de configuración que tiene asignados dicho dispositivo por defecto.

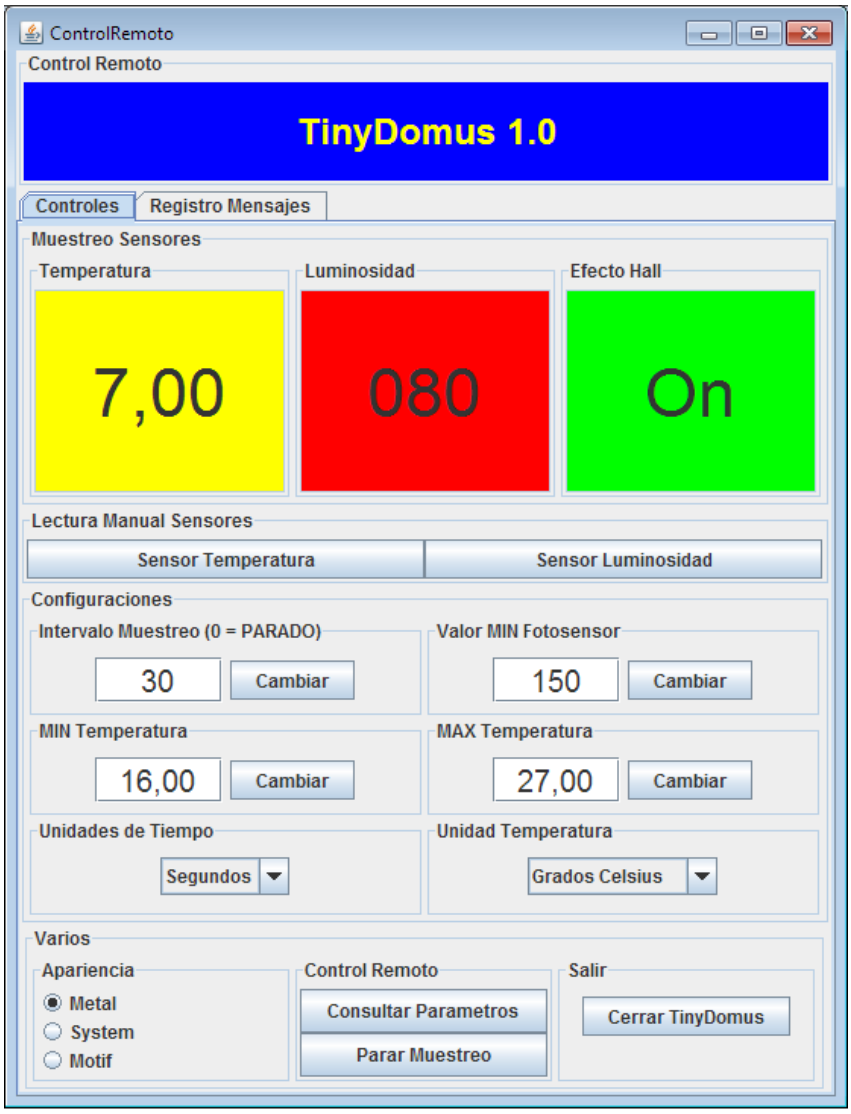

Figura 8. Aspecto Panel de Control TinyDomus

Como podemos observar en la imagen, la ventana princpal consta de dos pestañas: "Controles" y "Registro Mensajes". A continuación procederemos a describir detalladamente el contenido de cada una:

- **Pestaña "Controles"**: Desde aquí podremos obtener información acerca del estado de cada no de los sensores. Además, dispondremos de todas las opciones que nos permiten interactuar con el sistema.

  1) En la parte superior del formulario, apartado **"Muestreo Sensores"**, el sistema nos mostrará la última lectura realizadas a cada uno de los distintos sensores. Tomando como ejemplo imagen que se adjunta, tenemos la siguiente información:

     - **Sensor de Temperatura:** El último valor obtenido del sensor es de *7,00 grados Celsius*

     - **Sensor de Luminosidad:** El índice de luminosidad detectado por el fotosensor del nodo remoto ha sido de *080*

     - **Sensor Efecto Hall:** Estado *On* (Activo). Indica que, en el momento de capturar la imagen, el sensor de efecto Hall había detectado un pico magnético. En caso contrario su valor sería *Off* (Inactivo)

  Como podemos ver en la imagen, el color de fondo para cada uno de los cuadros de texto que muestran la información es de un color diferente. A continuación explicaremos la razón de dichas diferencias.

  Las motas disponen de tres Leds. Cada uno de estos Leds tiene un color diferentes (amarillo, rojo y verde). TinyDomus asocia cada Led a uno de los sensores de esta forma:

     - El Led AMARILLO se corresponde con el sensor de TEMPERATURA
     - El Led ROJO con el sensor de LUMINOSIDAD
     - Y el VERDE, con el sensor de EFECTO HALL

  El color de fondo de cada sensor, simplemente, tratará de imitar el estado actual de cada uno de estos Leds. Así, si el Led de Temperatura de la mota remota (color amarillo), está parpadeando, el color de fondo del cuadro de texto correspondiente a este sensor, también se verá de color amarillo y con efecto de parpadeo.

  En el apartado, "Configuraciones", explicaremos cada uno de los criterios que se han seguido para encender o apagar cada uno de estos Leds.

2) Apartado **"Lectura manual de Sensores"**. Por defecto al arrancar el sistema, el sistema tiene definido un periodo de tiempo por defecto para realizar lecturas periódicas de los sensores de luz y temperatura (más adelante veremos que este valor es totalmente configurable).

   La funcionalidad de este apartado es de que el usuario pueda poder solicitar lecturas de cada uno de estos sensores en cualquier momento, sin necesidad de esperar a que se actualice la información en pantalla. Para ello disponemos de dos botones:

   - **"Sensor Temperatura"**: Envía un solicitud al nodo remoto para que que haga una lectura del sensor de temperatura, nos envíe el valor leído y se actualice la información en pantalla
   - Y **"Sensor Luminosidad"**: La funcionalidad es exactamente igual que para el anterior pero con el sensor de luminosidad

   El comportamiento del sensor de Efecto Hall es independiente del resto de sensores. Su activación es completamente asíncrona y no depende de ningún temporizador o botón de lectura. Tan pronto como se detecta un campo magnético, la mota remota envía un mensaje indicando la activación del mismo (para comprobarlo sólo tenemos que acercar un imán al sensor en el dispositivo remoto). Esto provocará el encendido del Led de Efecto Hall en la mota remota (VERDE)

   Su finalidad es la de simular la activación del sistema de alarma. Además de encender el Led de efecto Hall, el nodo remoto envía un mensaje al sistema de control que será utilizado para actualizar en pantalla el estado del sensor (mostrando la palabra "On" y cambiando a color VERDE el color del fondo del cuadro de texto donde se muestra). Esta es la razón por la cual, en la imagen anterior, el sensor de efecto Hall aparecía con la palabra "On" y el fondo de color VERDE

   Por su parte, la aplicación de escritorio enviará un segundo mensaje de confirmación a la mota remota que provocará el apagado del Led (en pantalla aparecerá la palabra "Off" y se volverá a cambiar a BLANCO el color del fondo del cuadro de texto)

3) Apartado **"Configuraciones"**. Aquí encontraremos los valores configuración que nos permitiran personalizar a nuestro gusto el funcionamiento del sistema:

- **Intervalo muestreo:** Nos muestra con qué frecuencia, la mota remota, está realizando la lectura de los sensores. Para modificar este valor, tendremos que acceder al cuadro de texto, introducir el nuevo valor y pulsar el botón "Cambiar". Si se modifica el valor pero no se pulsa el botón adjunto "Cambiar" el nuevo valor no se actualizará en la mota remota y, cuando se actualize la información en pantalla, se volverá a mostrar el valor actual definido en el nodo remoto (este comportamiento es exactamente igual para el resto de parámetros)

- **Valor MIN Fotosensor:** Por debajo de dicho este valor Mínimo se activará el Led de Luminosidad en en el nodo remoto (Led ROJO). El compotamiento de este Led simula el encendido del sistema de iluminación del hogar

  Tiene un particularidad, el Led se irá encendiendo con mayor intensidad a medida que haya menor luz exterior. Volviendo al color de fondo del cuadro de texto del sensor de Luminosidad, veremos que, mientras el valor leído del fotosensor esté por encima de este valor Mínimo, el fondo será de color BLANCO. Tan pronto como la lectura del sensor esté por debajo de este Mínimo, el color de fondo cambiará a ROJO. En la imagen superior, el valor MIN del fotosensor es 150 mientras que el valor leído es de 080. Por lo tanto, el color de fondo aparece en rojo

- **MIN Temperatura:** El valor Mínimo de Temperatura indica el valor por debajo del cual se debería encender el Sistema de Calefacción. Si la lectura del sensor de Temperatura arroja un valor por debajo de este valor Mínimo, se encenderá el Led de Temperatura en la mota remota

  Esto también afectará en pantalla al color de fondo del cuadro de texto correspondiente a este sensor. Mientras el sensor de temperatura devuelva valores por encima del valor Mínimo y por encima del valor Máximo (lo veremos a continuación), el color de fondo será BLANCO. Tan pronto como el valor que devuelve este sensor esté por debajo de este valor Mínimo, cambiará a AMARILLO. Volviendo a la imagen de ejemplo, como el valor que devuelve el sensor (7,00 grados Celsius), está por debajo del mínimo (16,00), el color de fondo se muestra en color amarillo

- **MAX Temperatura:** Por encima de este valor, se activará el Sistema de Aire acondicionado. Si el valor devuelto por el sensor de temperatura está por encima de esta Máximo, el Led de Temperatura en el nodo remoto comenzará a parpadear

En pantalla, veremos como el color de fondo del sensor de Temperatura comienza a parpadear, cambiando entre los colores AMARILLO y BLANCO

- **Unidades de Tiempo:** Esta lista desplegable nos permite seleccionar las unidades en las que preferimos ver e introducir el valor del intervalo de muestreo. Podremos elegir entre Segundos o Minutos. Los datos de pantalla se adaptarán a la unidad seleccionada.

- **Unidad Temperatura:** Aquí podremos elegir en qué unidades queremos ver o introducir los valores relativos a temperaturas (valor del sensor de temperatura, o valores mínimo y máximo de configuración). Nos permite seleccionar entre grados Celsius o grados Farenheit.

4) Apartado **"Varios"**. Ofrece al usuario las siguientes posibilidades:

- **Apariencia**: Nos permitirá modificar la apariencia del interfaz gráfico. Los aspectos disponibles son Metal, Motif y System (GTK+ para sistemas Linux y Windows para los sistemas operativos de Microsoft)

- **Control Remoto**: En este apartado encontramos dos botones que tienen utilidades muy diferentes:

  - Botón **"Consultar Parametros"**: Envía una petición a la mota remota para que nos devuelva los parámetros de configuración que tiene almacenados. A la recepción del mensaje, estos valores se refrescarán en pantalla.

  - Botón **"Parar Muestreo"**: Su función es la de detener el temporizador que realiza el muestreo automático de los sensores de luz y temperatura.

    Al pulsar el botón se envía una petición al nodo remoto para detener el muestreo automático. A la respuesta, el valor de campo "Intervalo de Muestreo" cambiará a 0, indicando que el temporizador se ha detenido. Para ponerlo en marcha nuevamente sólo tendremos que modificar el valor "Intervalo de Muestre" y pulsar el botón "Cambiar"

La funcionalidad del botón parar muestreo es exactamente la misma que pulsar el botón "USR" en la mota base, es decir, la mota conectada al puerto USB.

- Botón **"Cerrar TinyDomus"**: Provocará el cierre de la aplicación

- **Pestaña "Registro Mensajes"**: En la pestaña "**Registro Mensajes**", podremos visualizar en modo texto, el trafíco de mensajes que está teniendo lugar entre la mota remota y el sistema de control. También se registra la fecha y hora en que se recibe o envía cada uno de los mensajes.

Como hemos podido comprobar, desde la interfaz gráfica de usuario, disponemos del control total del sistema. Todas y cada una de los funcionalidades que puede llevar a cabo la red de sensores, puede ser controlada desde aquí.

10.2. Ejecución y compilación

En el paquete de software que se entrega, los distintos componentes se encuentran ya compilados. No obstante en el repositorio central, ProyectoDomus, encontramos tres componentes:

- BaseDomus (basado en el código de BaseStation): Software a instalar en la mota que se conectará al puerto USB del pc (hace uso de los componentes MainC, LedsC, ActiveMessageC, HplAtm128GeneralIOC, HplAtm128InterruptC y WatchDogC). Para compilar BaseDomus, accederemos al directorio BaseDomus/src desde una sesión de terminal y ejecutaremos el comando "make cou24"

- RemoteDomus: Se instalará en la mota remota (utiliza los componentes MainC, LedsC, TimerMilliC, ActiveMessageC, AMSenderC, AMReceiverC, HplAtm128GeneralIOC, GestionLedsC y WatchDogC). En este caso accederemos al directorio RemoteDomus/src e igualmente ejecutaremos "make cou24"

Una vez compilados el software BaseDomus y RemoteDomus, conectaremos la mota base/remota (según sea el caso), al puerto USB y ejecutaremos el siguiente comando:

meshprog –t /dev/ttyUSB0 –f ./build/cou24/main.srec

Donde las motas se conectan al Puerto USB0 (sustituir por el adecuado en caso de utilizar uno diferente), y el comando se estará ejecutando desde el directorio principal de cada uno de los componentes: BaseDomus o RemoteDomus según sea el caso).

- TinyDomus: Software java de gestión y monitorización del sistema. Su clase principal, TinyDomus.java, se basa en el código de la clase MainTester.java (disponible en http://cv.uoc.edu/app/mediawiki14/wiki/Inici24, apartado "Test Program"). Implementa la clase MessageListener y utiliza las siguientes librerías:

```
net.tinyos.util.PrintStreamMessenger
import domus.mensajes.SensorsMsg;
import domus.mensajes.TimerMsg;
import domus.mensajes.TempMsg;
import domus.mensajes.PhotoMsg;
import domus.mensajes.HallMsg;
import net.tinyos.message.Message;
import net.tinyos.message.MessageListener;
import net.tinyos.message.MoteIF;
import net.tinyos.packet.BuildSource;
```

  Para la interfaz gráfica, TinyDomus.java, hace uso de una segunda clase: ControlRemoto.java. Esta nueva clase define completamente el nuevo interfaz de usuario. Hace uso de las siguientes librerías:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.Timer;
import java.text.*;
```

Tanto TinyDomus.java como ControlRemoto.java, se ubican en el paquete "domus". Adicionalmente, disponemos del paquete "domus.mensajes", formado por las clases SensorsMsg, TimerMsg y TempMsg, PhotoMsg y HallMsg. Estas clases, extienden la clase net.tinyos.message.Message, y se generar durante la compilación de RemoteDomus.

Para compilar todo el software definido en el proyecto Java TinyDomus, podemos utilizar las facilidades que nos proporcionan entornos de desarrollo tales como Eclipse, o bien, utilizar el siguiente comando:

/ProyectoDomus/TinyDomus/src$ javac –d ../bin ./domus/*.java

Al llegar a este punto, ya dispondremos del entorno necesario para ejecutar nuestra aplicación. Para ello, procederemos de la siguiente forma:

- Conectaremos la mota base al puerto USB0
- Iniciaremos SerialForwader en el equipo (java net.tinyos.sf.SerialForwarder –comm serial@/dev/ttyUSB0:19200).
- Pondremos en funcionamiento la mota remota.
- Iniciaremos TinyDomus en el equipo local mediante la siguiente línea de comandos:

/ProyectoDomus/TinyDomus/bin/java domus.TinyDomus

Desde este momento, la interfaz gráfica nos mostrará todo la información que se recibe/envía a la mota remota (lectura de sensores, parámetros de configuración, unidades de tiempo y temperatura). La mota remota, a su vez, realizará el muestreo de los sensores, el encendido de los Leds y gestionará todas las peticiones que reciba.

10.3. SerialForwarder

La aplicación SerialForwarder tiene como objetivo permitir a múltiples clientes comunicarse con la mota que opera como estación base. Opera desde el PC donde conectamos la mota al puerto USB y proporciona un canal de información bidireccional al software cliente que accede a dicho puerto a través de esta utilidad. Los principales objetivos de SerialForwarder son:

- Proporcionar acceso compartido a distintos clientes conectados al mismo tiempo: El acceso al puerto es un recurso limitado que sólo puede ser utilizado por una aplicación al mismo tiempo. SerialForwarder actúa de intermediario (proxy), entre la mota base y el resto de aplicaciones cliente.

- Los métodos de comunicación entre la mota y el PC anfitrión pueden cambiar con el tiempo. SerialForwarder proporciona a las aplicaciones cliente una abstracción basada en sockets para conectar con la mota. Esto simplifica en gran medida el acceso de las mismas a la mota ya que SerialForwarder es el responsable de gestionar la comunicación en ambos sentidos. La aplicación cliente puede centrarse en realizar su tarea dejando a SerialForwarder la responsabilidad de comunicar con la mota.

- Por otra parte, las aplicaciones cliente, no necesitan estar corriendo en el PC físicamente conectado a la mota. En ocasiones, las motas base pueden estar distribuidas en un amplio rango geográfico. Dado que SerialForwarder se basa en sockets, esto no supone un inconveniente porque los clientes pueden conectar desde cualquier lugar donde haya algún punto de acceso a la red a la que se conecta el PC anfitrión.

Básicamente, SerialForwarder se encarga de redirigir los paquetes de bytes de información que recibe desde el puerto hacia cada uno de los clientes conectados. Por otra parte, SerialForwarder crea un socket de servidor para escuchar a cada uno de los clientes y enviar al puerto serie, en tiempo real, el flujo de bytes que va recibiendo de cada uno de ellos.

Finalmente decir que SerialForwarder es una herramienta desarrollada en Java y utiliza el paquete javax/comm para acceder a los puertos del PC. Si estas librerías no están instaladas, el acceso al puerto serie no funcionará.