

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA
VISUALIZACION Y ALMACENAMIENTO DE DATOS DE LA PLATAFORMA
HARDWARE IEEE 802.15.4 XBEE**

DEYSY DEYSY MURILLO DIAZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA ELÉCTRICA ELECTRÓNICA Y
TELECOMINUCACIONES
OCTUBRE 2009**

**DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA
VISUALIZACION Y ALMACENAMIENTO DE DATOS DE LA PLATAFORMA
HARDWARE IEEE 802.15.4 XBEE**

DEYSY DEYSY MURILLO DIAZ

**Trabajo de investigación presentado como requisito para optar por el
título de Ingeniero Electrónico**

Director

MIE. JOSÉ DE JESUS RUGELES URIBE

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA ELÉCTRICA ELECTRÓNICA Y
TELECOMINUCACIONES
GRUPO DE INVESTIGACIÓN EN CONECTIVIDAD Y PROCESAMIENTO DE
SEÑALES (CPS)
OCTUBRE 2009**

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	1
Capítulo 1	
PLANTEAMIENTO DEL PROBLEMA	4
1.1 MARCO TEÓRICO	5
1.1.1 Estándar IEEE 802.15.4 / ZigBee	5
1.1.2 Dispositivos Xbee ZNet 2.5	9
1.2 HERRAMIENTAS HARDWARE Y SOFTWARE PARA LOS MODULOS XBEE	13
1.3 JUSTIFICACIÓN	16
Capítulo 2	
ESTUDIO DE LA PLATAFORMA DE COMUNICACIÓN	19
2.1 COMUNICACIÓN RS232	19
2.2 MODOS DE OPERACIÓN DE LOS MÓDULOS XBEE	23
2.2.1 Modo Transparente	23
2.2.2 Modo API	25
Capítulo 3	
DESARROLLO Y DESCRIPCIÓN DEL SOFTWARE	32
3.1 CONDICIONES Y REQUISITOS PARA EL BUEN FUNCIONAMIENTO DEL PROGRAMA	36
3.2 HERRAMIENTA SOFTWARE DESARROLLADA EN MATLAB	41
3.3 HERRAMIENTA SOFTWARE DESARROLLADA EN JAVA	52

Capítulo 4	
PRUEBAS Y RESULTADOS	58
4.1 PRUEBAS CON EL GENERADOR DE SEÑALES	60
4.2 PRUEBAS UTILIZANDO SENSORES DE TEMPERATURA Y HUMEDAD	70
4.3 PRUEBA CON TRES NODOS REMOTOS	82
4.4 PRUEBA CON EL ANALIZADOR DE RED	84
Capítulo 5	
CONCLUSIONES Y OBSERVACIONES	88
5.1 CONCLUSIONES	88
5.2 OBSERVACIONES	90
REFERENCIAS BIBLIOGRÁFICAS	92
ANEXOS	
ANEXO A. HOJAS DE ESPECIFICACIONES TÉCNICAS DE LOS SENSORES Y CIRCUITOS INTEGRADOS QUE HACEN PARTE DE LA PLATAFORMA HARDWARE UTILIZADA EN LAS PRUEBAS Y DE LA PLATAFORMA XBEE SENSOR ADAPTER	95
ANEXO B. PRUEBAS DE DESEMPEÑO	104
ANEXO C. DIAGRAMA DE FLUJO DEL PROGRAMA	108

LISTA DE FIGURAS

	Pág.
Figura 1.1 Modelo de capas para redes inalámbricas de sensores	7
Figura 1.2 Arquitectura del módulo Xbee	9
Figura 1.3 Diagrama de bloques del funcionamiento de transmisión y Recepción de la plataforma Xbee	12
Figura 2.1 Diagrama del sistema de flujo de datos en la UART	22
Figura 2.2 Secuencia para envío de comandos AT	24
Figura 2.3 Estructura de la trama de datos API	27
Figura 2.4 Estructura de la trama API 0x92	29
Figura 3.1 Filosofía del software implementado	33
Figura 3.2 Tarjeta de interfaz RS-232 para el montaje del módulo Xbee	37
Figura 3.3 X-CTU	38
Figura 3.4 “Modem Configuration” del XCTU	39
Figura 3.5 Interfaz grafica desarrollada en Matlab	42
Figura 3.6 Imágenes de las opciones del software	44
Figura 3.7 GUI de Configuración	45
Figura 3.8 Interfaz grafica desarrollada en Java	52
Figura 3.9 Imagen de las opciones del software	54
Figura 4.1 Imagen en el osciloscopio de la señal entregada al Xbee	60
Figura 4.2 Imagen obtenida con el software realizado en Matlab	62
Figura 4.3 Imagen en el osciloscopio de la señal entregada al Xbee	63
Figura 4.4a Imagen completa de los datos recolectados en la sesión de Trabajo	64
Figura 4.4b Imagen ampliada de la señal entregada por el generador al puerto AD2 del modulo remoto	64

Figura 4.5a Desempeño del sistema utilizando señales de prueba con frecuencias distintas, señal con frecuencia de 2,66 Hz	68
Figura 4.5b Desempeño del sistema utilizando señales de prueba con frecuencias distintas, señal con frecuencia de 0,470 Hz	69
Figura 4.6 Señales obtenidas en el prueba con el generador de señales	70
Figura 4.7 Condiciones del lugar de trabajo	72
Figura 4.8 Dispositivos utilizados en la prueba del software realizada en el salón social del grupo de investigación CPS	73
Figura 4.9 Visualización de los datos de temperatura y humedad con sus unidades correspondientes	74
Figura 4.10 Prueba de temperatura y humedad	75
Figura 4.11 Prueba sin el uso de coeficientes A y B	77
Figura 4.12 Temperatura del salón en el momento de la realización de las pruebas	78
Figura 4.13 Imágenes de la prueba	79
Figura 4.14 Imágenes de la prueba sin el uso de coeficientes A y B	81
Figura 4.15 Prueba con tres nodos	83
Figura 4.16 prueba con el analizador de red	86

LISTA DE TABLAS

	Pág.
Tabla 1.1 Especificaciones del módulo Xbee	10
Tabla 2.1 Descripción de pines de RS232	20
Tabla 2.2 Descripción de algunos comandos AT	25
Tabla 2.3 Caracteres especiales para modo de operación API con escape	26
Tabla 2.4 Nombre y valores de los identificadores ID de la trama	28
Tabla 3.1 Parámetros a configurar en el XCTU	40
Tabla 3.2 Funciones y comandos utilizados para la implementación de la interfaz en Matlab	51
Tabla 3.3 Librerías, clases y métodos utilizados para la implementación del software	56
Tabla 4.1 Resumen de pruebas realizadas para probar el funcionamiento de la herramienta	59
Tabla 4.2 Valores de frecuencia y muestras capturadas por el modulo Xbee	65
Tabla 4.3 Resumen de los resultados de las pruebas	87

TITULO
DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA VISUALIZACIÓN Y
ALMACENAMIENTO DE DATOS DE LA PLATAFORMA HARDWARE IEEE 802.15.4 XBEE*

DEYSY DEYSY MURILLO DIAZ**

Palabras claves: software, interfaz grafica, procesamiento de datos, Xbee Znet 2.5 (serie 2), IEEE 802.15.4, red de sensores inalámbricos, Java, Matlab.

La reciente convergencia entre la tecnología de sensores y de comunicaciones inalámbricas de baja potencia, han permitido el surgimiento de redes inalámbricas de sensores (WSN) para las cuales se han identificado una gran cantidad de aplicaciones. En el grupo de investigación en conectividad y procesamiento de señales (CPS) de la Universidad Industrial de Santander, se venido desarrollando investigaciones e implementaciones de técnicas de WSN con un dispositivo hardware conocido comercialmente con el nombre de Xbee ZNet 2.5 (serie 2).

Los proyectos desarrollados en el grupo CPS, tradicionalmente enfocaron sus esfuerzos en el desarrollo de herramientas hardware que utilizan módulos Xbee para recolectar datos de un ambiente o fenómeno en estudio. Dichos datos son transmitidos inalámbricamente hacia una estación base conectada a un computador para realizar el procesamiento, la presentación y el almacenamiento de estos.

El presente trabajo entrega como producto final una herramienta software que permite presentar y almacenar de forma clara y precisa la información transmitida por los módulos Xbee. La interfaz grafica desarrollada suple las necesidades del grupo de investigación CPS en cuanto a flexibilidad y presentación de información, otorgando a este último una nueva herramienta para el desarrollo de otros proyectos de investigación que hagan uso de las potencialidades de los módulos Xbee.

* Trabajo de Grado

** Facultad de Ingenierías Físico Mecánicas, Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Director: José de Jesús Rúgeles Uribe

TITLE
**DEVELOPMENT OF A SOFTWARE TOOL FOR VIEWING AND STORAGE DATA
GENERATED BY A HARDWARE PLATFORM XBEE IEEE 802.15.4***

DEYSY DEYSY MURILLO DIAZ**

Keywords: software, GUI, data processing, Xbee ZNet 2.5 (Series 2), IEEE 802.15.4, wireless sensor network, Java, Matlab.

The recent convergence of sensor technology and low-power wireless communications, have enabled the emergence of wireless sensor networks (WSN) for which it have been identified a large number of applications. In the research group in connectivity and signal processing (CPS) of the Industrial University of Santander (UIS), had being developed research and implementations of techniques of WSN with a hardware device, known commercially under the name of Xbee ZNet 2.5 (Series2).

The projects developed in the CPS group, traditionally focused their efforts on developing hardware tools that use Xbee modules to collect data from an environment or phenomenon under study. These data are transmitted wirelessly to a base station connected to a computer that allows the processing, the presentation and the storage of data of interest.

This work, give as a final product a software tool that enables to present and store in a clear and accurate manner, the information supplied by Xbee modules. The graphical user interface developed in this project, meets the needs of the CPS research group in terms of flexibility and presentation of information, giving a new tool for the development of future research projects that makes use of the potentialities of the Xbee modules.

* Degree Work.

** Faculty of physics-Mechanical Engineering, School of Electrical, Electronic and Telecommunications Engineering. Director: Jose de Jesus Rugeles Uribe.

INTRODUCCIÓN

La reciente convergencia entre la tecnología de los sensores y de la comunicación inalámbrica de baja potencia han permitido el surgimiento de las redes inalámbricas de sensores (WSN) para las cuales se han identificado una gran cantidad de aplicaciones como por ejemplo: La monitorización ambiental, la agricultura de precisión, el control de procesos, el mantenimiento de maquinaria, la domótica y los proyectos militares en donde estas redes inalámbricas de sensores, prometen generar grandes impactos.

Una WSN está conformada por un número de dispositivos sensores que están distribuidos ya sea dentro del fenómeno que se desea estudiar o muy cerca a él. En el grupo de investigación en conectividad y procesamiento de señales (CPS) de la Universidad Industrial de Santander, se cuenta con un dispositivo hardware conocido comercialmente con el nombre de Xbee ZNet 2.5 (serie 2) con el cual han venido desarrollando estudios e implementaciones de técnicas de WSN, en algunos proyectos de investigación. Los datos recolectados del fenómeno en estudio utilizando este dispositivo hardware (módulos Xbee), son transmitidos inalámbricamente hacia una estación base conectada a un computador en el cual se realiza el procesamiento de dichos datos. A pesar de que los módulos Xbee se comunican con un equipo de buena capacidad de procesamiento, como lo es el computador, la herramienta software que se tiene actualmente no posee una interfaz grafica que facilite la interpretación y posterior toma de decisión, por parte del usuario del sistema de información, además de no permitir almacenar los datos recolectados en una sesión de trabajo.

En este documento se presentan los resultados del proyecto de pregrado titulado “DESARROLLO DE UNA HERRAMIENTA SOFTWARE PARA LA

VISUALIZACION Y ALMACENAMIENTO DE DATOS DE LA PLATAFORMA HARDWARE IEEE 802.15.4 XBEE”, trabajo enmarcado dentro de la línea de investigación en redes de datos del grupo de investigación CPS perteneciente a la escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander. Este proyecto entrega como producto final una herramienta software que permite presentar de forma clara y precisa la información transmitida por los módulos Xbee y además posibilita el almacenamiento de los datos recolectados por estos dispositivos. Asimismo, esta herramienta se adapta a las necesidades del grupo de investigación CPS, debido al gran interés de este último en el desarrollo de otros proyectos que hacen uso de las potencialidades de los módulos Xbee.

A partir de un marco teórico general sobre la implementación del estándar IEEE 802.15.4, en los dispositivos utilizados para la creación de plataformas de redes de sensores, el primer capítulo busca familiarizar al lector con el problema por resolver en este proyecto, posteriormente se presenta un estado del arte donde se describen algunos proyectos de investigación a nivel mundial y a nivel local (específicamente dentro de la Universidad Industrial de Santander), los cuales han permitido el fortalecimiento de las investigaciones en el tema de las redes de sensores inalámbricos. Al final de esta primera sección se expresa de manera formal el problema planteado y su respectiva justificación.

En el segundo capítulo se muestra un estudio general de la plataforma de comunicación de los módulos Xbee Znet 2.5. Se presentan conceptos básicos relacionados con la comunicación serial y con los modos de operación de los módulos Xbee: el modo transparente y el modo API. Luego se explica de forma general, el funcionamiento de cada uno de los modos de operación, como se realiza la configuración de los parámetros del radio con comandos AT, como es la estructura de las tramas utilizadas para comunicación inalámbrica y cuáles son las

principales ventajas que presenta el uso del modo de operación API para el tipo de aplicaciones presentadas en este en este proyecto.

El tercer capítulo, presenta el desarrollo y descripción del software implementado, sus características principales, el diagrama de flujo de la lógica de programación y las condiciones que se requieren para su funcionamiento. Definido lo anterior, se presentan las herramientas software realizadas haciendo uso de las potencialidades que ofrece el entorno computacional MATLAB y el lenguaje de programación orientado a objetos JAVA.

El cuarto capítulo, presenta la descripción de las pruebas realizadas para demostrar el funcionamiento de las herramientas software desarrolladas y los resultados conseguidos con dichas pruebas.

En el quinto capítulo, se establecen las conclusiones obtenidas en el desarrollo del proyecto, además de algunas recomendaciones hechas para próximos trabajos de investigación que hagan uso de esta herramienta y que deseen continuar trabajando con esta línea de investigación.

Para finalizar, se encuentra la bibliografía empleada durante la investigación y desarrollo de este trabajo, así como los anexos que complementan la información necesaria para la comprensión del trabajo realizado.

Con la realización de este proyecto se logró desarrollar una herramienta de visualización tanto en el lenguaje de programación orientada objetos Java, como en el entorno computacional Matlab, que no solo permite almacenar y observar el comportamiento o tendencia de los datos capturados por los nodos remotos de una red conformada por dispositivos Xbee, sino que también cuentan con diversas funciones que brindan al usuario la oportunidad de interactuar con dicha herramienta y ajustarla a sus aplicaciones.

Capítulo 1.

PLANTEAMIENTO DEL PROBLEMA

Desde hace aproximadamente dos años, dentro del grupo de investigación en Conectividad y Procesamiento de Señales (CPS) de la Universidad Industrial de Santander (UIS), se han venido desarrollando estudios e implementaciones de técnicas de redes de sensores inalámbricos (WSN) en diversos campos de la ciencia y la tecnología (ejemplo: Agricultura, Energía, Telecomunicaciones, etc). La estrategia con la que se ha trabajado todo este tiempo consiste en recolectar datos del ambiente o del fenómeno en estudio y transmitirlos inalámbricamente (utilizando el estándar IEEE 802.15.4) hacia una estación base o concentrador, el cual teóricamente tiene la capacidad de realizar el procesamiento, la presentación y el almacenamiento de datos.

La aparición de plataformas más actualizadas en términos de protocolos de comunicaciones y desempeño (menor consumo de potencia eléctrica, mayor potencia en la transmisión de radio frecuencias, mayor poder de procesamiento, menor costo, entre otras) han forzado, de igual manera, la actualización de las herramientas software con las que tradicionalmente se han trabajado dentro del grupo. Actualmente en el grupo de investigación CPS se están desarrollando proyectos de pregrado y maestría utilizando un dispositivo hardware conocido comercialmente con el nombre de Xbee Znet 2.5 (Serie 2). La herramienta software utilizada conjuntamente con dicho hardware, es muy limitada, hasta el punto que simplemente actúa como un canal directo entre la comunicación inalámbrica de los datos de una red y el computador que hace las veces de estación base, algo que perfectamente puede ser realizado desde el hipertexto de cualquier sistema operativo.

Hasta el momento, los datos son presentados en el monitor de una estación base como una cadena de números hexadecimales que solo tienen sentido para quien conoce perfectamente la trama de datos en la que está contenida la información transmitida por los módulos y aun así, es complicado y dispendioso hacer el análisis de esta trama, restándole tiempo a actividades de interpretación de datos, que dependiendo de la situación, precisen de una toma de decisión urgente (ejemplo: falla en la comunicación de la red, lecturas de datos del ambiente que pueden ser peligrosas, etc). Además de no poseer una interfaz gráfica que facilite la interpretación y posterior toma de decisión, por parte del usuario del sistema de información, la herramienta software con la que se cuenta actualmente no permite almacenar los datos recolectados de una sesión de trabajo y por tanto hace que el sistema sea propenso a pérdida de información valiosa y a la realización de un esfuerzo mayor en la recuperación de esta.

A raíz de estos problemas, el grupo de investigación CPS ha decidido enfocar sus esfuerzos al desarrollo de una herramienta software que además de presentar de forma clara y precisa la información transmitida por los módulos Xbee, también posibilite el almacenamiento de datos y se adapte a las necesidades de los proyectos que se vienen desarrollando con los dispositivos mencionados anteriormente.

1.1 MARCO TEÓRICO

1.1.1 Estándar IEEE 802.15.4 / ZigBee

Existen en la actualidad un gran conjunto de estándares dedicados a transferir altas tasas de información para diferentes aplicaciones de redes. No obstante, hasta hace pocos años no existía un estándar inalámbrico dedicado a sensores y dispositivos de control. Dichos dispositivos no requieren un gran ancho de banda pero necesitan una baja latencia y bajo consumo de energía para lograr una

mayor duración de las baterías y arreglos con más número de dispositivos. Existen soluciones creadas por diferentes fabricantes para cumplir con estos requerimientos, pero esto provoca un problema de interoperabilidad entre ellas, lo que hace necesario la creación de un estándar global y abierto para redes inalámbricas de sensores y control.

El estándar IEEE 802.15.4 / ZigBee nace como una alianza tecnológica, sin fines de lucro, conformada por más de 100 empresas, la mayoría de ellas fabricantes de semiconductores, con el objeto de favorecer el desarrollo e implementación de una tecnología inalámbrica para redes a bajo costo[3]. Los principales objetivos de estas redes son: la fácil instalación, confiabilidad de transferencia de datos, bajo costo y un razonable tiempo de vida de las baterías manteniendo un protocolo simple y flexible. La arquitectura del estándar IEEE 802.15.4 / ZigBee está basado en la estructura de capas del sistema OSI, en donde cada capa es responsable por una parte del estándar y ofrece servicios a las capas superiores; las interfaces entre las capas sirven para definir los enlaces lógicos descritos en este estándar [4].

La figura 1.1 presenta un modelo de capas en donde se muestran las responsabilidades que tienen tanto ZigBee Alliance como IEEE en el desarrollo de las redes inalámbricas de sensores. Nótese que las capas en donde ZigBee Alliance trabaja son las capas de red y seguridad y están soportadas por las capas de acceso al medio y física, las cuales son tema de estudio de IEEE. Tal y como se aprecia en la figura 1.1, la inclusión de la propuesta hecha por ZigBee Alliance le confiere al estándar desarrollado por IEEE, la posibilidad de establecer redes del tipo malla las cuales no están contempladas en las directrices de este.

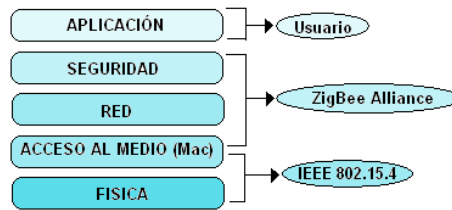


Figura 1.1 Modelo de capas para redes inalámbricas de sensores

Fuente: Adaptado por el autor de “khanh tuan le, Designing a ZigBee transceiver” [5].

El protocolo Zigbee/ IEEE 802.15.4 ofrece las siguientes características:

- Estándar para comunicación de corto alcance.
- Bajo consumo y administración de energía.
- Utiliza modulación de Secuencia Directa de Espectro Ensanchado (DSSS).
- Alcance de hasta 75 metros con velocidades de 20, 40 y 250 Kbps.
- Bandas de operación ISM6 de 2,405-2,480 GHz (16 canales), 902-928 MHz (10 canales) y 868 MHz (1 canal), aunque la mayoría de fabricantes optan por la primera, ya que puede ser usada en todo el mundo, mientras que las dos últimas sólo se aplican en Estados Unidos y Europa, respectivamente.
- Topología Maestro/Esclavo.
- Hasta 65,536 nodos por red, 255 por subred.
- Control de acceso al medio CSMA-CA.
- Incorpora algoritmos de seguridad: de encriptación y de control de acceso.
- Bajo costo de los dispositivos, instalación y mantenimiento de la red [4].

ZigBee Alliance define tres tipos diferentes de dispositivos según el papel que cumplen en la red:

Coordinador: El tipo de dispositivo más completo. Existe uno por red, independientemente de la topología utilizada. Inicia la formación de la red y selecciona la frecuencia del canal a ser usado. Es el responsable de la asociación y desasociación de dispositivos.

Router: Interconecta dispositivos separados en la topología de la red. Se asocia con el coordinador de la red o con otro router ZigBee, es capaz de enrutar mensajes entre dispositivos y soportar asociaciones. Puede actuar como terminal o dispositivo final pero no puede estar en estado de “sleep” (dormido).

Terminal o Dispositivo final: Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), puede transmitir información destinada a otros dispositivos pero no se responsabiliza de tareas de control de pérdida de paquetes. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

A su vez el estándar *IEEE 802.15.4* establece dos tipos de dispositivos clasificados según su funcionalidad:

Dispositivos de funcionalidad completa (FFD): Gracias a la memoria adicional y a la capacidad de cómputo, puede funcionar como coordinador o router, o puede ser usado en dispositivos de red que actúen de interface con los usuarios (dispositivo final).

Dispositivos de funcionalidad reducida (RFD): opera con la mínima implementación del protocolo *IEEE 802.15.4*, tiene capacidad y funcionalidad limitadas, con el objetivo de conseguir un bajo costo y una gran simplicidad. Además solo pueden asociarse a un FFD a la vez. [20].

Es posible encontrar en el mercado dispositivos de relativo bajo costo que utilicen el estándar *IEEE 802.15.4*. Sin embargo, estos requieren de herramientas hardware y software para su programación lo cual incrementan el costo en el desarrollo de aplicaciones. Para la creación de plataformas de redes de sensores, en el grupo CPS existen algunos productos, certificados (validados) por la alianza

ZigBee. Estos productos son y han sido utilizados en proyectos de investigación que buscan implementar este tipo de tecnologías en el ámbito local y nacional. Entre los dispositivos propiedad del grupo de investigación CPS podemos encontrar los módulos Xbee fabricados por la empresa Digi [18] y los módulos Telos fabricados por Crossbow [6].

1.1.2 Dispositivos X-Bee ZNet 2.5

Este dispositivo es la plataforma hardware utilizada para el desarrollo de este proyecto. Los módulos Xbee/Xbee-PRO ZNet 2.5 OEM (formalmente conocidos como serie 2 y serie 2 pro) son diseñados por la compañía estadounidense Maxstream y están diseñados para operar bajo el estándar de comunicación IEEE 802.15.4/ZigBee y cumplir con las especificaciones de bajo costo y bajo consumo de potencia de las redes inalámbricas de sensores; esta plataforma consume muy poca potencia eléctrica (alrededor de 14[mA]) en su operación y otorga una alta calidad de enlace para la entrega de datos entre nodos remotos [7]. Además, los módulos trabajan dentro de la frecuencia ISM de 2.4 GHz y son compatibles pin a pin entre ellos. La figura 1.2 muestra el modulo en su apariencia física.



Figura 1.2 Arquitectura del módulo Xbee a. Xbee b. Xbee pro c. Distribución de Pines.

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

Como se muestra en la figura 1.2 c, el módulo Xbee consta de 20 pines, entre los cuales se destacan los pines 1 (Vcc-Alimentación), 2 (Dout- Salida dato UART), 3 (Din/config- Entrada dato UART), 9 (DTR/Sleep_RQ/DIO8- Pin control de modo dormido o E/S digital), 10 (GND- tierra) y 16 (RTS/DIO6- Control de flujo del buffer

de transmisión o E/S digital); estos son necesarios para poner en funcionamiento el dispositivo, permitiéndole transmitir datos a través de radio enlaces además de transmitir datos a través de la Unidad de Recepción y Transmisión Asíncrona (UART). Los pines, como por ejemplo el 17, 18, 19 y 20, le permiten al módulo interactuar con dispositivos externos que necesiten enviar información analógica a través del aire.

En la tabla 1.1 se muestran algunas de las características básicas de estos elementos:

ESPECIFICACIÓN	RANGO DE VALORES
Alcance en ambientes cerrados	Sobre los 40 mts
Alcance en espacios abiertos (con línea de vista)	Sobre los 120 mts
Potencia de Transmisión	2 mW (+ 3dBm), con el modo <i>boots</i> habilitado 1.25mW (+1dBm), con el modo <i>boots</i> deshabilitado
Sensibilidad del Receptor	Desde -95 dBm
Velocidad de Tx. de RF	250.000 bps
Consumo de energía en Tx	40 mA con 3.3V de polarización
Consumo de corriente en Rx	40 mA con 3.3V de polarización
Velocidad de Interface Serial	1.200 a 115.200 bps
Voltaje de polarización	De 2,1 a 3,6VDC
Temperatura de Operación	Desde -40 a 85 °C
Dimensiones	2,438 x 2,761 cms.
Número de Canales	16 canales en Secuencia Directa
Opciones de Direccionamiento	PAN ID, canales y direcciones
Frecuencia de operación Banda	ISM (2.4 GHz)
Topologías de Red soportadas	Punto a punto, punto a multipunto y malla

Tabla 1.1 Especificaciones del modulo Xbee.

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

Los módulos Xbee son montados o usados comúnmente con tarjetas de interfaz serial o USB y se comunican con un computador (PC) por medio del estándar de comunicación serial asíncrona RS232, estas tarjetas de interfaz cumplen esencialmente dos funciones: proveer los voltajes de polarización necesarios para el correcto uso de los dispositivos ZigBee, y convertir los voltajes establecidos por el estándar RS232 que entrega el computador a niveles de voltaje TTL (de 3.3 VDC), tanto para la transmisión como para la recepción de datos. Estas tarjetas de interfaz serial o USB, también pueden ser utilizadas para programar los módulos XBee con un software propietario llamado X-CTU. Con este software se puede definir de una forma rápida todos los parámetros que se quiera modificar en los módulos. Los módulos Xbee son muy útiles principalmente porque requieren un reducido número de componentes externos para su correcto funcionamiento; además de utilizar un software gratis para su configuración como se menciona anteriormente.

El módulo Xbee ZNet 2.5 tiene la capacidad de interactuar con dispositivos de mayor capacidad de procesamiento de datos, como Microcontroladores y computadores mediante comunicación asíncrona a través de puerto serial. De manera general es posible comunicar los módulos Xbee con cualquier dispositivo (*host*) que contenga UART. Para realizar esta comunicación dicho módulo cuenta con *buffers* que le permiten almacenar temporalmente tanto datos recibidos por la comunicación serial entre un host y un modulo como datos provenientes del aire, resultado de la comunicación inalámbrica entre dos módulos Xbee. Los datos oriundos de la comunicación serie son almacenados en el *buffer* de recepción serie hasta que estos puedan ser procesado por el módulo. Por otro lado, el *buffer* de transmisión serial almacena todos los datos provenientes del radio enlace y posteriormente son transmitidos a la UART del dispositivo host que está haciendo uso de las características inalámbricas del Xbee. La figura 1.3 presenta en diagrama de bloques la configuración interna del módulo.

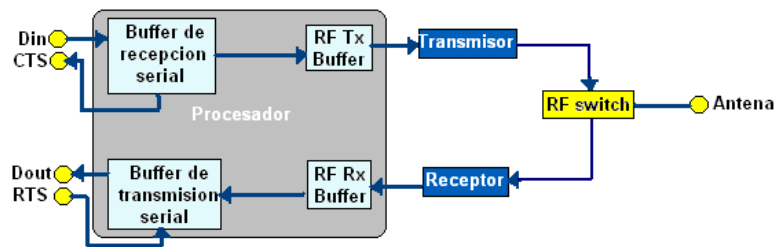


Figura 1.3 Diagrama de bloques del funcionamiento de transmisión y recepción de la plataforma Xbee.

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

El módulo soporta dos protocolos de comunicación serial denominados: modo transparente (por medio de comandos AT) y API (interfaz para la aplicación de programación o "application programming interface").

Cuando el módulo opera en modo transparente, este actúa como un camino directo entre la UART y el canal de radio comunicación, es decir que todo dato que llegue al dispositivo a través de un radio enlace, será dirigido inmediatamente a la UART de manera serial. El modo API se basa en una serie de tramas de datos creadas y organizadas para mejorar y hacer más flexible la forma en que se lleva a cabo la comunicación entre la aplicación del *Host* y el módulo Xbee[1]. Posteriormente se explicará con más detalle la importancia de estos modos de operación.

Con los Xbee serie 2 es posible crear redes con relativo grado de complejidad, como las llamadas redes enmalladas (MESH). Estas permiten acceder a un punto remoto, utilizando módulos intermedios que actúan como enrutadores (routers). El protocolo de comunicación ZigBee, permite que estos módulos automáticamente generen su propia red, sin necesidad de la intervención de un agente coordinador externo, permitiendo así la reparación de la red en caso que algún nodo falle, al mismo tiempo la red por sí sola resuelve la mejor ruta para un determinado paquete [8].

El predecesor a estos módulos es conocido como Xbee serie 1, a diferencia de los anteriormente expuestos, estos solo soportan el estándar IEEE 802.15.4, lo cual no les permite utilizar todas las ventajas que posee una red Zigbee como son por ejemplo: la formación de redes enmalladas, la asociación a una red a través de nodos intermedios y no solamente a través de un nodo coordinador, así como la reparación automática de la red, entre otras. Los Xbee serie 1 son los más apropiados para aplicaciones donde haya que implementar comunicaciones punto-punto y punto-multipunto.

1.2 HERRAMIENTAS HARDWARE Y SOFTWARE PARA LOS MODULOS XBEE.

Las posibilidades de aplicación industrial de las redes de sensores inalámbricas son amplias y por tanto pueden suplir varias necesidades empresariales. Aunque existe una gran oferta de productos en el mercado, las empresas tienen necesidades específicas que requieren de un proceso de integración tecnológica y de diseño a la medida. Lo anterior representa una oportunidad para el desarrollo de investigaciones en este tipo de aplicaciones.

El desarrollo de software para el procesamiento de datos en redes de sensores, desarrollo de interfaces hardware para diversos tipos de transductores, análisis de las alternativas de conectividad con otras redes de datos, empleo de herramientas de simulación para profundizar en temas como técnicas de enrutamiento de las redes inalámbricas, desarrollo de productos a partir de dispositivos comerciales e identificación de ventajas de plataformas de diversos fabricantes; son algunas áreas de trabajo que han sido identificadas para el fortalecimiento de las investigaciones en el tema de las redes de sensores inalámbricos durante los últimos años en la Universidad Industrial de Santander y en particular en el grupo de investigación en conectividad y procesamiento de señales (CPS), esto ha permitido el desarrollo de los siguientes proyectos de investigación:

- En el año 2007, fue realizado un trabajo de investigación a nivel de maestría en Ingeniería Electrónica titulado “Diseño, implementación y caracterización de un prototipo de red Zig Bee en ambientes abiertos y cerrados”, por José Rugeles Uribe y un proyecto de pregrado titulado “Caracterización de dispositivos de redes de sensores inalámbricos en ambientes indoor y outdoor”, por Diego Maldonado con los cuales se exploró la posibilidad de emplear esta tecnología en el desarrollo de soluciones para la industria nacional. En estos estudios se presentan conceptos relacionados con el estándar IEEE 802.15.4, la tecnología ZigBee, las herramientas de desarrollo Hardware y Software involucradas en la formación de redes malladas de sensores y un estudio del mercado de dispositivos electrónicos que emplean esta tecnología. Como legado fundamental, estos dos proyectos entregan la caracterización de dispositivos IEEE 802.15.4 y una visión sobre el fuerte impacto que puede llegar a tener esta clase de tecnología a nivel industrial [2] [9].
- En el 2008, se realizó el proyecto de pregrado titulado “Diseño e implementación de una plataforma de red de sensores inalámbricos para la monitorización de variables físicas ambientales en un galpón avícola”, por Héctor De la Hoz y Alexander Castro, con el cual se desarrolló el diseño y la implementación de una plataforma hardware compatible con dispositivos IEEE 802.15.4 Telos B y Xbee serie 2, para la monitorización y recolección de datos concernientes a las variables físicas ambientales influyentes en la mortalidad de aves de corral. Los datos obtenidos por los dispositivos Telos B fueron visualizados y almacenados utilizando un software desarrollado en el lenguaje de programación orientada a objetos JAVA, dicho software fue adaptado por los autores a las necesidades específicas del proyecto, mientras los datos obtenidos por los dispositivos Xbee serie 2 fueron procesados de forma manual ya que para estos módulos no existe una herramienta software que permita automatizar y realizar este tipo de procedimientos [1].

- En el presente año se elaboró el proyecto de pregrado titulado “Diseño, implementación y caracterización de una red de sensores Zigbee para la monitorización de contadores de energía digitales en una red de distribución de baja tensión empleando nodos Xbee”, por Luis Eduardo Muñoz y Erik Jácome Contreras, con el cual se desarrolló el diseño e implementación de un prototipo de red de sensores inalámbricos empleando nodos Xbee que permite el envío y recepción de datos desde y hacia un grupo de contadores digitales de energía, para esto realizaron un conjunto de pruebas de desempeño de los módulos inalámbricos Zigbee, además de un análisis de propagación inalámbrica empleando modelos de propagación en exteriores para ambientes urbanos [16].

En el ámbito internacional, los módulos Xbee son ampliamente utilizados para el desarrollo de aplicaciones de automatización y monitorización de ambientes abiertos y cerrados. Sin embargo el hecho de estos presentar comercialmente dos familias (Serie1 y Serie 2) ha hecho que los esfuerzos para el desarrollo de aplicaciones software sean dejadas a un lado y más bien se enfoquen en la parte del diseño hardware. Lo anterior se evidencia al momento de indagar sobre los avances desarrollados en herramientas que permitan el procesamiento de datos transmitidos por esta clase de dispositivos en cualquiera de sus dos presentaciones. A continuación se muestra el resultado de esta investigación.

- En la Universidad de Illinois se desarrolló un software para la familia Xbee serie 1 aplicado a la prevención y atención de desastres. En este, además de graficar una estadística sobre las condiciones ambientales, se alerta a los organismos de prevención de desastres (Bomberos, Policía, Defensa civil, etc.) sobre los posibles fenómenos naturales que amenacen en forma inminente a la población. Cabe resaltar que esta herramienta a pesar de realizar un buen procesamiento de datos, no los grafica y no los almacena, simplemente

funciona como sistema de alarma en las estaciones de los organismos de seguridad antes mencionados [10].

- La Universidad del estado de New York desarrolló una herramienta software para la familia Serie 1 que analiza y grafica los datos recolectados por la interacción de estos dispositivos con acelerómetros a través de los puertos ADC. Esta herramienta fue desarrollada en un lenguaje de programación dedicado a interfaces graficas llamado *processing* (basado en OpenGL) [11].

El hecho que estos avances y estudios sean dedicados para la familia de dispositivos Xbee Serie 1, imposibilita la integración de estas mismas a los proyectos de plataformas hardware desarrollados por el grupo de investigación CPS (los cuales son desarrollados con la familia de dispositivos Xbee serie 2). Lo anterior se explica porque las tramas de datos generadas por la familia Xbee serie 1 son incompatibles con las tramas generadas por la familia de dispositivos Xbee serie 2. Además de eso, el protocolo de comunicación entre los Xbee serie 2 es totalmente diferente al de los Xbee serie 1, lo que descarta la posibilidad de tener un Xbee serie 1 haciendo las veces de concentrador de datos y así poder utilizar las herramientas desarrolladas y mencionadas en el aparte anterior.

1.3 JUSTIFICACIÓN

Para enmarcar este proyecto de pregrado en un área específica del plan de estudio de la carrera de ingeniería electrónica, a continuación se presentan los resultados de una investigación previa sobre el significado de ciertos conceptos que comparten términos y que demuestran el protagonismo de este tipo de proyectos en el avance tecnológico actual.

Al observar las aplicaciones desarrolladas dentro del grupo de investigación CPS, es posible identificar que estas tienen como característica fundamental, el cumplir con la definición de sistema de información.

En general un **Sistema de Información** está descrito por tres componentes: estructura, canales y comportamiento. La estructura está compuesta por repositorios que almacenan los datos e interfaces que permiten el intercambio de información con el mundo no digital; ejemplo de repositorios e interfaces son buffers de almacenamiento y monitores respectivamente. Los canales tienen la tarea de conectar la estructura, en especial los repositorios y como ejemplo de estos pueden mencionarse los enlaces inalámbricos. Finalmente el comportamiento se traduce en servicios y mensajes; Los primeros proveen algún valor a los usuarios o a otros servicios a través del intercambio de mensajes los cuales acarrearán un contenido o significado [12][13][14].

Según el Diccionario McGraw-Hill de Ingeniería Eléctrica y Electrónica, las tecnologías de información y comunicación son “*herramientas teórico conceptuales, soportes y canales que procesan, almacenan, sintetizan, recuperan y presentan información*” [15].

Analizando estos dos conceptos, podemos observar que en sí, con estos proyectos se está desarrollando un sistema de información utilizando la base de las tecnologías de información y telecomunicaciones, lo cual demuestra la pertinencia de este tipo de proyectos en el ámbito académico de la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones de la Universidad Industrial de Santander.

Una vez definido esto, es necesario mirar la importancia de este proyecto dentro de las necesidades del grupo de investigación.

Las redes de comunicación inalámbricas, tal y como están planteadas en los modelos de los proyectos de investigación del grupo CPS, están conformadas por un número de nodos remotos, los cuales recolectan datos del medio ambiente y un nodo concentrador, el cual a su vez debe ser capaz de recolectar la información de cada nodo y procesarla. Haciendo la suposición que una red está conformada por 8 nodos remotos y que cada nodo transmite información hacia el nodo concentrador cada minuto, al término de una hora al concentrador llegarán 240 tramas y suponiendo que cada trama tiene 20 bytes, esto significaría el análisis de 4800 bytes en una hora; si se parte del hecho que las redes inalámbricas de sensores están diseñadas para permanecer en funcionamiento durante meses o inclusive años, el no tener una herramienta que permita la automatización de la lectura interpretación y almacenamiento de datos sería un hecho gravísimo que comprometería la viabilidad de la implementación de un sistema (como el descrito anteriormente) de una forma eficiente.

Cabe mencionar que a pesar de tener la posibilidad de comunicar los módulos Xbee con un equipo de buena capacidad de procesamiento (como es el caso de un computador), no existe una herramienta software que permita agilizar el proceso de interpretación de los datos provenientes de estos módulos (hasta ahora este proceso ha venido siendo realizado en forma manual). Por tanto el desarrollo de una herramienta para la automatización del análisis de los datos en un computador supliría esta necesidad y además permitiría la detección y la solución de posibles complicaciones en aplicaciones de redes de sensores inalámbricos cuya plataforma hardware esté compuesta por dispositivos Xbee. Esta herramienta software se realizó en el lenguaje de programación orientada a objetos Java y en el entorno computacional Matlab ya que ambas herramientas cuentan con librerías que permiten interactuar con el puerto serial de un PC.

Capítulo 2.

ESTUDIO DE LA PLATAFORMA DE COMUNICACIÓN

En este capítulo se presenta información general sobre las capacidades de comunicación de la plataforma hardware Xbee Znet 2.5. En las redes de sensores inalámbricos existen dos tipos de comunicaciones: Comunicación de módulos remotos con un modulo coordinador y comunicación del modulo coordinador con el PC. Los datos recolectados por los módulos remotos son transmitidos al modulo coordinador vía RF en 2.4 GHz utilizando el estándar IEEE 802.15.4 y el coordinador a su vez se comunica con el PC a través del puerto serial. A continuación se profundizan las formas de comunicación anteriormente descritas.

2.1 COMUNICACIÓN RS232

Debido a la gran variedad de equipos, sistemas y protocolos que existen y ante la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre sí, surge el estándar de comunicación RS232. La Asociación de Industrias Electrónicas (**EIA**) fue la encargada de elaborar la norma RS-232, la cual define la interface mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial. El estándar RS232 fue diseñado inicialmente para utilizar conectores de 25 pines, sin embargo luego fue modificado y actualizado para trabajar con conectores de 9 pines. A pesar que la versión de 25 pines permite tener mucha más información sobre la transferencia de datos, la más utilizada es la versión de 9. La Tabla 2.1 muestra una descripción de cada uno de los pines.

Conector 25 pines	Conector 9 pines	Nombre	Descripción	Tipo de señal e/s
1	1	-	Masa chasis	-
2	3	TxD	Trasmitir Datos	S
3	2	RxD	Recibir Datos	E
4	7	RTS	Solicitud de envío	S
5	8	CTS	Libre para envío	E
6	6	DSR	Equipo de datos listos	E
7	5	SG	Tierra	-
8	1	DCD	Detección de portadora	E
15	-	TxC	Reloj de Transmisión	S
17	-	RxC	Reloj de recepción	E
20	4	DTR	Terminal de Datos listos	S
22	9	RI	Indicador de llamada	E
24	-	RTxC	Reloj de Trans/Recep	S

Tabla 2.1 Descripción de pines de RS232

Fuente: Adaptado por el autor de Murcia Barba Jose Manuel, "Puerto serie RS232". ITIG.

RxD y TxD son los pines que portan los datos, los demás se encargan de otros trabajos como por ejemplo, el DTR indica que el ordenador esta encendido, DSR que el dispositivo conectado al puerto esta encendido, RTS que el ordenador puede recibir datos al no estar ocupado, mientras que CTS informa que el dispositivo puede recibir datos, DCD detecta que existe presencia de datos. Finalmente, existen otras señales poco comunes que no es necesario explicar en este documento por rebasar el alcance del mismo.

El puerto RS232 es el puerto serial de los computadores, y una de las ventajas de este puerto es que permite comunicaciones entre diferentes dispositivos tales como otro computador, el mouse, impresora y para este caso con los dispositivos

Xbee. Antes de iniciar cualquier comunicación el usuario debe configurar ambos extremos del enlace, estableciendo parámetros como: número de bits, paridad, bit de parada, Velocidad del puerto y protocolo de control de flujo (RTS/CTS o XON/XOFF)[17].

Existen dos tipos de comunicaciones seriales: síncrona y asíncrona.

En la comunicación serial síncrona además de utilizar una sola línea para la transmisión y recepción de datos se necesita de una línea adicional la cual contendrá los pulsos de reloj que indicaran cuando un dato es válido. Por otra parte, en la comunicación serial asíncrona no son necesarios los pulsos de reloj. La duración de cada bit está determinada por el parámetro de velocidad de puerto configurado antes de iniciar la transferencia de datos.

Como se ha mencionado anteriormente, el estándar RS232 define la manera en que se debe dar el intercambio serial de datos binarios entre un equipo terminal de datos (DTE) y un equipo de comunicación de datos (DCE). En el caso particular de los módulos Xbee, estos se comunican con un computador por medio del estándar de comunicación serial asíncrona RS232.

Los datos provenientes de un host (como lo puede ser un computador) están compuestos por 8 bits y entran a la UART del modulo a través del pin 3 (DIN). La forma en que están organizados estos bits se muestra en la figura 2.1 a. En esta figura se aprecia que el dato de ocho bits está precedido por un bit de inicio y sucedido con un bit de parada, además como se dijo anteriormente la duración de cada símbolo está determinada por la velocidad a la cual fue configurada la velocidad del puerto.

Las tareas de control de flujo y comprobación de paridad son desarrolladas por la UART, la cual a su vez tiene buffers de almacenamiento de datos, los cuales los

retienen hasta que estos sean debidamente procesados². En la figura 2.1 b, se muestra simbólicamente el flujo de datos entre la comunicación de un módulo Xbee y un host (microprocesador). Nótese la intervención de las líneas CTS y RTC encargadas del control de flujo mientras que las línea Din y Dout están encargadas de la transmisión y recepción de datos.

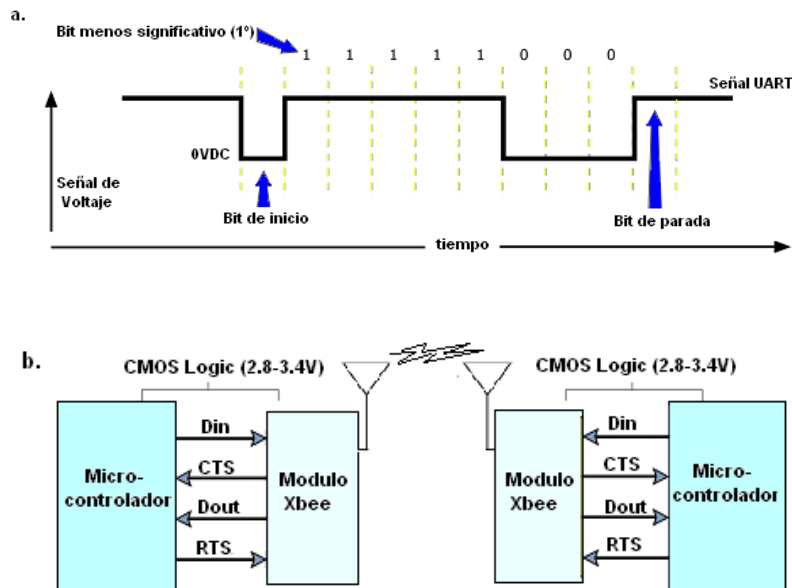


Figura 2.1 a. Paquete de datos de la UART b. Diagrama del sistema de flujo de datos en la UART

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

Las tarjetas de circuitos impresos diseñadas especialmente para realizar este tipo de comunicación con los módulos Xbee y con las que cuenta el grupo de investigación CPS, tienen un dispositivo conversor de datos seriales, con niveles lógicos de tensión CMOS (2.8V, 3.4V), a niveles de comunicación RS232 (-12V, 12V); de tal modo que los datos transmitidos por los nodos Xbee llegan al PC utilizando los niveles de tensión establecidos por dicho estándar.

² Referirse al capítulo 1.

2.2 MODOS DE OPERACIÓN DE LOS MODULOS XBEE

Los módulos XBEE pueden trabajar en dos diferentes modos de interfaz serial: modo API y modo transparente.

2.2.1 Modo Transparente

Los módulos Xbee por defecto trabajan en este modo. Al operar en modo transparente los módulos actúan como un reemplazo de la línea serial, esto es: Los datos transmitidos por el host y recibidos a través del pin DI (*DataIn*) de la UART, son puestos en espera para la transmisión RF. Así mismo, cuando un dato RF es recibido, este es transmitido hacia el host a través del pin DO (*DataOut*) de la UART. Cuando el dato recibido desde el host no puede ser transmitido inmediatamente, este es almacenado en el búfer de recepción hasta que pueda ser enviado a través del radio enlace. Los datos son empaquetados en tramas y enviados cada vez que el tiempo de empaquetamiento de tramas (parámetro RO) se cumpla o cuando el módulo ha recibido 72 bytes (máximo tamaño de paquetización). Si el búfer de recepción empieza a llenarse, se debe implementar algún tipo de control de flujo ya sea vía hardware o vía software, esto con el fin de prevenir pérdidas de datos entre el host y el dispositivo (un sobre flujo) [7] [21].

Para modificar o leer los parámetros del módulo Xbee cuando se trabaja en modo transparente se utilizan comandos AT, con el uso de estos comandos se pueden cambiar diferentes parámetros como por ejemplo la paridad, la cantidad de bits de parada, el canal en que este trabajando el modulo y la velocidad de los baudios de la transmisión serial. Como se ha mencionado anteriormente la empresa DIGI ofrece un software gratis llamado X-CTU el cual permite de una forma sencilla emitir comandos AT y cambiar parámetros del radio.

Con el fin de preparar el modulo Xbee para la recepción de comandos AT es necesario transmitirle una secuencia especifica de caracteres que son

identificados por el modulo como una secuencia de entrada o código de escape al modo comandos, por defecto esta secuencia es “+++”. Esta secuencia de escape debe escribirse en un periodo delimitado de tiempo, el cual está determinado por el parámetro GT establecido por defecto en un segundo. En conclusión, para entrar al modo de comandos se debe escribir la secuencia especial “+++” evitando consumir en este proceso más de 2 segundos.

Una vez ingresado al modo comandos el modulo Xbee responderá enviando los caracteres “OK” como código de resultado. Si no lo hace, puede ser posible que no se haya respetado el tiempo GT o se haya recibido incorrectamente la secuencia de entrada. Una vez recibido el OK, se puede modificar o leer cualquier parámetro del radio antecedido por los caracteres AT. Para enviar comandos AT y parámetros de configuración se usa la sintaxis mostrada en la figura 2.2.



Figura 2.2 Secuencia para envío de comandos AT.

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

En esta secuencia primero se escribe el prefijo “AT”, a continuación el parámetro que se desea modificar (en formato ASCII)³, seguido del valor que se quiera asignar a dicho parámetro de configuración, este valor debe estar escrito en formato Hexadecimal. Finalmente se debe pulsar la tecla “enter” o retorno de carro, el cual indica al módulo que debe devolver una respuesta de confirmación de haber recibido la secuencia mencionada anteriormente. Para grabar los cambios realizados en el modulo Xbee, se escribe el comando ATWR⁴, de lo contrario, el modulo no grabara en la memoria permanente los cambios realizados

³ Si desea conocer la lista completa de comando AT referir se al manual de usuario Xbee Znet 2.5

y en una nueva sesión de trabajo no poseera las características modificadas. Es importante recordar que si no se provee un retorno de carro, el modulo ignora los comandos ingresados. Si lo que se busca es leer el valor de un parámetro determinado no se debe escribir el valor hexadecimal de la secuencia descrita anteriormente [21].

La tabla 2.2 muestra algunos ejemplos de parámetros AT.

COMANDO AT	DESCRIPCION
ATRE	Restaura los parámetros de los módulos a los valores por defecto.
ATCH	Define o lee el # del canal usado para transmitir y recibir datos entre módulos RF
ATNB	Parida serial
ATSC	Encuesta cuales canales están disponibles para conexión
ATRO	Tiempo para empaquetamiento de tramas
ATD1	Configuración de la entrada/salida AD1/DIO1
ATGT	Configuración del tiempo de silencio

Tabla 2.2 Descripción de algunos comandos AT.

Fuente: autor.

Para salir de modo de comando se ingresa la cadena ATCN, el modulo regresará un OK como respuesta si no hubo problema. Otra forma de salir del modo comando se presenta si el modulo después de cierto tiempo no recibe más caracteres, por defecto el tiempo de espera es de 3 segundos, transcurrido este tiempo, los caracteres siguientes no los tomará como comandos, sino como caracteres que transmitirá de forma inalámbrica.

2.2.2 Modo API

El modo de operación API (por sus siglas en ingles “*application programming interface*”) es un modo alternativo al modo de operación transparente. Cuando el dispositivo trabaja en modo API, todos los datos que ingresan y abandonan el

módulo están contenidos en tramas que definen las operaciones o eventos, dentro del módulo. El modo de operación API facilita muchas operaciones, tales como: La transmisión de datos hacia múltiples destinatarios (sin ingresar al modo *Comando*), estatus de recepción exitosa o fallida para cada uno de los paquetes RF transmitidos y la identificación de las direcciones fuente de cada paquete recibido [21][7].

En API, a su vez existen dos modos de operación los cuales se configuran mediante el comando AP. Estos modos de trabajo son: modo "normal" ($AP=1$) y modo "con escape" ($AP=2$). En el modo "normal", la presencia de un carácter 0x7E indica el inicio de una trama sólo si es el primer carácter que se recibe luego de un silencio. El modo "con escape" es necesario sólo cuando se envían y se reciben bytes que interfieren con la estructura de la trama, en este modo siempre se indica el inicio de una trama (0x7E) para luego detectar las secuencias especiales de escape que se muestran en la tabla 2.3. Si lo que se desea es enviar una trama con caracteres especiales lo que este modo hace es insertar un carácter de escape y realizar otra operación sobre el byte de interferencia. Esto hace que la trama sea más grande, al agregar bytes, pero evita que la cabecera de la trama se confunda con los datos enviados. Por otro lado cuando una trama con caracteres especiales se recibe en modo API con escape, se detectan las secuencias de escape especiales y se reemplazan por los caracteres originales [22] [7] [24].

CARÁCTER DE ESCAPE	DESCRIPCION
0x7E	Delimitador de la trama.
0x7D	Escape.
0x11	XON
0x13	XOFF

Tabla 2.3 Caracteres especiales para modo de operación API con escape

Fuente: Adaptado por el autor del manual de usuario del Xbee Znet 2.5

La figura 2.3 muestra estructura general API que utilizan los módulos Xbee para la comunicación en este modo de operación.

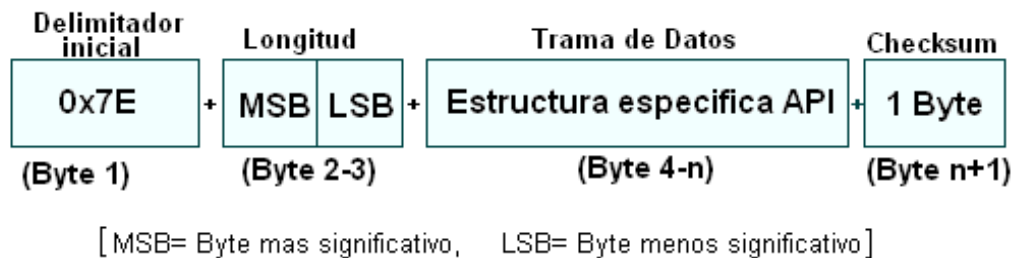


Figura 2.3 Estructura de la trama de datos API [7].

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

El inicio de la trama siempre es el carácter 0x7E, seguido de dos bytes que indican la cantidad de datos (LEN) contenidos en el resto de la trama (exceptuando el checksum), es decir, los bytes del campo trama de datos. Luego sigue, un byte adicional conocido como identificador ID el cual define la estructura específica API dependiendo del tipo de información contenida en el campo trama de datos de esta. Para terminar, la trama contiene un byte que permite verificar que la información que se recibió o se envió es correcta, este byte se llama checksum.

En la tabla 2.4 se muestran los diferentes Identificadores de la trama API que soportan los módulos Xbee [7].

NOMBRE DE LA TRAMA API	VALOR
Estado de modem	0x8A
Comandos AT	0x08
Comandos AT-Valor del parámetro de cola	0x09
Respuesta de comandos AT	0x88
Solicitud de comando remoto	0x17
Respuesta de comando remoto	0x97
Solicitud de transmisión ZigBee	0x10
Direccionamiento explícito en la trama ZigBee	0x11
Estado de la transmisión ZigBee	0x8B
Paquete recibido - ZigBee	0x90
Indicador explícito de Rx - ZigBee (AO=1)	0x91
Indicador Rx de los datos IO - ZigBee	0x92
Indicador de lectura -sensores XBee (AO=0)	0x94
Indicador del nodo de identificación (AO=0)	0x95

Tabla 2.4 Nombre y valores de los identificadores ID de la trama

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

En este Proyecto la trama API utilizada lleva el identificador ID 0x92. Como se muestra en la tabla 2.4, esta trama lleva muestras analógicas y digitales, por lo tanto cuando un modulo XBee recibe muestras IO desde un dispositivo remoto, este envía dichos datos fuera de la UART utilizando este tipo de trama.

La estructura de la trama API 0x92 se muestra en la figura 2.4. El campo datos contenido en esta trama, tiene la distribución que se presenta a continuación:

- Bytes del 5-12: Dirección de 64 bit dividida en bytes comenzando con el bit más significativo y terminando con el menos significativo (MSB first).
- Bytes del 13-14: Dirección de red de 16 bit dividida en MSB y LSB.

- Byte 15: Opción de Recepción. Si es un paquete reconocido este campo tiene un valor de 0x01 mientras que si es un paquete recibido por difusión (broadcast) este campo tiene el valor de 0x02.
- Byte 16: Numero de muestras incluidas en el mensaje. Siempre se establece en 1.
- Bytes del 17-18: Mascara Digital. Campo de mascara de bits que indica que líneas de entrada/salida digitales están activadas.
- Byte 19: Mascara Analógica. Campo de mascara de bits que indica que líneas de entrada/salida analógicas están activadas.
- Byte del 20-21 (Opcional): Muestras digitales. Si el conjunto de las muestras incluye líneas E/S digitales, estos dos bytes contienen las muestras de todas las entradas digitales habilitadas. Las líneas que no tienen habilitado el muestreo retornan 0.
- Byte del 22- 29: Muestras analógicas. Si el mensaje incluye estas líneas de entradas analógicas, cada entrada analógica habilitada devuelve un valor de dos bytes correspondiente a la medición de esta entrada. Las muestras analógicas están ordenadas secuencialmente de AD-0 a AD-3 [7].

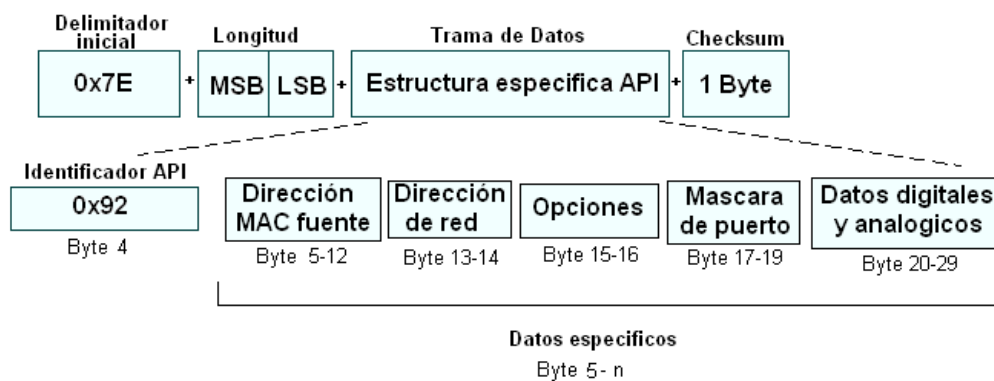


Figura 2.4 Estructura de la trama API 0x92.

Fuente: Adaptado por el autor de Digi, hoja de datos "Xbee ZNet 2.5".

El trabajar con el modo API presenta algunas ventajas frente al modo transparente. Si simplemente se desea crear enlaces punto a punto, los módulos Xbee funcionan muy bien en el modo transparente sin necesidad de realizar mucha programación. Sin embargo, si el objetivo es construir una red compuesta por más de dos dispositivos, el modo AT se vuelve difícil de soportar, gastando tiempo mientras se realiza el proceso de entrar y salir al modo comando a la vez que consume energía de las baterías de los módulos Xbee durante este procedimiento.

A continuación se mencionan otras ventajas que presenta el modo API sobre el modo transparente:

- ✓ Los datos son enviados en tramas con formatos especiales que permite identificar características específicas e importantes como por ejemplo la dirección origen de los paquetes recibidos, longitud de los datos, anuncio sobre si un paquete fue entregado con éxito (ACK), tipo de datos contenidos en la trama, etc.
- ✓ Permite la comunicación de difusión y de comunicación entre varios módulos Xbee.
- ✓ Obtener el RSSI (potencia de la señal).
- ✓ Solicitud de forma remota del estado de los pines del módulo Xbee como información ó para realizar configuraciones como por ejemplo cambiar el nivel de salida de los pines.
- ✓ Con el modo API se utiliza un firmware que permite configuraciones extras y control en ambos lados del enlace (coordinador y dispositivos remotos) [7] [23] [25].

En este proyecto se aprovechan estas características especiales del modo de operación API. Además el estudio de la plataforma de comunicación de los módulos Xbee y en especial el conocimiento de la estructura de la trama API, permite identificar parámetros relevantes para la creación e implementación del software, independientemente del lenguaje de programación que se utilice.

La trama API con que trabajan los módulos Xbee es una trama desarrollada y ajustada por la empresa Digi para la capa de aplicación del protocolo de comunicación que define el estándar IEEE 802.15.4. En el capítulo correspondiente a “Pruebas y resultados”, se muestra una prueba realizada con el *Chipcon Packet Sniffer*, el cual permite analizar la estructura de la trama Zigbee y comparar sus características con la trama API (Digi) utilizada en el desarrollo del software.

Capítulo 3.

DESARROLLO Y DESCRIPCIÓN DEL SOFTWARE

El objetivo principal de este proyecto consiste en el desarrollo de una herramienta software (una interfaz Hombre-Máquina) que permita visualizar y almacenar datos, provenientes de la comunicación entre plataformas hardware IEEE 802.15.4 X-Bee ZNet 2.5, en un archivo de datos localizado en el computador que alberga la aplicación. En este capítulo se presenta el diseño y el resultado de la implementación de la interfaz grafica (GUI) desarrollada haciendo uso de las potencialidades que ofrece el entorno computacional Matlab y el lenguaje de programación orientado a objetos Java.

Como se plasma en capítulos anteriores, los datos medidos por los puertos ADC de los radios Xbee en ubicación remota son recolectados por un modulo coordinador. La comunicación entre un modulo remoto y un coordinador se realiza inalámbricamente a través de los protocolos establecidos por el estándar IEEE 802.15.4. El módulo coordinador, recibe los datos y los almacena en su UART para luego enviarlos al PC a través de comunicación serial RS232. De manera general es posible comunicar los módulos Xbee con cualquier dispositivo (*host*) que contenga UART. Los módulos Xbee cuentan con *buffers*, que les permiten almacenar temporalmente datos recibidos por comunicación serial con el PC y/o comunicación inalámbrica con otros módulos Xbee⁴.

La comunicación entre dos módulos Xbee se caracteriza por incluir en el campo correspondiente a la capa de aplicación del datagrama de transmisión, una estructura de datos conocida como trama API. Esta trama es implementada por la

⁴ Referirse al capítulo 1.

empresa fabricante del dispositivo para facilitar el procesamiento de los datos recolectados por el modulo concentrador⁵. El programa realizado “lee”, a través del puerto serial de un PC, estas tramas API y analiza su estructura byte a byte. La trama API utilizada para recolección de los datos medidos por los ADC de los módulos Xbee se diferencia de otras tramas porque en la posición correspondiente al cuarto byte, tiene asignado el valor 0x92 y permite que para cualquier trama de esta clase la posición de los datos de cada uno de los puertos ADC no varíe⁶. Lo anterior fue aprovechado para el desarrollo del programa, dado que la información o los datos necesarios, tienen posiciones fijas [7].

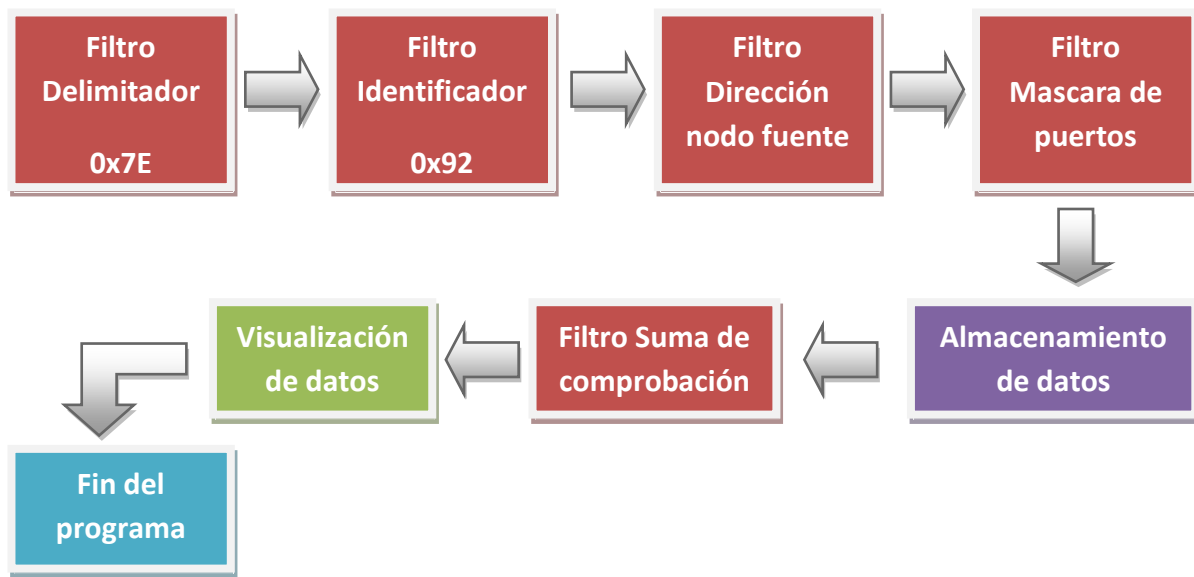


Figura 3.1 Filosofía del software implementado.

Fuente: Autor.

En la figura 3.1 se presenta la filosofía utilizada para el desarrollo del software. Como se puede apreciar el software está basado en un serie de filtros (en total cinco filtros) que clasifican las tramas en aptas y no aptas para el software.

⁵ Referirse al capítulo 2.

⁶ La estructura de la trama se plasma en el capítulo 2, en la sección 2.2.2

El primer filtro se encarga de decantar o aceptar tramas validas, esto se logra gracias a que toda trama recibida por el puerto serial debe comenzar con el valor hexadecimal 0x7E (126 en valor decimal). Este byte (el primero) es conocido como delimitador.

El segundo filtro determina si la trama lleva o no datos relacionados a los puertos analógicos digitales. Como se ha mencionado, el programa revisa si el valor del cuarto byte de la trama tiene el valor hexadecimal 0x92 (equivalente a 146 en el sistema decimal). Este byte (el cuarto) es conocido como identificador de trama. Es importante anotar que el segundo y tercer byte de la trama corresponde a la longitud de esta y que solo es utilizado a manera de información pero sin ninguna repercusión en la clasificación de las tramas, ya que se supone que el sistema solo le interesa saber si la trama que llegó contiene o no muestras de los ADC.

El tercer filtro toma los bytes que se encuentran en las posiciones de la cinco (5) a la doce (12), para determinar el nodo remoto que originó la trama en estudio. Esto fue aprovechado para darle al software la capacidad de manejar datos provenientes de diferentes nodos. Como característica importante, el software fue implementado suponiendo que la red tendrá un tamaño pequeño (máximo tres nodos remotos y un nodo concentrador), sin embargo basta con añadir unas cuantas líneas de código (en verdad repetir algunas línea de código) para permitirle a la aplicación manejar datos originarios de mas nodos.

Una vez determinado (a través de la dirección) el nodo al cual pertenecen los datos, son analizados los siguientes siete bytes. Estos bytes a pesar de no llevar una información útil para el software, llevan información relacionada con la cantidad de puertos habilitados, lo cual es una constante (cuatro puertos ADC y tres entradas digitales), estos bytes son conocidos como patrón o mascara de puertos. El filtro toma dicha mascara y la compara con los siete bytes siguientes a la dirección, si existe alguna incongruencia rechaza la trama. Por lo explicado anteriormente, es de suma importancia para el buen funcionamiento del software,

que los módulos remotos tengan habilitados sus cuatro entradas analógicas y las entradas digitales cuatro (4), once (11) y doce (12).

Hasta aquí la trama recibida debe haber pasado por los cuatro filtros anteriormente descritos. Si la trama es aceptada por todos los filtros quiere decir que los ocho bytes que suceden a la máscara de puertos pertenecen a los datos que el software debe graficar, es decir, esos ocho bytes representan las medidas tomadas por los ADC del nodo que origino la trama (nodo fuente).

El último filtro, es el filtro de suma de comprobación. Hasta el momento se ha dado por sentado que todos los bytes analizados son correctos, pero si la suma de comprobación o “*checksum*” falla, quiere decir que durante la transmisión de los datos hubo alguna perturbación y por lo tanto los datos no son correctos.

Si al final de todo el proceso la trama recibida pasa todos los filtros, los datos almacenados en el proceso descrito entre los filtros cuatro y cinco, son graficados en ventanas de visualización. Este proceso debe hacerse con cada una de las tramas recibidas y son graficadas secuencialmente respetando el orden de llegada.

En el Anexo C, se muestra el diagrama de flujos que representa la lógica del programa. El haber sido plasmado solo la lógica del programa, significa que el diagrama no describe características propias de los lenguajes de programación en los que este fue implementado (Java y Matlab), así como tampoco especifica detalles relacionados con herramientas adicionales habilitadas por estos e implementadas para otorgar al usuario una mayor interactividad con la interfaz final.

3.1 CONDICIONES Y REQUISITOS PARA EL BUEN FUNCIONAMIENTO DEL PROGRAMA.

Para la ejecución de la aplicación desarrollada debe tenerse en cuenta ciertas condiciones y requisitos relacionados con los módulos Xbee y su funcionamiento en red. Como se ha indicado en capítulos anteriores⁷, para el establecimiento de una red de sensores inalámbricos es necesaria la existencia de un nodo coordinador. El nodo coordinador es un módulo Xbee programado con el firmware “ZNET 2.5 COORDINATOR API” y conectado directamente al PC. Este se encarga de entregar a dicho computador, los datos recolectados provenientes de los nodos remotos de la red.

El trabajo del nodo coordinador en la red, consiste en apropiarse de un canal libre en la banda ISM de 2.4 GHz y asignar en este una dirección. Por lo tanto todos los nodos remotos pertenecientes a esa misma red, deben estar configurados para comunicarse con el nodo coordinador utilizando el canal y la dirección establecidos por dicho coordinador. Esta dirección es conocida como dirección de red.

Los nodos remotos que conforman la red, son módulos Xbee programados con el firmware “ZNET 2.5 ROUTER/END DEVICE AT”. Para el buen funcionamiento del software, en los nodos remotos deben estar habilitadas todas las entradas analógicas ADC (pines 17, 18, 19 y 20) además las entradas digitales 4, 11 y 12 (pines 11, 7 y 4 respectivamente). También es necesario que se establezca el periodo de muestreo de estas entradas el cual determina cada cuanto tiempo son enviados datos desde el nodo remoto al nodo coordinador.

⁷ Referirse al capítulo 2.

Para facilitar la comunicación entre el PC y el módulo coordinador, dentro del kit de desarrollo de los módulos Xbee se cuenta con una tarjeta de interfaz de referencia XBIB-R/U. Esta tarjeta viene en dos presentaciones: una tarjeta con conector DB9 (tradicional en el uso de comunicaciones en el estándar RS232) y otra que permite utilizar conectores USB. Esta segunda presentación utiliza el chip de referencia FT232-R, el cual requiere de unos drivers para su utilización. Estos drivers pueden ser adquiridos, descargándolos de la página web de la empresa FTDI (Solo existen drivers para windows XP) [26].

La figura 3.2 muestra una imagen de la tarjeta de interfaz RS-232 para el montaje del módulo Xbee.

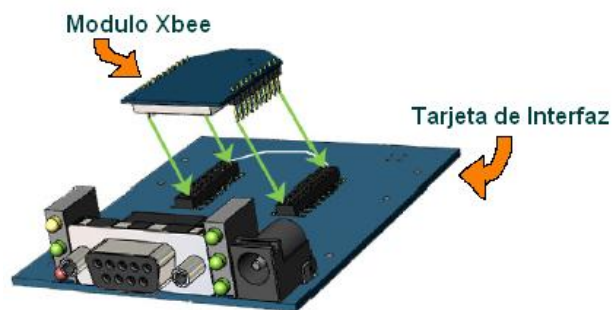


Figura 3.2 Tarjeta de interfaz RS-232 para el montaje del módulo Xbee.

Fuente: Adaptada por el autor del Manual de usuario de los módulos Xbee [7].

Con el modulo coordinador conectado al PC y utilizando cualquiera de las dos presentaciones de la tarjeta de interfaz, es posible utilizar el software propietario llamado XCTU para cambiar los firmware y configurar los parámetros del modulo conectado. La figura 3.3, muestra la ventana principal del XCTU.

El XCTU tiene varios menús (pestañas) que son utilizadas dependiendo de la aplicación que se desee, estas pestañas son: PC Settings, Range Test, Terminal y Modem configuration.

PC Settings está dividida en dos áreas, un área en donde se muestra el puerto de comunicación (COM) al cual está conectado el modulo y otra en donde se muestran opciones de configuración de este puerto (entre las que se encuentra *Baute rate* la cual define la velocidad del puerto). En esta pestaña también existe un botón que permite verificar si el modulo y el puerto se encuentran bien configurados para la comunicación, esta opción se llama **Test Query**.

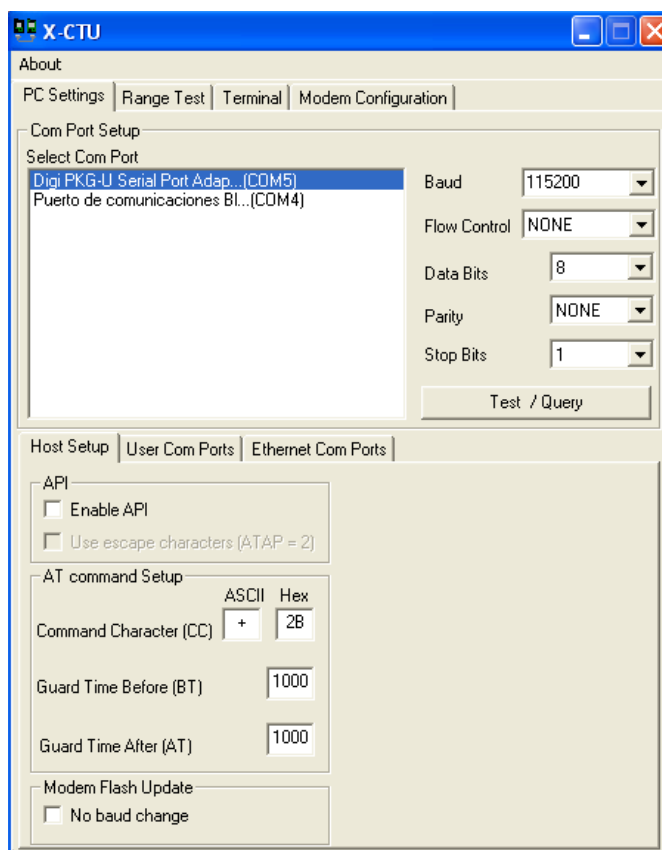


Figura 3.3 X-CTU

Fuente: Adaptado por el autor de Manual del Xbee [28]

Las pestañas **Range Test** y **Terminal** esta por fuera de los objetivos de esa investigación, por lo tanto no se realiza ninguna explicación al respecto.

Por último se encuentra la pestaña **Modem Configuration**. Esta pestaña permite, con el uso del botón *Read* y *Write*, la lectura y modificación de parámetros y

firmware del módulo conectado al PC. Además presenta una opción para realizar configuración remota (*remote configuration*), con la cual el coordinador es capaz de modificar parámetros de un nodo remoto sin necesidad de establecer una conexión física entre el nodo remoto y el PC.

En la figura 3.4 se puede observar la ventana del XCTU con la opción *Modem configuration* seleccionada.

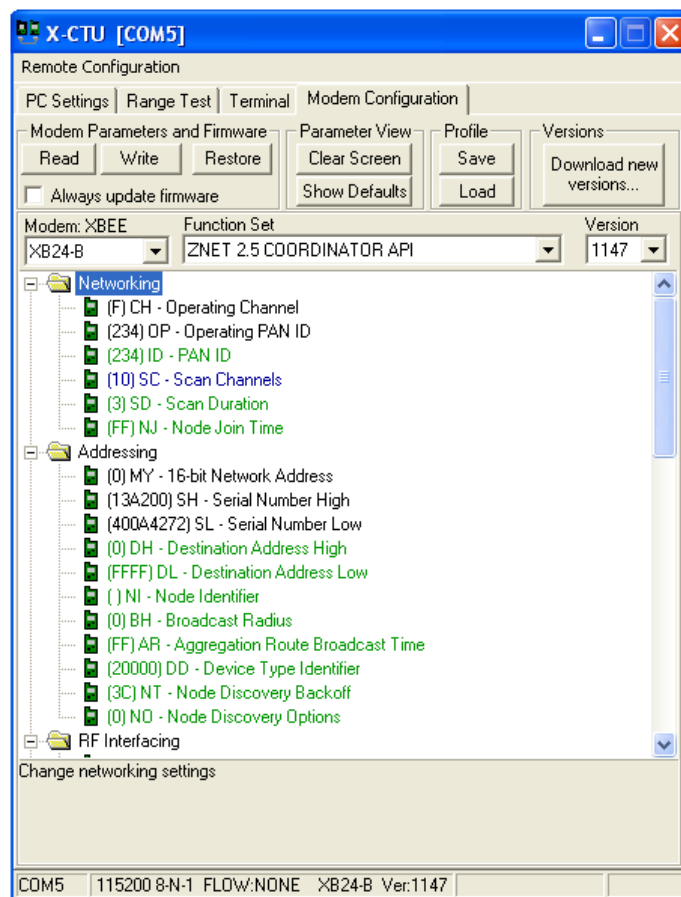


Figura 3.4 “Modem Configuration” del X-CTU

Fuente: Adaptado por el autor de Manual de Xbee [28]

A continuación se presenta la tabla 3.1, en la cual se muestran parámetros que deben ser modificados en los módulos Xbee (tanto en coordinador como en nodos

remotos) para el correcto funcionamiento del software de visualización desarrollado⁸.

DISPOSITIVO	PARAMETRO	DESCRIPCION
Coordinador	Scan channels (SC)	Posible canal de frecuencia en el que va a operar la red
	Baute rate (BD)	De acuerdo con el baute rate que tenga el puerto del PC
	# de Bit por dato	En este caso los datos están conformados de 8 bits
	Paridad (NB)	Ninguna
	Bit de parada	1
Router	SC	Igual al canal del coordinador
	D0,D1,D2,D3	Habilitar entradas analógicas
	D4, P1, P2	Habilitar entradas digitales
	IR	Tiempo de muestreo

Tabla 3.1 Parámetros a configurar en el XCTU

Fuente: Autor.

Como el software tiene la capacidad de visualizar datos recolectados por varios nodos remotos, es necesario conectar a la misma entrada ADC de cada uno de los nodos remotos la misma variable de instrumentación. Esto último con el fin de visualizar los datos utilizando un eje vertical escalado con la unidad correspondiente a la variable medida. Es decir, que si se va a medir temperatura en una red de tres nodos remotos y se establece para uno de los nodos que el sensor de temperatura (o termómetro) debe estar conectado a la entrada AD1, se debe utilizar la entrada AD1 en los dos nodos restantes para conectar el respectivo termómetro. Esto se presenta porque el software muestra la grafica de los datos agrupados según la entrada analógica de la cual provengan, entonces si

⁸ Para más información sobre los parámetros de configuración, referirse al manual de usuario del modulo Xbee Znet 2.5 [7].

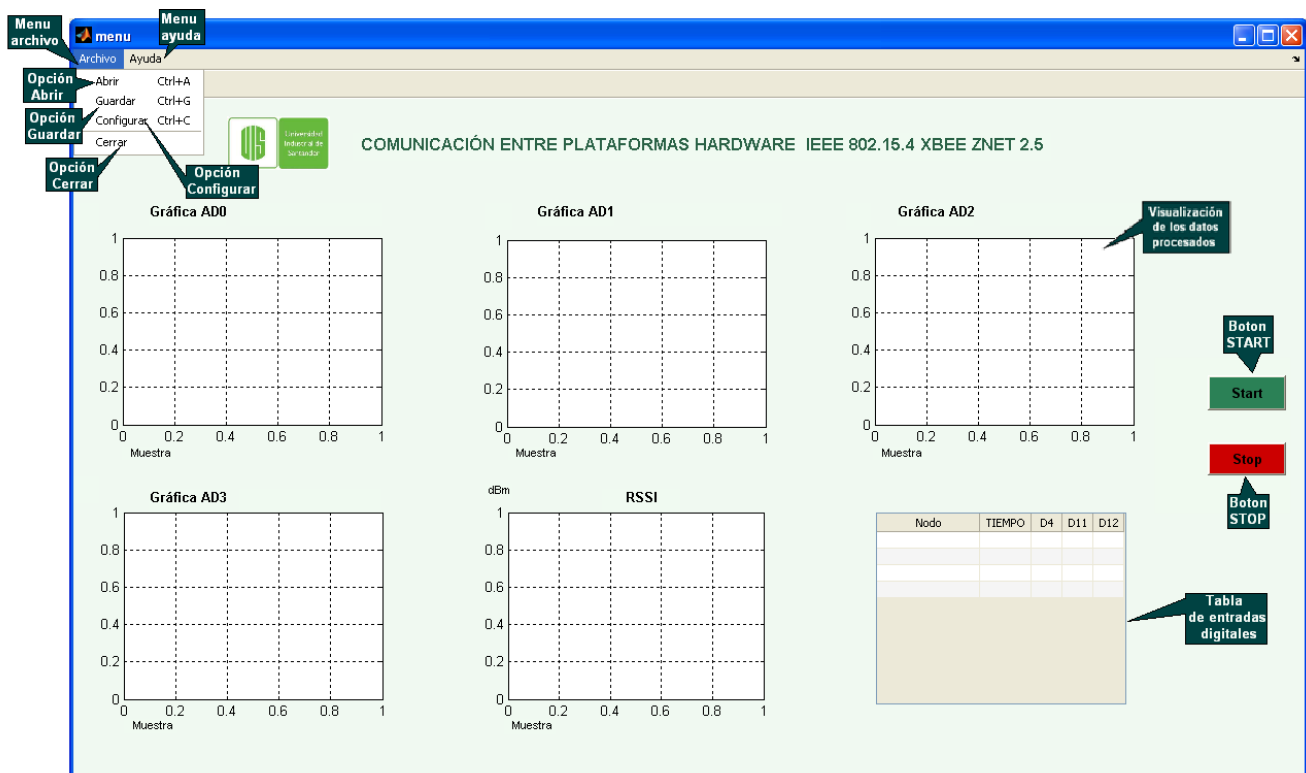
se miden variables diferentes usando la misma entrada ADC, el eje Y de la grafica no puede mostrar los valores con las unidades reales, a menos que la medida se muestre en volts. Si esto ocurre se perderían potencialidades del software porque el usuario tendría que realizar manualmente el cálculo del verdadero valor medido. Para finalizar, es importante que en el PC donde se utilizará la herramienta software desarrollada, se encuentren instalados los SDK (*Software Development Kit*) de los lenguajes de programación Matlab y Java, ya que fue en estos en donde se implementó la lógica del programa realizada con la filosofía descrita al inicio del capítulo.

A continuación, se presenta la descripción de las herramientas software desarrolladas. Cada una de estas tiene la misma funcionalidad, pero poseen cualidades propias del lenguaje de programación utilizado que las diferencian entre sí.

3.2 HERRAMIENTA SOFTWARE DESARROLLADA EN MATLAB

Tal como lo indica uno de los objetivos de este proyecto, debe desarrollarse un software que permita presentar de forma clara y precisa la información transmitida por los nodos de una red conformada por módulos Xbee, haciendo uso de las herramientas que ofrece el entorno computacional Matlab.

Como ya se menciono, para poder utilizar el software que cumple el objetivo descrito anteriormente, debe tenerse instalado en el PC el kit de desarrollo de Matlab. Cumpliendo con esta condición se habilita al usuario para utilizar la ventana de comandos y así ejecutar el código donde esta implementado toda la lógica de la aplicación.



d.

Figura 3.6 *Imágenes de las opciones del software. a. Barra de menú b. Barra de herramientas c. Ventanas de visualización de datos d. Imagen completa de la interfaz.*

Fuente: autor.

En la figura 3.6a, se observa la **barra de menú**, la cual contiene los siguientes componentes:

Menú archivo: En este componente son agrupadas todas las funciones relacionadas con la configuración de la sesión de trabajo y con la gestión de archivos que almacenan los datos en esta. El menú archivo contiene las siguientes opciones:

- **Opción abrir:** Permite abrir un archivo que contiene datos recolectados en una sesión de trabajo anterior con el fin de graficarlos. Al seleccionar la opción abrir, se despliega una ventana de dialogo que permite al usuario navegar por la estructura de carpetas del PC y elegir el archivo deseado. Una vez seleccionado el archivo, el programa recopila los datos contenidos

en él, para luego graficarlos como una señal continua en cada una de las ventanas de visualización.

- **Opción cerrar:** Cierra la Interfaz grafica y finaliza la sesión de trabajo.
- **Opción guardar:** Realiza una operación similar a la de la tecla “*print Screen*” de los computadores, es decir, toma una foto instantánea de la interfaz y la almacena en un archivo llamado “*screenshot*” localizado en la carpeta donde esta almacenada la herramienta software desarrollada.
- **Opción configurar:** Despliega una nueva GUI diseñada especialmente para brindarle al usuario una mayor flexibilidad en el uso de la herramienta software desarrollada, permitiendo así su utilización en diversas aplicaciones. En esta nueva GUI se introducirán valores requeridos para iniciar la sesión de trabajo.

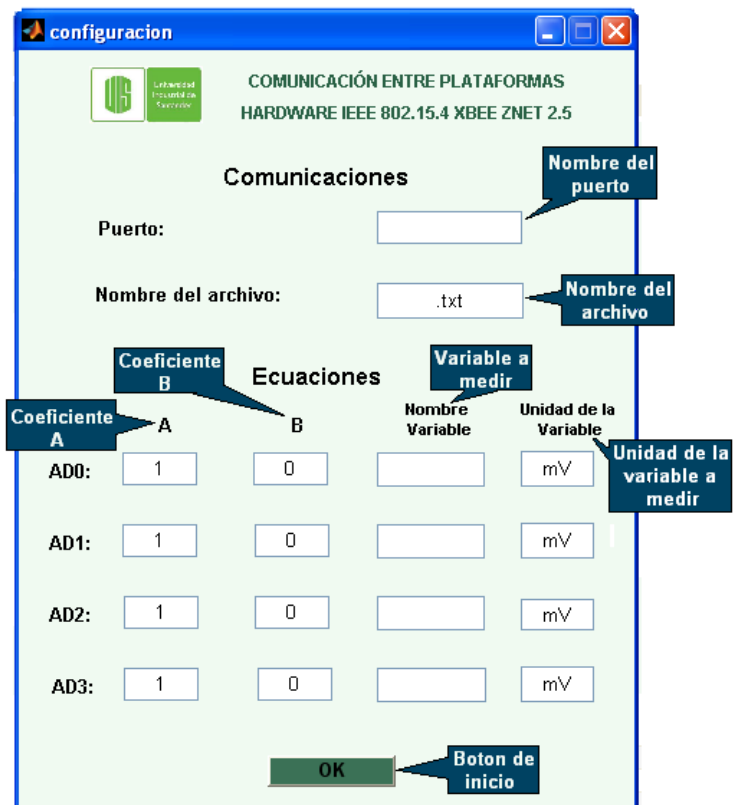


Figura 3.7 GUI de Configuración.

Fuente: Autor

En la figura 3.7 se muestra la GUI de configuración y a continuación se explican cada uno de sus componentes. Los archivos de diseño e implementación de la interfaz de configuración tienen asignados los nombres `configuracion.fig` y `configuracion.m` respectivamente.

- **Nombre del puerto:** Permite ingresar el nombre del puerto de comunicación al cual está conectado el modulo coordinador. El nombre del puerto se debe escribir en mayúscula y presenta el formato COMX, donde la variable X corresponde al número del puerto de comunicación. Por ejemplo, "COM3", "COM5", etc. El usuario debe saber con anticipación el nombre del puerto al cual está conectado el dispositivo Xbee. Si el usuario desconoce esta información puede seguir los siguientes pasos:

Entrar en Inicio > Panel de control > Sistema > Hardware > Administrador de dispositivos > Puertos (COM & LPT).

- **Nombre del archivo:** Esta opción permite que el usuario escriba el nombre del archivo tipo texto en donde se irán guardando, en tiempo real, los datos recolectados por los módulos Xbee durante la sesión de trabajo. Importante recordar que el nombre del archivo debe terminar con la extensión ".txt".
- **Variable a medir:** Guarda el texto ingresado en este campo para posteriormente mostrarlo en el ventana principal de la GUI como titulo de la grafica correspondiente a los datos del puerto ADC en el que está conectado el instrumento de medición.
- **Unidad de la variable a medir:** Guarda el texto ingresado en esta casilla como las unidades correspondientes a la variable de medición que fue configurada en el campo anterior, esta unidad es mostrada al lado del eje Y de la grafica correspondiente.

- **Coeficiente A y B:** Se establece que la relación existente, entre la tensión entregada a un terminal ADC y alguna variable física medida por un sensor conectado a dicho puerto, es lineal. Por lo tanto esta relación es modelada con la fórmula matemática: $y = Ax + B$, donde “x” es la tensión inyectada en el puerto ADC y “y” es la variable física en estudio. Conociendo esto, basta con ingresar el valor de la pendiente y el del corte con el eje de las ordenadas en los campos A y B respectivamente, para que en la ventana de visualización correspondiente se obtenga una gráfica coherente con el valor real de la variable en estudio. Cabe mencionar que si el usuario no ingresa ningún valor en los campos correspondientes a los coeficientes A y B y en los campos nombre y unidades de la variable de interés, los datos recolectados serán visualizados como valores expresados en Volts.
- **Botón inicio (OK):** Guarda los valores de la configuración realizada para utilizarlos posteriormente en la sesión de trabajo.

Menú Ayuda: En este componente son agrupadas todas las funciones relacionadas con el suministro de información útil para el buen manejo de la herramienta software desarrollada

En la figura 3.6 b, se presenta la barra de herramientas. En esta barra se encuentran agrupadas las funciones que dan una mayor interactividad al usuario con las ventanas de visualización. Estas funciones permiten un mejor análisis de los datos recolectados y otorgan una mayor comodidad a la hora de ver regiones específicas de la grafica. Las funciones de la barra de herramientas fueron implementadas haciendo uso del editor de barras de herramientas (“*editor toolbar*”) ofrecido por Matlab, este poderoso editor se encuentra dentro del ambiente de diseño GUIDE y con él, no es necesario programar una sola línea de código ya que estas son funciones estándar en aplicaciones de visualización de

gráficas. A continuación se presentan los componentes de la barra de herramientas:

Zoom In y Zoom out (1 y 2): Permite alejar o acercar el campo visual en una ventana de visualización y así tener mejor control del comportamiento de los datos.

Pan (3): Permite localizar una región que no se esté visualizando debido a un aumento de escala con la opción Zoom In.

Cursor (4): Permite observar de manera explícita y textual el valor de un punto de la gráfica.

La figura 3.6 c, muestra algunas opciones que también son muy importantes para el funcionamiento del software:

Botón START: Inicia la visualización de datos. Este botón debe accionarse después de haber realizado correctamente la configuración, en caso contrario el software no realizará tarea alguna.

Botón STOP: Este botón termina la captura y visualización de los datos, además cierra el puerto de comunicación sin cerrar la ventana principal.

Ventanas de visualización: Los datos medidos son visualizados en sus graficas correspondientes de acuerdo al puerto ADC del cual provengan. En el eje de las abscisas se indica el numero de la muestra tomada y en el eje de las ordenadas el valor medido en los puertos ADC del nodo remoto con sus unidades correspondientes. En caso de existir distintos nodos transmitiendo información al nodo concentrador, los datos estarán diferenciados por colores.

Tabla de entradas digitales: Está conformada por cinco columnas y un número indeterminado de filas. En la primera columna se indica el nombre del nodo del

cual provienen los datos, en la segunda columna se informa el número de la muestra mientras que en la columna 3, 4 y 5 se presenta el estado de las entradas digitales habilitadas en el módulo remoto (D4, D11 y D12).

En la parte central de la figura 3.6 d, se puede observar una grafica con el título “**RSSI**”, esta opción permite observar el valor de la potencia de la señal recibida durante la trasmisión de los datos.

El autor de este proyecto entrega a la Universidad Industrial junto con este documento un conjunto de anexos y archivos que hacen parte del producto final del proyecto. Entre los archivos entregados se encuentran algunos videos tutoriales que facilitan el aprendizaje y manejo de la herramienta software, además se entregan archivos ejecutables para poder utilizar la herramienta software.

Por cuestiones de confidencialidad solo se hace entrega del código fuente, al grupo de investigación CPS, el cual hará uso de esta herramienta en aplicaciones posteriores.

En la tabla 3.2 se presenta a groso modo las funciones y los comandos utilizados en la programación de cada uno de los componentes descritos anteriormente.

Interfaz	Componente	Función	Comando	Descripción
M e n u	Panel principal	Menu DMD_OpeningFcn DMD1_OutputFcn Figura1_CloseRequestFcn	get('ScreenSize')	Toma el tamaño de la pantalla
			set(gcf,'Position')	Coloca la interfaz en el medio de la pantalla
			imread	Carga la imagen con el logo de la UIS
			delete	Quita la ventana principal al pulsar el botón cerrar.
	Botón START	uipanel5_Selection	Global	Asigna los valores traídos de la interfaz de configuración a variables locales
			fopen(archivo) (com)	Crea y abre un archivo texto Abre un puerto de comunicación
	Botón STOP	ChangeFcn	serial(com)	Configura el puerto de comunicación
			fprintf(archivo)	Escribe datos con formato en un archivo texto
			fread(com)	Lee los bytes del puerto de comunicación
			fwrite(com)	Escribe bytes en el puerto de comunicación
	Menu archivo	mArchivo_Callback	plot	Grafica en las ventana de visualización los datos del puerto de comunicación
			No Aplica (NA)	Abre las opciones relacionadas con la configuración de la sesión de trabajo y con la gestión de archivos
	Opción abrir	mAbrir_Callback	uigetfile	Abre interfaz para navegar por estructura de archivos
			Textread (archivo)	Lee datos del archivo texto
			plot	Grafica los datos del archivo texto en las ventanas de visualización
	Opción guardar	mGuardar_Callback	java.awt.Robot	Carga la librería Robot de java
			createScreenCapture	Toma la foto instantánea de la pantalla
			java.io.File	Crea el archivo .jpg donde guarda la foto tomada
			ImageIO.write	Escribe la foto tomada en el archivo .jpg creado.
	Opción configurar	mConfigurar_Callback	NA	Llama la interfaz de configuración
Menu ayuda	mAyuda_Callback	NA	Informa al usuario sobre las opciones de ayuda entregadas con este software	
Panel principal	mCerrar_Callback	NA	Cierra la interfaz y el puerto de comunicación utilizado	

Interfaz	Componente	Función	Comando	Descripción
Configuración	Panel de configuración	Configuración	imread	Lee la imagen del logo de la UIS
		configuracion_OpeningFcn		
		configuracion_OutputFcn	imshow	Coloca la imagen leida en la GUI.
	Nombre puerto	edit1_Callback	NA	Crea y habilita escritura en todos los campos de textodefinidos
		edit1_CreateFcn	NA	
	Boton OK	pushbutton1_Callback	global	Asigna los valores de los campos texto a variables globales para poderlos utilizar en la interfaz principal
Panel de configuración	figure1_CloseRequestFcn	NA	Quita la GUI después de pulsar el botón OK	

Tabla 3.2 Funciones y comandos utilizados para la implementación de la interfaz grafica en Matlab.

Fuente: autor

3.3 HERRAMIENTA SOFTWARE DESARROLLADA EN JAVA

Otro de los objetivos planteados para el desarrollo de este proyecto, es realizar la misma herramienta software desarrollada con Matlab pero utilizando el lenguaje de programación orientado a objetos Java. En la figura 3.7 se presenta la interfaz grafica que evidencia el resultado del cumplimiento de este objetivo.

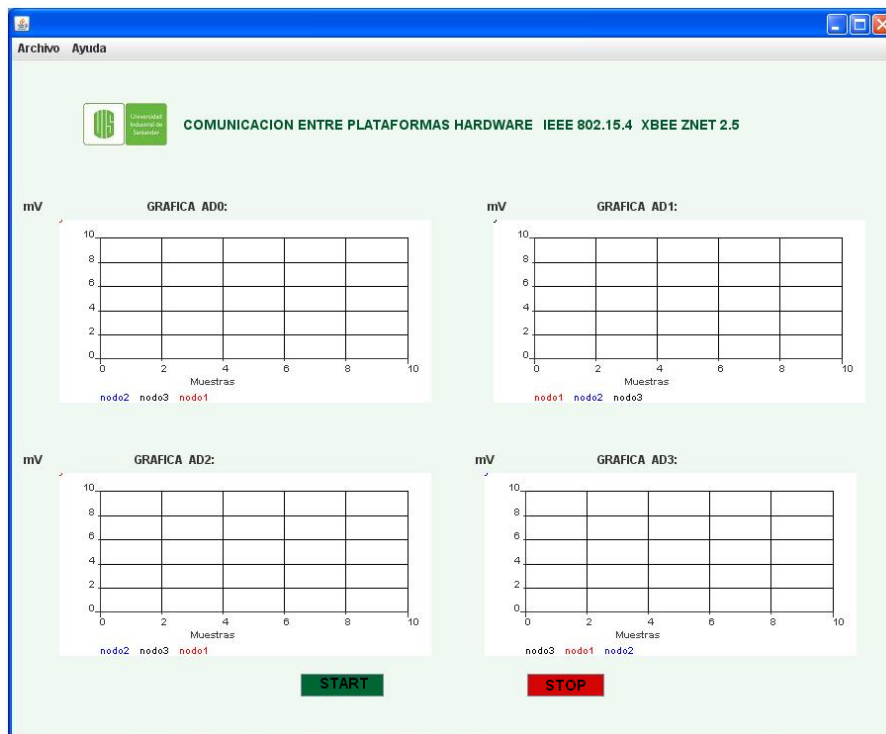
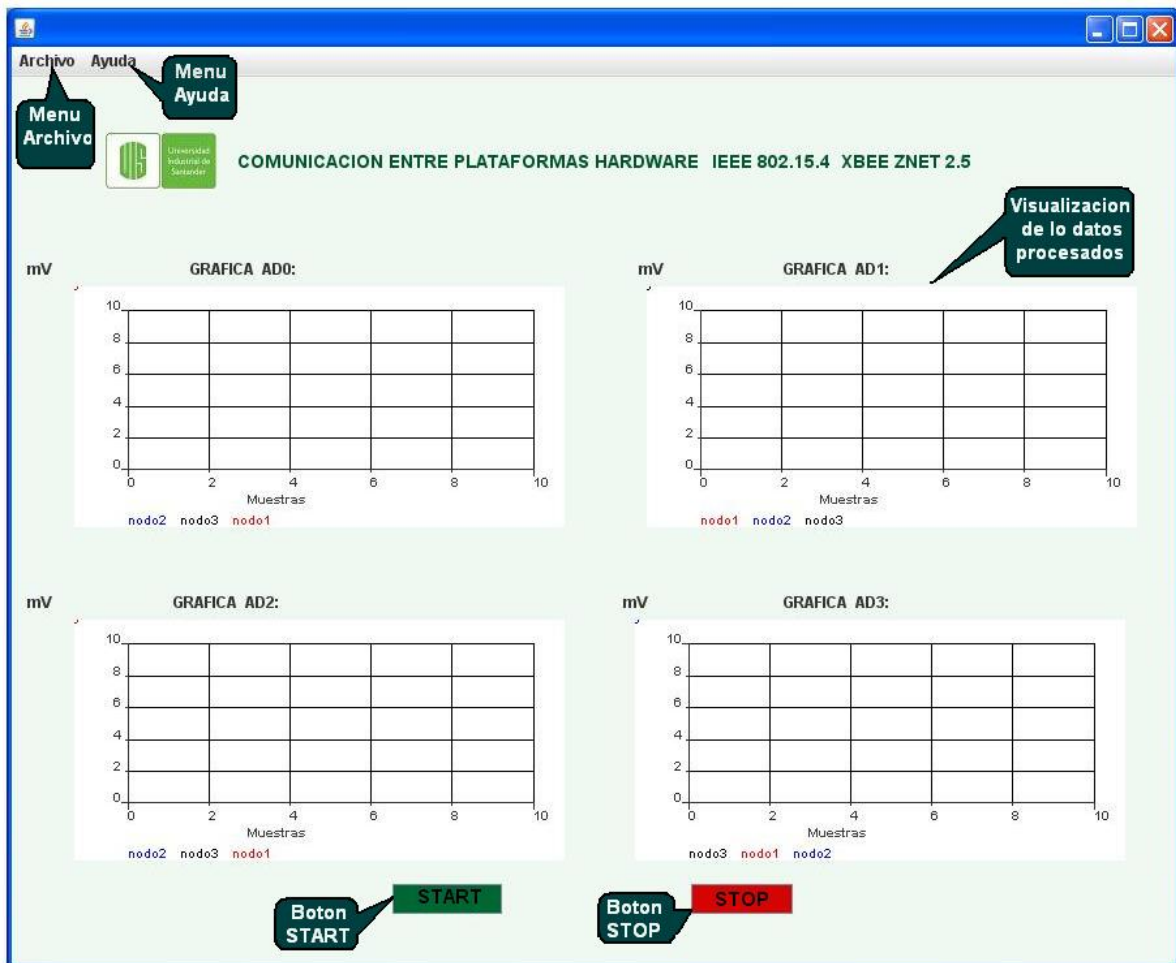


Figura 3.7 Interfaz grafica desarrollada en Java.

Fuente: Autor

En la figura 3.8 se muestra la imagen de la interfaz realizada y se señalizan cada uno de sus componentes. En la parte superior de esta figura, se distingue una barra de menú. Además de esto la interfaz cuenta con cuatro ventanas de visualización y dos botones que permiten iniciar y finalizar una sesión de trabajo (botón START y botón STOP respectivamente).

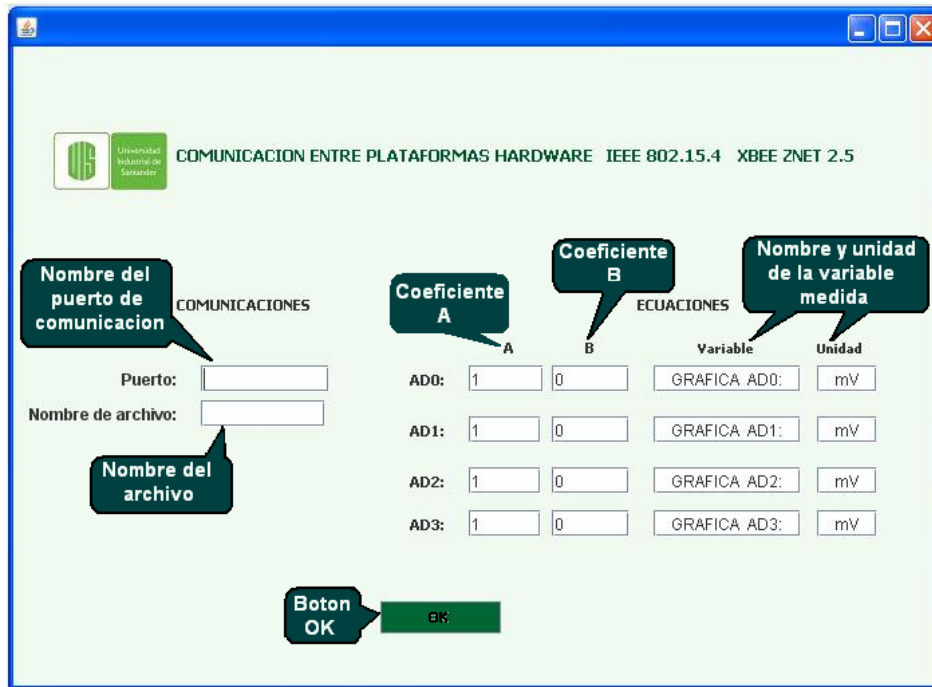
El uso y funcionamiento de la versión realizada en Java fue diseñada basándose en la herramienta implementada con Matlab, dándole características de funcionamiento completamente idénticas, por lo tanto no es necesario realizar una explicación sobre cada uno de los componentes utilizados por esta GUI y por la ventana auxiliar de configuración.



a.



b.



c.

Figura 3.8 Imagen de las opciones del software a. Imagen de la ventana principal de la interfaz y sus componentes b. Diferentes opciones de la barra de menú c. Imagen de la interfaz de configuración .Fuente: Autor

En la tabla 3.3 se presentan las librerías, clases y métodos utilizados para la implementación del software.

Clase	Componente (Nombre de componente)	Método	Clases que utiliza	Descripción
JChart2D	Panel principal (panel1)	Main	NA	Hace visible la ventana principal y llama al método constructor (JChart2D)
		JChart2D	NA	Es el método constructor y llama al método InicializaComponentes()
		InicializaComponentes	GroupLayout Swing ZoomableChart ChartPanel	Asigna nombre, valores y dimensiones a cada uno de los componentes de la GUI y los coloca en sus lugares respectivos a través de la clase GroupLayout.
	Botón Start (startButton)	jButton1ActionPerformed	SerialPort	Configura el puerto serial seleccionado y habilita la notificación de eventos sobre el puerto. (notifica si el puerto recibe datos)
			CommPortIdentifier	Identifica cuales son los puertos disponibles para comunicación.
			PrintWriter FileOutputStream	Crea los archivo texto donde serán almacenados los datos de la sesión
	Botón Stop (stopButton)	jButton2ActionPerformed	InputStream SerialPort	Cierra el puerto de comunicación (SerialPort) y lee los datos que faltaron por procesar (InputStream)
	Menú Abrir (jMenuItem1)	jMenuItem1ActionPerformed	JFileChooser	Permite seleccionar un archivo en la estructura de carpetas del PC
			Scanner	Lee el archivo seleccionado utilizando el formato correcto
			ITrace2D	Toma los valores leídos y los añade a la gráfica correspondiente
	Menú Guardar (jMenuItem2)	jMenuItem2ActionPerformed	JFileChooser	Permite seleccionar una ubicación al archivo dentro de la estructura de carpetas del PC
			BufferedImage	Guarda una imagen instantánea de la Interfaz
			File	Crea el archivo donde se guarda la foto tomada con BufferedImage
			ImageIO	Escribe la imagen en el archivo creado con File que se encuentra en la ubicación determinada por JFileChooser.
	Menú Configurar (jMenuItem3)	jMenuItem3ActionPerformed	config	Llama a la clase interna config que permite la configuración de los parámetros de la GUI.

Clase	Componente (Nombre de componente)	Método	Clases que utiliza	Descripción
Config (Clase Interna)	Panel de Configuración (jPanel1)	main	NA	Hace visible la ventana de configuración y llama al método constructor (config)
		config	NA	Es el método constructor y llama al método initComponents()
		initComponents	Swing AWT GridLayout	Asigna nombre, valores y dimensiones a cada uno de los componentes de la GUI y los coloca en sus lugares respectivos a través de la clase GridLayout.
	Botón OK (ok)	okActionPerformed	NA	Asigna valores a variables globales correspondientes a los campos texto de la ventana de configuración. Al pulsarlo se hace invisible la ventana de configuración y se regresa a la ventana principal.

Tabla 3.3 Librerías, clases y métodos utilizados para la implementación del software en java

Fuente: Autor

Capítulo 4.

PRUEBAS Y RESULTADOS.

Con el propósito de cuantificar el desempeño del software y verificar el funcionamiento de los elementos que conforman la interfaz, fueron realizadas varias pruebas utilizando diferentes herramientas e instrumentos de medición que pudiesen interactuar con los módulos Xbee. Como instrumentos de prueba fueron seleccionados un generador de señales (HUNG CHANG 9205C), un sensor de humedad (HIH 4000 de Honeywell), un sensor de temperatura (TMP36 de Analog devices), un analizador de redes IEEE802.15.4 (Packet Sniffer CC2420EM+CC2400EB) y tres módulos Xbee configurados como nodos remotos.

La prueba con el generador de señales fue diseñada con dos objetivos:

El primero, busca ilustrar el correcto funcionamiento del software, al mostrar en forma precisa, una señal producida con dicho generador (esto implica la realización de todas las configuraciones pertinentes tanto en los módulos Xbee como en el generador de señales). Por otro lado el segundo objetivo pretende determinar el valor aproximado de la máxima frecuencia de muestreo (F_s) con la cual es posible configurar el módulo Xbee para garantizar la reproducción exacta, en la interfaz gráfica, de una señal que presente variaciones de amplitud considerables en pequeños periodos de tiempo (señales de alta frecuencia).

La prueba en donde fueron utilizados los sensores de temperatura y humedad relativa, contemplaban el objetivo de mostrar las potencialidades del software cuando este es empleado en aplicaciones de monitorización de variables físicas ambientales y otras aplicaciones que requieran enviar información de variables de instrumentación en forma inalámbrica.

La prueba en donde el analizador de red fue utilizado, enfocaba sus esfuerzos en verificar que los datos recibidos por el software a través de comunicación serial, eran los mismos datos recibidos por el módulo coordinador a través de la comunicación inalámbrica con otros nodos remotos. Esta prueba tenía como objetivo secundario, demostrar que los módulos Xbee basan su comunicación inalámbrica en el estándar IEEE 802.15.4.

La última de la de las pruebas realizadas, buscaba probar el funcionamiento del software desarrollado en un ambiente de red conformado por tres nodos remotos y un nodo coordinador. Con esto se pretendía mostrar que el software es capaz de manejar simultáneamente información proveniente de tres fuentes distintas.

En la tabla 4.1 se presenta, de manera resumida, la descripción de cada una de las pruebas realizadas.

PRUEBA	OBJETIVOS
Generador de señales	<ul style="list-style-type: none"> • Mostrar el desempeño y la precisión de la herramienta desarrollada. • Hallar la máxima velocidad con que los módulos Xbee pueden adquirir muestras de sus puertos ADC.
Sensores	<ul style="list-style-type: none"> • Demostrar la utilidad del software en aplicaciones de monitorización ambiental.
Analizador de red	<ul style="list-style-type: none"> • Mostrar la relación existente entre la comunicación serial y la comunicación inalámbrica en una red de módulos Xbee. • Realizar análisis sobre los protocolos de comunicación inalámbrica utilizada por los módulos Xbee.
Varios nodos remotos	<ul style="list-style-type: none"> • Verificar el desempeño de la herramienta en un ambiente de red de tres nodos.

Tabla 4.1 Resumen de pruebas realizadas para probar el funcionamiento de la herramienta.

Fuente: Autor.

4.1 PRUEBAS CON EL GENERADOR DE SEÑALES

A continuación se presentan los resultados obtenidos con las pruebas realizadas utilizando el generador de señales. Para cumplir con el primer objetivo planteado en esta prueba, se utilizó la herramienta software desarrollada en el entorno computacional Matlab, sin embargo los resultados no varían cuando se utiliza la herramienta desarrollada en el lenguaje de programación Java.

La señal del generador es entregada (a través de uno de los puertos ADC) a un módulo Xbee configurado como nodo remoto. A este modulo Xbee se le establece un periodo de muestreo que fue escogido de acuerdo con los límites establecidos por el teorema de Nyquist para garantizar que las muestras adquiridas formarán fielmente la señal original. La figura 4.1, muestra la pantalla del osciloscopio graficando la señal entregada al puerto ADC del modulo Xbee remoto. Las muestras tomadas de esta señal son transmitida a un nodo coordinador, quien a su vez está ligado a un PC en donde se ejecuta la aplicación desarrollada. Como puerto de entrada, en el nodo remoto fue seleccionado el terminal AD1 (pin 19), cabe resaltar que la elección de un puerto ADC distinto al escogido no cambia los resultados de la prueba.

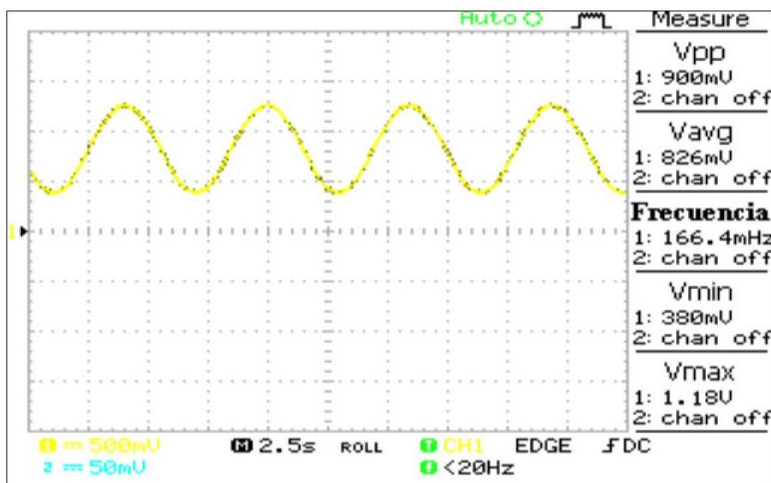


Figura 4.1 Imagen en el osciloscopio de la señal entregada al Xbee.

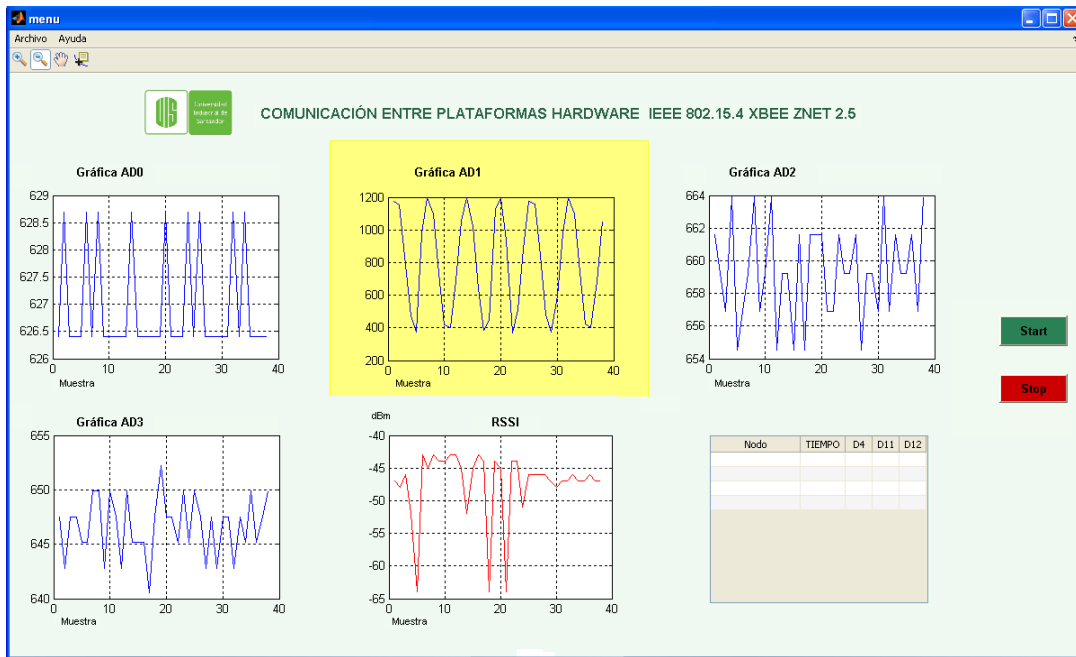
Fuente: Adaptada por el autor de la imagen capturada con el osciloscopio

De acuerdo con el Teorema de Nyquist, debe tenerse en cuenta que los puertos ADC del modulo remoto tienen que ser “encuestados” con una frecuencia (f_s) igual o mayor al doble de la frecuencia de la señal producida por el generador (f). Esto último se hace con el fin de asegurarse que la señal observada en la interfaz grafica de Matlab sea una representación correcta e idéntica de la señal original. Debe recordarse que el periodo de muestreo ($1/f_s$) de los puertos ADC en el nodo remoto es configurado a través del parámetro ATIR.

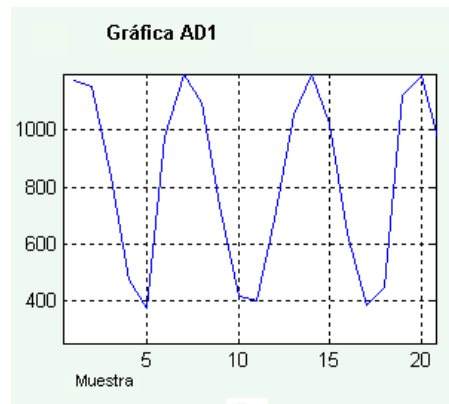
Durante la prueba, el generador de señales fue manipulado hasta conseguir una señal sinusoidal con una amplitud pico-pico de 900[mV] y una frecuencia de 166[mHz] (tal y como se aprecia en la figura 4.1), la cual fue entregada a un modulo Xbee configurado para encuestar sus puertos ADC con una frecuencia de 670[mHz] (cada 1,5 segundos). Lo anterior está en plena concordancia con lo expuesto anteriormente sobre el principio de Nyquist [29].

La sesión de trabajo (captura y visualización de datos) tuvo una duración de 60 segundos. A continuación, se muestra en la figura 4.2 una imagen de los datos capturados con el software realizado en Matlab.

Al contrastar la grafica correspondiente al puerto AD1 (resaltada en color amarillo en la figura 4.2A), con la grafica que se observó en la pantalla del osciloscopio (figura 4.1), se puede apreciar que los valores correspondientes a la máxima y mínima tensión de la señal (y obviamente la forma de esta) son idénticos, lo cual habla de la precisión del sistema y corrobora la veracidad de los datos graficados.



A.



B.

Figura 4.2 Imagen obtenida con el software realizado en matlab. A. Imagen de la interfaz B. Imagen ampliada de la grafica AD1

Fuente: Autor

El cumplimiento del segundo objetivo planteado para las pruebas realizadas usando el generador de señales, se llevo a cabo empleando la herramienta software desarrollada en Java.

Con esta prueba se pretende determinar el valor aproximado de la máxima frecuencia de muestreo (F_s) con la cual se puede configurar el módulo Xbee para

garantizar una reproducción exacta de una señal que presente variaciones de amplitud considerables en pequeños periodos de tiempo. Adicionalmente, con la realización de la prueba se busca verificar el buen funcionamiento de la herramienta software desarrollada con Java, en cuanto a precisión y coherencia.

Las imágenes relacionadas con el montaje de las pruebas, son mostradas en el anexo B.

En la prueba se puede utilizar cualquier generador al que se le pueda variar la frecuencia de la señal de salida. Para esta prueba, inicialmente se configuró el modulo Xbee remoto con un periodo de muestreo de 1 Hz y se le entregó en su puerto AD2 una señal sinusoidal cuya frecuencia fue igual a 166,7 mHz y con otras características mostradas en la figura 4.3.

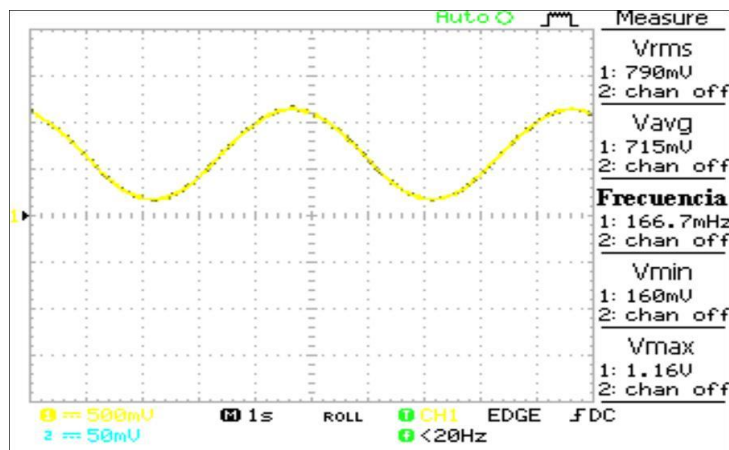


Figura 4.3 Imagen en el osciloscopio de la señal entregada al Xbee.

Fuente: Adaptada por el autor de la imagen capturada con el osciloscopio

La figura 4.4 muestra los datos graficados en la sesión de trabajo correspondiente a la realización de la prueba, en esta se incluye una ampliación de la imagen mostrada en la grafica correspondiente al puerto AD2. En esta figura se pueden observar detalles de la correcta reproducción de la señal entregada por el

generador al nodo remoto en la interfaz, confirmando el funcionamiento y las potencialidades de la herramienta software.

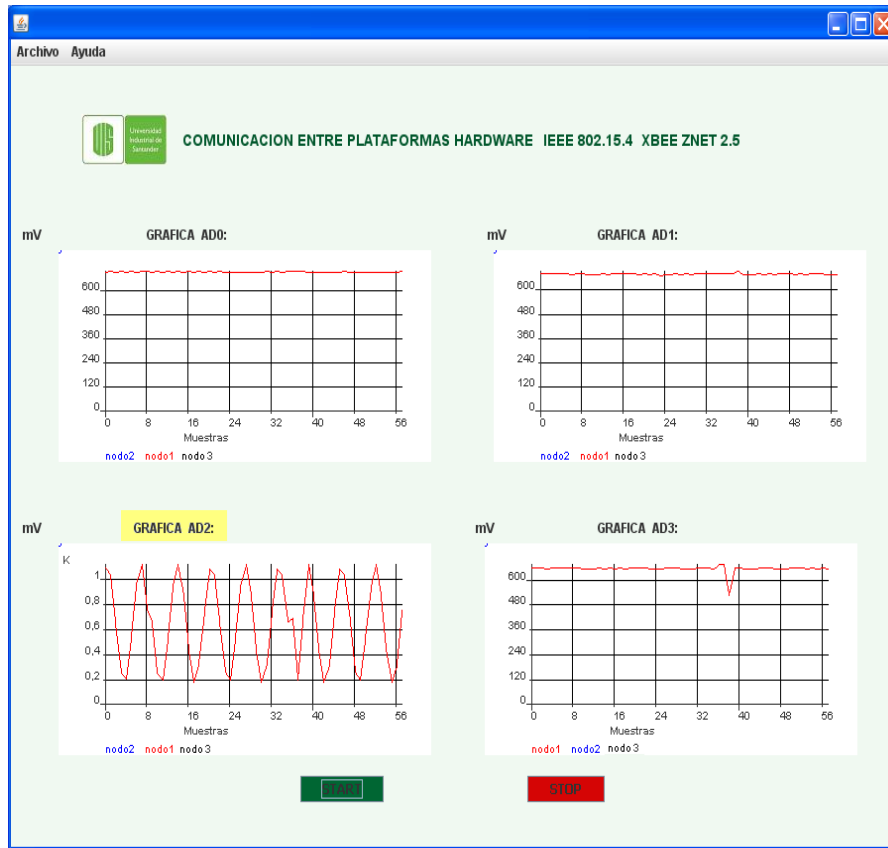


Figura 4.4a. Imagen completa de los datos recolectados en la sesión de trabajo.

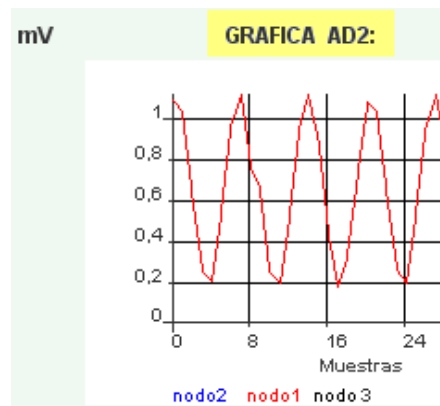


Figura 4.4 b. Imagen ampliada de la señal entregada por el generador al puerto AD2 del modulo remoto. Fuente: Autor.

Para calcular, el valor de la máxima frecuencia de muestreo, se realizaron varias sesiones de trabajo, las cuales consistieron en tomar datos durante sesenta segundos para luego comparar la cantidad de muestras graficadas en la interfaz, con la cantidad de muestras que teóricamente deberían haber sido recolectadas por el sistema. El número teórico de muestras es el resultado de dividir el tiempo total de una sesión de trabajo entre el periodo de muestreo configurado en el Xbee remoto. En caso de existir alguna diferencia entre el número de muestras graficadas y el número teórico de muestras, era necesario calcular la frecuencia con que el Xbee realmente encuestaba sus puertos ADC. Este valor se denomina frecuencia real de muestreo y se obtiene dividiendo el tiempo total de la sesión de trabajo (60 segundos) y el número de muestras graficadas en la interfaz.

En la tabla 4.2 se presentan los datos de configuración del nodo Xbee remoto como lo es la frecuencia de muestreo ($1/T_s$), además datos de configuración del generador de señales en cuanto a frecuencia de señal de salida, la cantidad de muestras tanto graficadas como teóricas y los resultados de los cálculos de la frecuencia real de muestreo al final de cada sesión de trabajo.

Prueba numero:	Frecuencia de muestreo del Xbee (Fs) [Hz]	Periodo de muestreo del Xbee (Ts) [s]	Valor hexa. de Ts	Frecuencia de la señal del generador (F) [Hz]	Numero teórico de muestras	Numero de muestras reales	Frecuencia real de muestreo del Xbee [Hz]
1	0,985	1,015	3F7	0,4699	59	59	0,973
2	1,97	0,508	1FC	0,9387	118	115	1,89
3	3,93	0,254	FE	1,6732	238	227	3,74
4	7,87	0,127	7F	3,91	477	360	5,93
5	15,625	0,064	40	7,29	945	353	5,8

Tabla 4.2 Valores de frecuencia y muestras capturadas por el modulo Xbee.

Fuente: Autor

El periodo de muestreo (T_s), plasmado en la columna tres de la tabla 4.2, corresponde al valor con el cual se configura el módulo Xbee para que capture muestras de la información que esté recibiendo a través de sus puertos y las envíe al modulo coordinador. Para configurar los módulos Xbee con el periodo de muestreo deseado es necesario convertirlo a formato hexadecimal, el resultado de esta conversión es asignado al parámetro ATIR. Es posible utilizar el software propietario XCTU para facilitar esta tarea.

Las pruebas iniciaron empleando un periodo de muestreo de 1,015 segundos. Este tiempo fue escogido por comodidad ya que permite hacer un seguimiento claro de cada una de las muestras graficadas en las ventanas de visualización. Este periodo de muestreo fue disminuyendo a razón del cincuenta por ciento del valor inicial en cada prueba realizada. Las pruebas finalizaron al alcanzar un periodo de muestreo igual a 64[ms], con este valor ya no es posible reproducir los cambios de la señal del generador.

La frecuencia real de muestreo, para cada una de las pruebas, fue tabulada en la columna ocho de la tabla 4.2. En dicha columna se aprecia que los dos últimos valores son similares y que los valores correspondientes en la columna de muestras graficadas (columnas siete) son prácticamente iguales, esto indica que entre encuestas sucesivas de los puertos ADC no puede existir un periodo de tiempo inferior a **172[ms]**. Lo anterior establece un límite superior a la frecuencia de la señal entregada por el generador, es decir, que el sistema no será capaz de capturar fielmente señales con frecuencias superiores a 2,9[Hz]. Este resultado evidencia que la frecuencia máxima de muestreo (o periodo mínimo de muestreo) para los módulos Xbee es de 5,8[Hz].

A partir de los datos obtenidos con las pruebas realizadas (documentados en la tabla 4.2) se concluye que:

- La influencia del software, en la restricción impuesta con la máxima frecuencia de muestreo, es nula. Según el manual de usuario de los módulos Xbee existen dos razones para tener pérdidas de datos en la comunicación entre nodo remoto y nodo concentrador. La primera es hacer una configuración errada de la interfaz de comunicación serial, causando que la velocidad con que salen los datos hacia el computador sea más lenta que la velocidad con que el modulo coordinador los recibe a través del radio enlace con el modulo remoto y la segunda es tener un control de flujo que impida que el nodo coordinador envíe datos hacia el computador. Ninguna de las dos razones está presente en el sistema ya que la autora de este proyecto se ha cerciorado de no incluir control de flujo y de configurar correctamente el puerto de comunicación. Por lo tanto la fuente de la restricción es el hardware del modulo Xbee.
- El software, es capaz de procesar efectivamente todas las tramas entregadas por el modulo concentrador a través del puerto de comunicación serial al que está conectado.
- La máxima frecuencia que puede tener una señal inyectada en cualquier puerto ADC del modulo Xbee es de 2,9 [Hz], esta limitación no es impuesta por el software, por el contrario es una limitación del hardware.

Para finalizar, se presenta la figura 4.5, en la cual se muestran los dos extremos de las pruebas realizadas.

En la figura 4.5a, se observa que la señal entregada al puerto ADC del nodo remoto, tiene una frecuencia de 2.7 [Hz]⁹. Este valor se encuentra en el límite

⁹ Frecuencia de muestreo máxima=5.8 HZ, Periodo de muestreo mínimo= 172.4 ms , se tiene una señal rápida.

superior de la frecuencia que puede tener una señal en estudio y cuyos datos desean ser transmitidos en una red formada por módulos Xbee. De acuerdo con lo anterior es completamente comprensible, que en la imagen de la ventana de visualización, no se distinga correctamente la forma esperada de la señal (forma sinusoidal), ya que la cantidad de muestras perdidas en la sesión de trabajo es significativa. A pesar de esto es posible comprobar que los valores graficados, son muestras pertenecientes a la señal entregada al nodo remoto, si se observa que el valor máximo y mínimo de la tensión en cada uno de los extremos del sistema (nodo remoto y nodo concentrador) es igual.

En la figura 4.5b, la señal observada tiene una frecuencia de 4.7 [mHz] y el periodo de muestreo configurado en el nodo remoto es 1.015 [s]. En estas condiciones se aprecia una excelente reproducción de la señal que entra al puerto AD2 de dicho nodo remoto, ya que no se perdió una sola muestra en la sesión de trabajo.

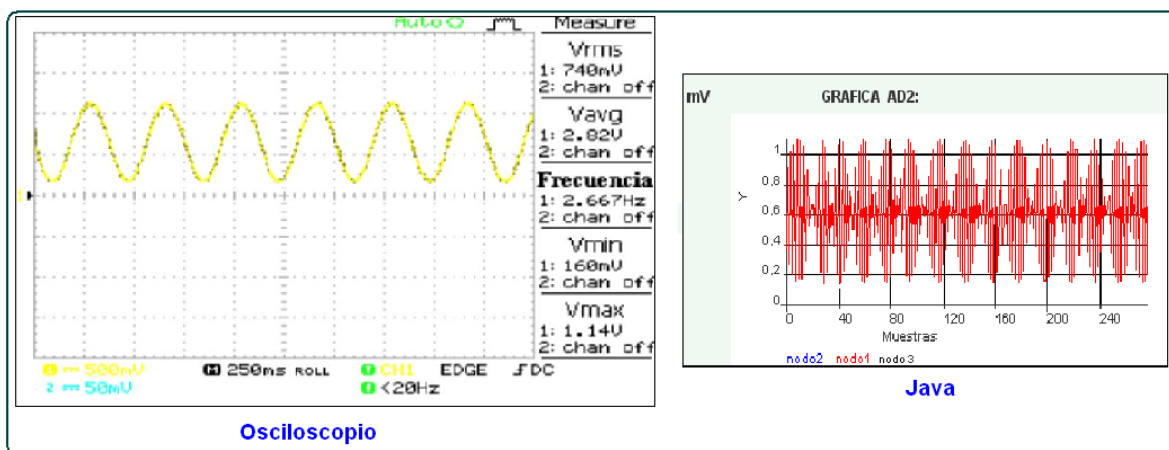


Figura 4.5a. Desempeño del sistema utilizando señales de prueba con frecuencias distintas, Señal con frecuencia de 2,66 [Hz]

Fuente: Autor

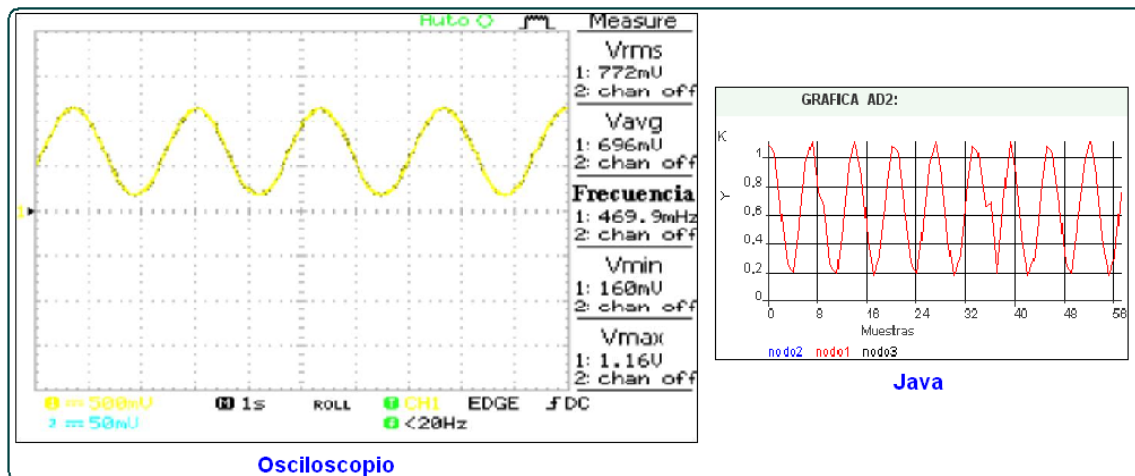


Figura 4.5b. Desempeño del sistema utilizando señales de prueba con frecuencias distintas, Señal con frecuencia de 0,470 [Hz]

Fuente: Autor

La figura 4.6, muestra imágenes de señales visualizadas con la herramienta software de acuerdo al número de la prueba y al periodo de muestreo con que se haya configurado el modulo Xbee para la sesión de trabajo. La secuencia de estas señales permite observar cómo afecta el periodo de muestreo mínimo la precisión del sistema, es decir, como se perjudica la reproducción correcta de la señal entregada al puerto ADC cuando se llega al límite inferior del periodo de muestreo del módulo Xbee (172ms).

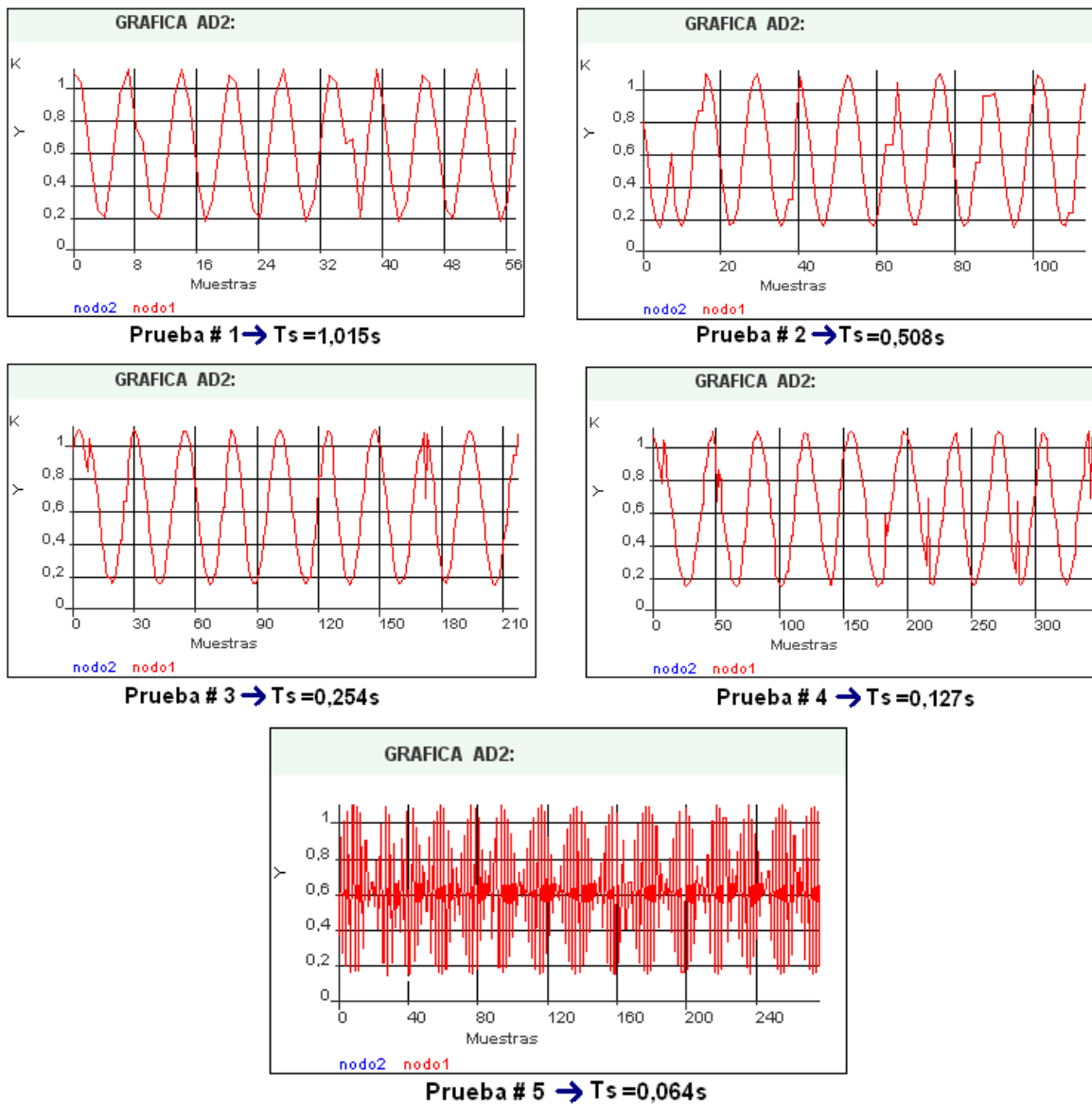


Figura 4.6 Señales obtenidas en la prueba con el generador de señales.

Fuente: Autor.

4.2 PRUEBAS UTILIZANDO SENSORES DE TEMPERATURA Y HUMEDAD.

Con esta prueba, se busca demostrar el funcionamiento del software mediante la visualización de los datos medidos a través de una plataforma hardware

constituida por sensores de temperatura y humedad y que transmite los datos de dichas variables físicas a través de un modulo Xbee. La plataforma escogida para la realización de esta prueba, hace parte de las herramientas de trabajo que tiene el grupo de investigación CPS y es el resultado del proyecto de pregrado titulado “Diseño e implementación de una plataforma de red de sensores inalámbricos para la monitorización de variables físicas ambientales en un galpón avícola”, realizado por Héctor De la Hoz y Alexander Castro [1]. En el anexo A, se encuentran las hojas de especificaciones técnicas de los sensores utilizados por esta plataforma.

Los datos mostrados a continuación, corresponden a los resultados obtenidos al implementar la metodología de la prueba, con la herramienta software realizada en Matlab. Las imágenes que evidencia la realización de la pruebas y que muestran las condiciones en que estas fueron desarrolladas se presentan en el anexo B. Sin embargo, la figura 4.7a, muestra un bosquejo o plano del lugar de la prueba y ubicación de los dispositivos empelados en esta.

La prueba se desarrolló en el salon social del grupo de investigacion CPS el cual esta equipado con un aire acondicionado de control digital. Este salón es lo suficientemente pequeño como para obtener variaciones de temperatura y humedad en un periodo de tiempo relativamente corto, si se concidera que para estas dos variables físicas ambientales los cambios en el ambiente son percibidos mucho despues de haber desactivado la fuente que las controla (en este caso, el aire acondicionado). El set point (o punto de operación deseado) de la temperatura en lugar donde se realizó la prueba, fue graduado para que esta variable alcanzara un valor de 18° C, tal y como se muestra en la figura 4.7b.

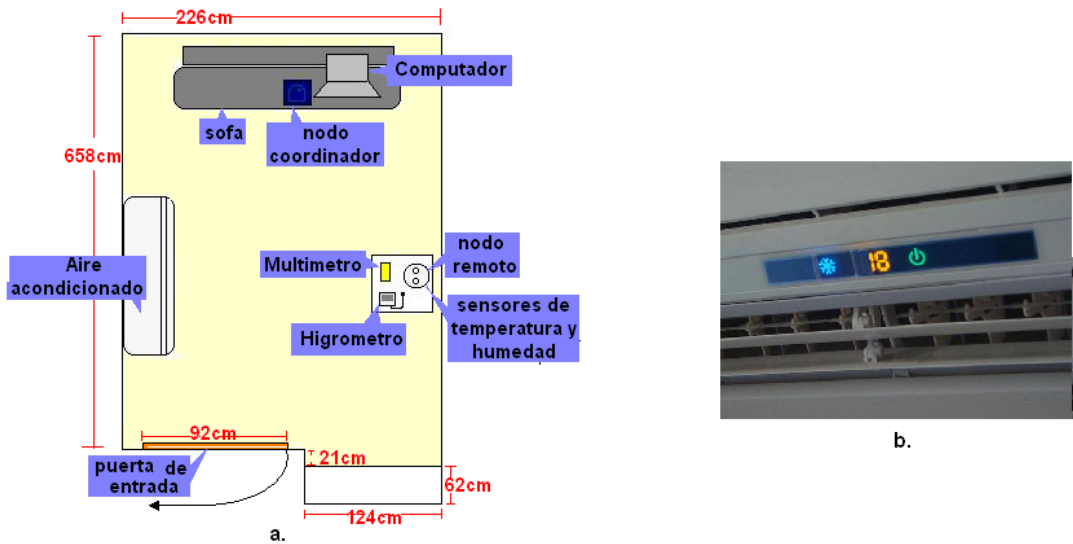


Figura 4.7 Condiciones del lugar de trabajo a. Ubicación de los dispositivos empleados en la prueba b. Valor de la temperatura programada al aire acondicionado del lugar de trabajo.

Fuente: autor

Hay que destacar que la variación de humedad está relacionada con la variación de temperatura, es decir que si la temperatura se mantiene constante, se espera que la humedad relativa también se mantendrá constante.

Se realizaron dos sesiones de trabajo: en la primera sesión se alteraron los coeficientes A y B¹⁰ de la ventana de configuración, con el objetivo de visualizar los datos de temperatura y humedad relativa con sus respectivas unidades reales (Celsius y porcentaje de humedad relativa). En la segunda sesión, se omitió el uso de dichos coeficientes, dejando los valores que trae el software por defecto para observar los datos visualizados con la interfaz gráfica con unidades de volts (V).

Los datos de temperatura se visualizaron en la gráfica correspondiente a la entrada analógica AD1 y los datos de humedad en la gráfica correspondiente al

¹⁰ Revisar los componentes de la GUI de configuración.

puerto AD3. Los puertos ADC utilizados son fijados por la plataforma hardware empleada y no hay opción de cambiarlos.

El valor de los datos tomados por los instrumentos de medición y visualizados con el software, fueron corroborados con el uso del higrómetro de referencia 445713 fabricado por la empresa EXTECH Instruments, el cual está diseñado para medir exclusivamente las variables de interés para la prueba (temperatura y humedad).

La figura 4.8, muestra los dispositivos utilizados para el desarrollo de la prueba: Un computador (necesario para ejecutar el software de visualización), un módulo coordinador, una plataforma hardware (equipada con sensores de temperatura, humedad y un modulo xbee configurado como nodo remoto) y finalmente, como instrumentos de contraste, un higrómetro y un multímetro.

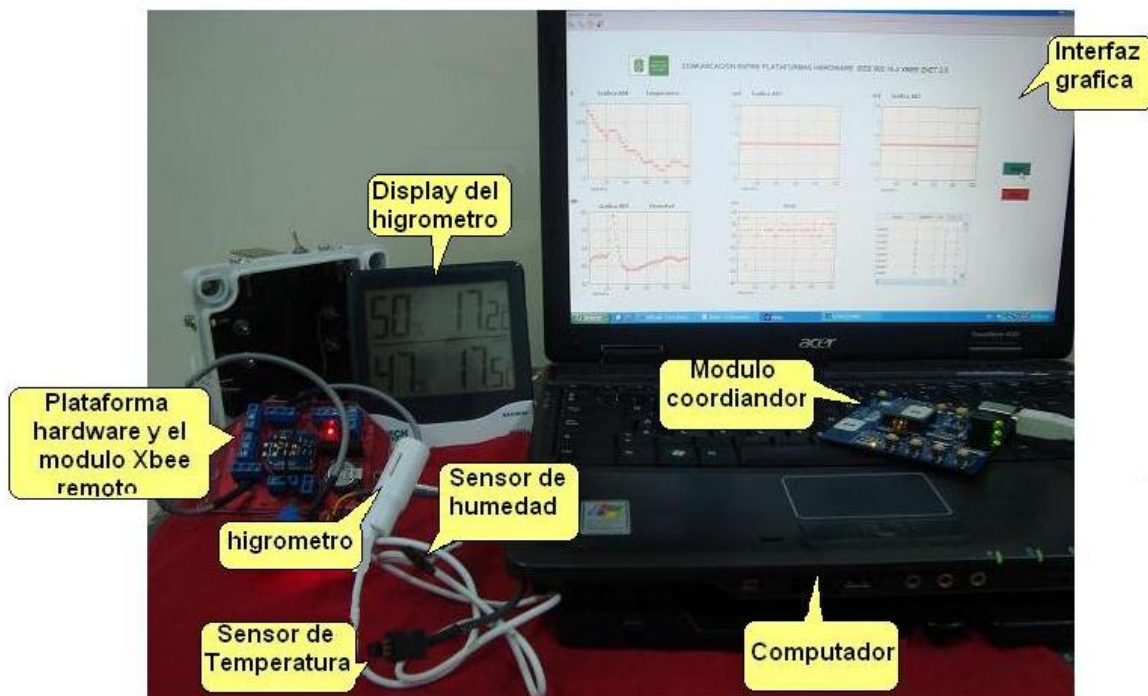


Figura 4.8 *Dispositivos utilizados en la prueba del software realizada en el salón social del grupo de investigación CPS.*

Fuente: Autor.

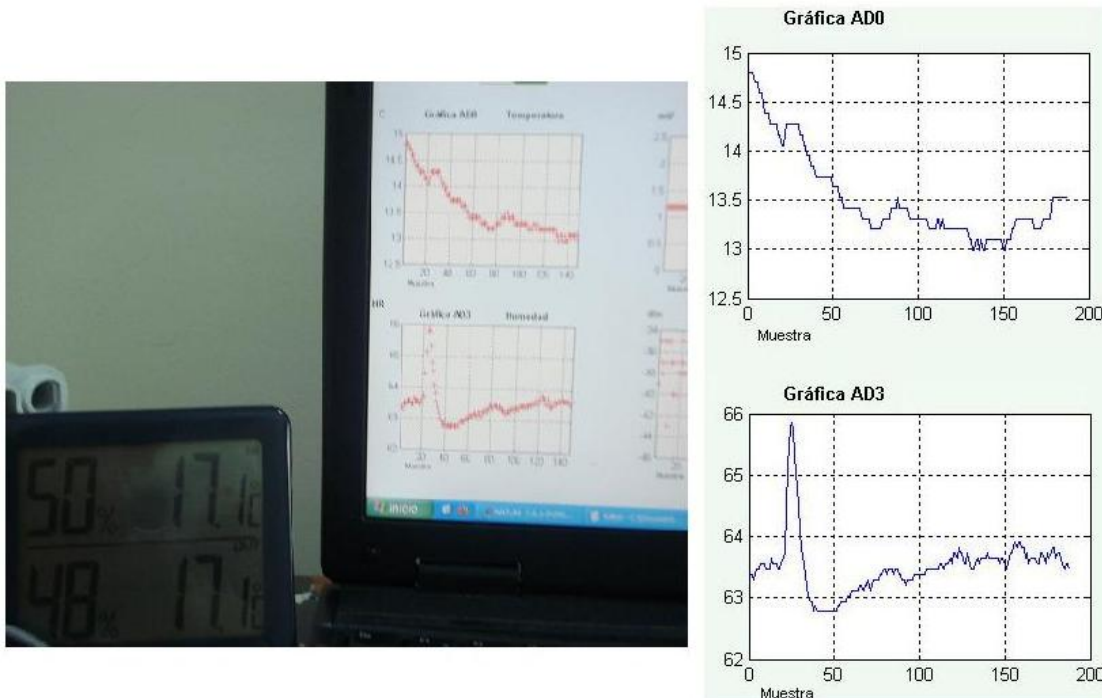
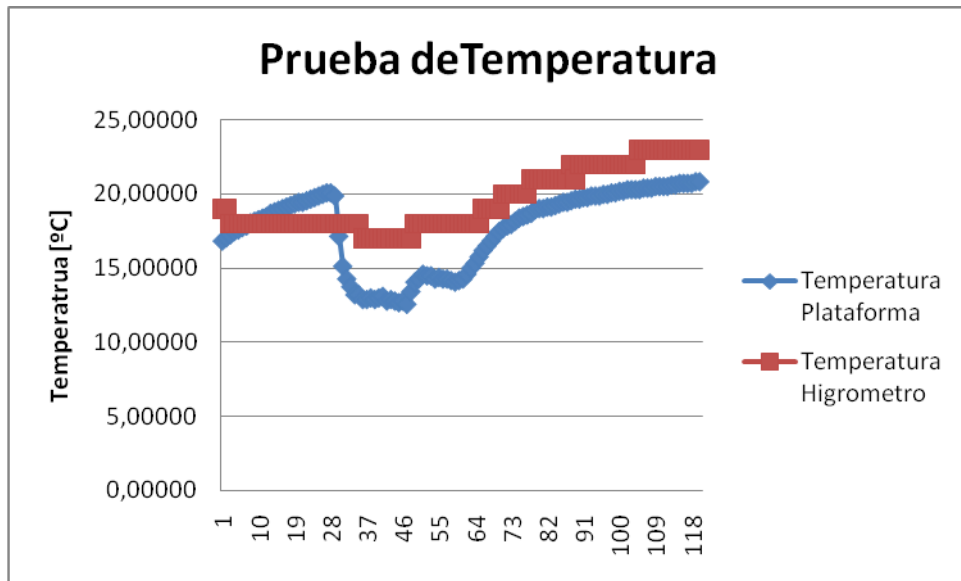


Figura 4.9 Visualización de los datos de temperatura y humedad con sus unidades correspondientes.

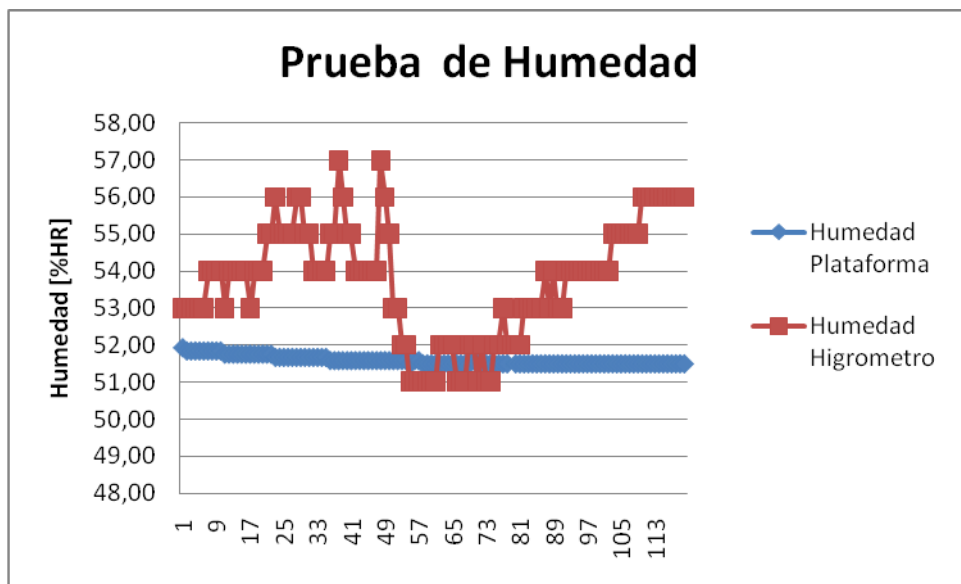
Fuente: Autor.

En la figura 4.9, se puede observar que el valor mostrado en la grafica (transmitido por la plataforma hardware) concuerda con el valor medido por el higrómetro y corresponde a la temperatura establecida en el lugar de la prueba.

La figura 4.10, muestra el resultado de haber desarrollado la prueba descrita en esta sección durante sesenta minutos. Con esto se busca exaltar la precisión con que son mostrados los datos en la interfaz desarrollada. En la grafica de color azul se muestran los datos tomados por la plataforma hardware equipada con sensores de temperatura y humedad, mientras que en la grafica de color rojo se muestran los datos presentados por el higrómetro. Los valores de temperatura y humedad fueron registrados con una diferencia de tiempo de treinta segundos.



a.



b.

Figura 4.10 a. Prueba de temperatura b. Prueba de humedad

Fuente: Autor

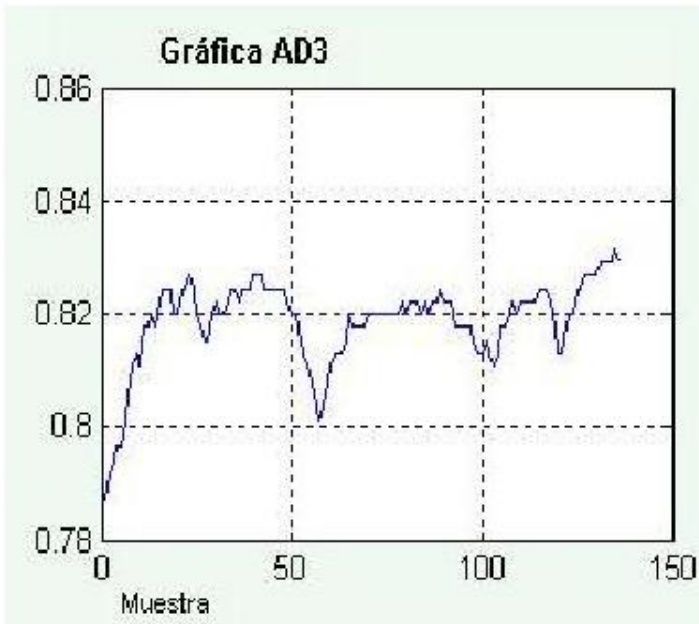
Durante las pruebas se presentaron errores de medida que deben tenerse en cuenta y que están relacionados con el hardware utilizado. Entre estos factores se pueden mencionar:

- La sensibilidad de los sensores y errores porcentuales de estos¹¹.
- El error cometido en las etapas de adecuación de señal debido a las tolerancias de los elementos electrónicos utilizados por la plataforma.
- El valor de los coeficientes A y B ingresados durante la configuración, ya que estos dependen de la caracterización y calibración realizada por los autores de la plataforma hardware empleada en la prueba.

La realización de una prueba sin el uso de los coeficientes que permiten visualizar los datos con sus unidades correspondientes, tiene como propósito observar el valor graficado en la interfaz con unidades de Volts y compararlo con la tensión medida en el puerto ADC del nodo remoto al cual está conectado el sensor. Con esto se verifica el correcto funcionamiento de la herramienta software creada y se evita presentar errores cometidos por la correcta calibración de los instrumentos (parecido a lo que se hizo con la prueba del generador de señal).

La figura 4.11 muestra los datos graficados y la tensión medida en los puertos ADC (donde están conectados los sensores de temperatura y humedad relativa) con un multímetro de referencia MY-64 fabricado por la empresa MASTECH, como se puede observar los valores de tensión son exactamente iguales.

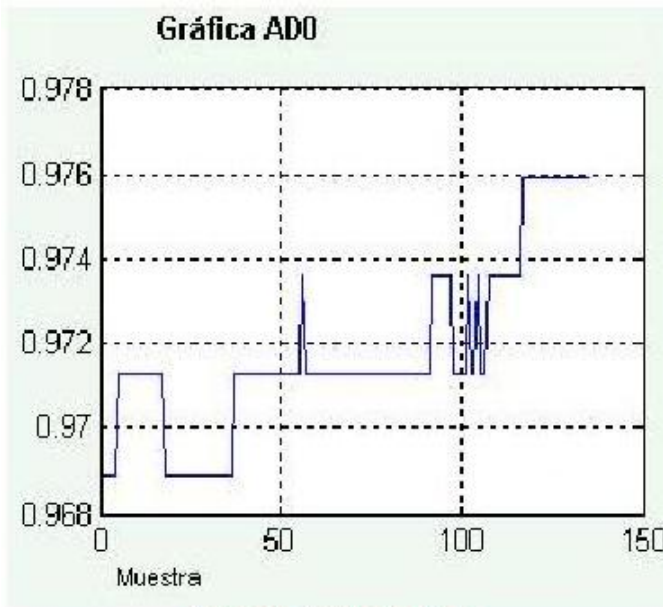
¹¹ Referirse al anexo A.



Medida del multímetro

Imagen ampliada de la interfaz gráfica

a.



Medida del multímetro

Imagen ampliada de la interfaz gráfica

b.

Figura. 4.11 Prueba sin el uso de coeficientes A y B. a. Medida del sensor de Humedad. b. Medida del sensor de Temperatura Fuente: Autor.

Con las pruebas anteriores se logró corroborar que no existen errores en tiempo de ejecución lo cual indica el buen funcionamiento del software implementado. Al mismo tiempo se logró verificar que los datos medidos por un nodo remoto son visualizados de forma precisa en la interfaz gráfica, denotando la coherencia de la aplicación.

La metodología descrita anteriormente empleada para la realización de la prueba con la herramienta software desarrollada en Matlab, fue aplicada al software hecho en JAVA buscando alcanzar objetivos completamente iguales. Los resultados se muestran a continuación.

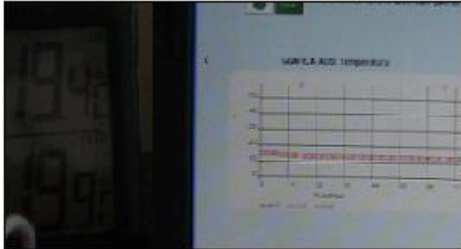
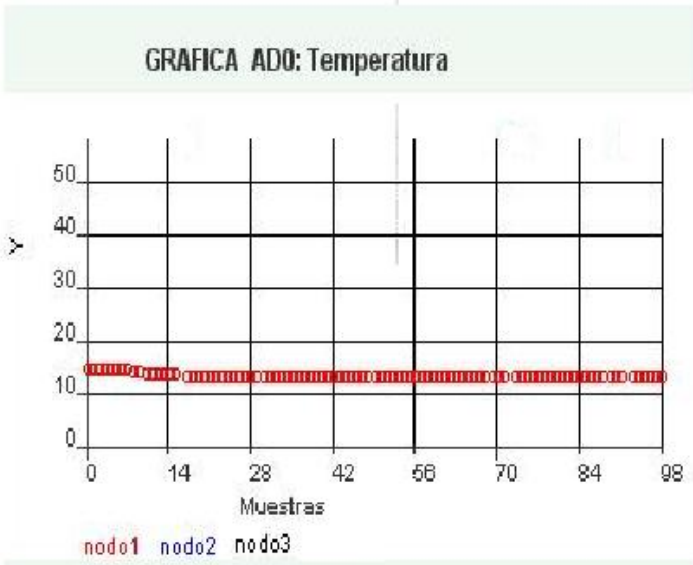
En este caso la temperatura del salón fue fijada en 20° C, tal como se muestra en la figura 4.12.



Figura 4.12 Temperatura del salón en el momento de la realización de las pruebas con Java.

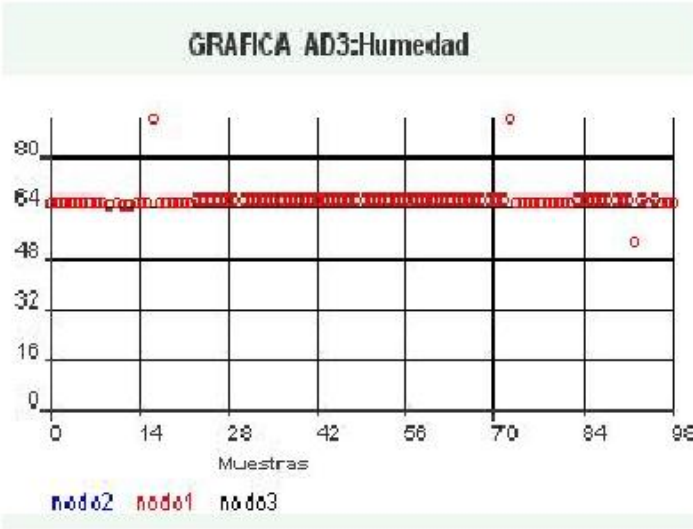
Fuente: Autor.

En la figura 4.13, se aprecian simultáneamente la medida de las variables de interés tomada por el higrómetro y los datos graficados en la interfaz desarrollada. Al comparar las formas de las señales de los datos graficados, se verifica la equivalencia y veracidad de estos. En este caso el valor medido para la temperatura es de 15°C y el valor de la humedad es de 64 %HR.



Medida con el higrometro

a.



Medida con el higrometro

b.

Figura 4.13 Imágenes de la prueba a. Medida del sensor de Temperatura b. Medida del sensor de Humedad.

Fuente: Autor.

En la figura 4.14 se muestran los resultados de la realización de la segunda parte de la prueba ajustada a la interfaz grafica desarrollada con java. En estas imágenes se observan los datos graficados en unidades de volts, lo cual corresponde a no alterar los valores que trae por defecto la interfaz de configuración de la sesión de trabajo para los coeficientes A y B.

Los datos visualizados en la interfaz grafica corresponden a la tensión de salida de los sensores y son verificados al compararlos con la medida suministrada por otro instrumento de medición (multímetro). La figura 4.14a y 4.14b, corresponden a la tensión de salida del sensor de temperatura y del sensor de humedad, respectivamente. En este caso la tensión registrada para la variable temperatura corresponde a 1 V y la tensión medida para la variable humedad relativa fue de 0.75 V.



a.

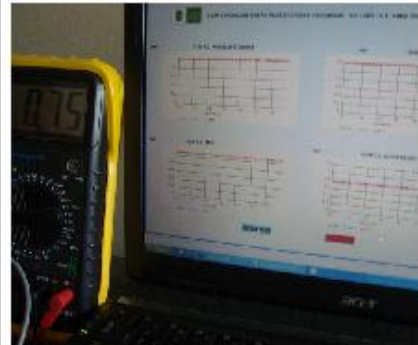
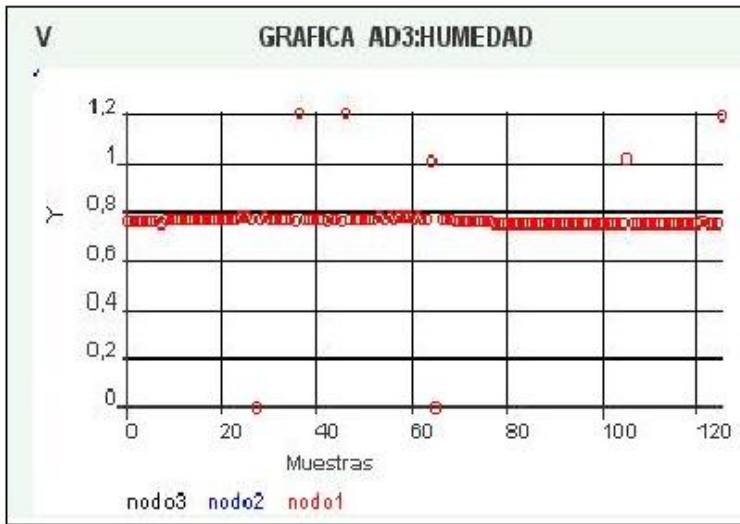


Imagen ampliada de la interfaz gráfica

b.

Figura 4.14 Imágenes de la prueba sin el uso de coeficientes A y B a. Medida del sensor de temperatura b. Medida del sensor de humedad

Fuente: Autor.

Con la realización de las pruebas anteriores se logro verificar la coherencia con que son graficados los datos en la interfaz grafica desarrollada con java. También se muestra como puede ser aprovechado el desarrollo de este proyecto en aplicaciones que tengan como objetivo monitorear variables medidas a través de sensores analógicos.

Como se mencionó anteriormente, los errores apreciados en las graficas anteriores, son producto de fenómenos ajenos al diseño del software entre los que se encuentran errores de calibración de la plataforma hardware, errores de sensibilidad y precisión de los sensores empleados, además de errores adquiridos en el proceso de digitalización de la señal recolectada (numero efectivo de bits).

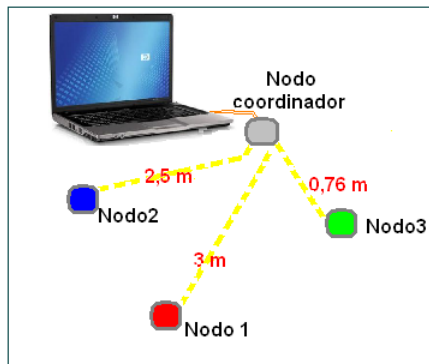
4.3 PRUEBA CON TRES NODOS REMOTOS

En el diseño del software de procesamiento de datos de módulos Xbee, se tuvo en cuenta la capacidad que tienen dichos módulos de establecer redes conformadas por varios nodos remotos y un nodo coordinador. Lo anterior muestra que una de las características importantes del software, es la posibilidad de manejar datos provenientes de varios nodos remotos y presentarlos de forma organizada, clara y precisa.

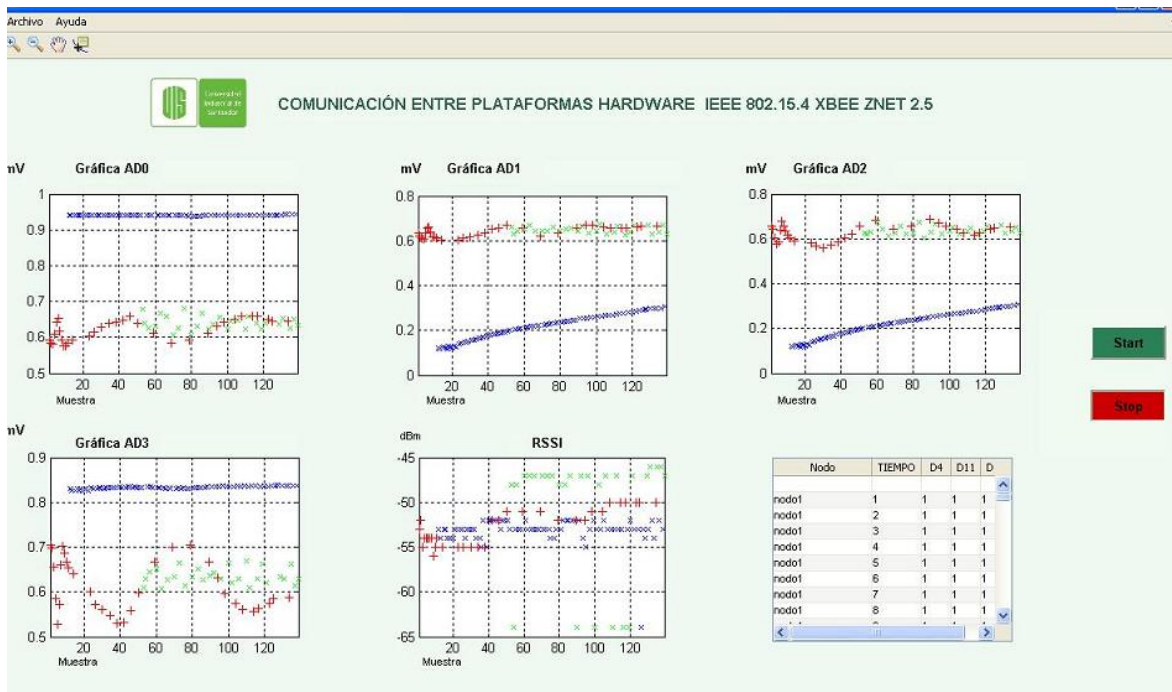
Para comprobar que efectivamente el software es capaz de trabajar en un ambiente de red, se realizó una prueba en el salón social del grupo de investigación CPS que consistió en configurar tres nodos remotos para que transmitieran información a un nodo concentrador de manera simultánea. Los tres nodos remotos se encontraban localizados a distancias diferentes del nodo concentrador y su disposición en el lugar de pruebas es similar al mostrado en la figura 4.15a. Cabe anotar que los nodos fueron introduciéndose uno después de otro (en forma escalonada) para observar las reacciones del software ante estos cambios de entorno y para observar también los cambios en el comportamiento de la red.

Los datos de la red establecida para la prueba, los cuales estaban siendo recolectados por el nodo concentrador, fueron graficados en un computador en el que se ejecutó la interfaz gráfica desarrollada en Matlab, en la figura 4.15b se muestra el resultado de esta prueba. Nótese que la información correspondiente a cada nodo está diferenciada por colores; la información para el nodo1 se visualiza en color rojo, la del nodo 2 en color azul y la del nodo 3 en color verde. La diferencia espacial entre cada uno de los nodos y el nodo concentrador está evidenciada por los valores de la potencia de señal recibida (RSSI) graficada en una de las ventanas de visualización de la interfaz.

Es interesante observar el RSSI del nodo más alejado mejoró sustancialmente con la inclusión de los otros dos nodos. Esto se debe a que estos módulos, cuando forman redes, mejoran su desempeño gracias a los protocolos de enrutamiento y de control de acceso al medio definido en el estándar IEEE 802.15.4.



a.



b.

Figura 4.15 a. Disposición de los nodos en el salón social del grupo CPS
b. Imagen de la interfaz grafica en el momento de la prueba.

Fuente: Autor.

4.4 PRUEBA CON EL ANALIZADOR DE RED

Hasta el momento se ha confiado plenamente en que todos los datagramas analizados por el software desarrollado, hacen parte de la trama API utilizada por los módulos Xbee para transmitir datos concernientes a información recolectada por sus puertos de conversión analógica digital¹². Sin embargo es posible utilizar un analizador de red (SmartRF Protocol Packet Sniffer v2.1.2.0-Chipcon Packet Sniffer for IEEE 802.15.4 MAC and ZigBee v1.0 NWK/APS), sintonizado justamente en el canal configurado por el nodo coordinador, para capturar el intercambio de información que se presenta en la comunicación de la red IEEE 802.15.4 formada por los módulos antes mencionados.

Para la realización de la prueba que mostró con detalles la forma en que los módulos Xbee se comunican, se contó con un analizador de redes IEEE 802.15.4 que contiene como dispositivo central un radio de referencia CC2420EM y una tarjeta de desarrollo CC2400EB, fabricado por la empresa Chipcon (actualmente propiedad de Texas Instruments)[30], el cual opera justamente en la banda de 2.4 GHz (misma banda de operación de los módulos Xbee). Obviamente fueron empleados dos módulos Xbee, uno configurado con firmware de coordinador y otro configurado con firmware de nodo remoto, además de un computador para ejecutar las aplicaciones de visualización tanto de las tramas capturadas con el Sniffer como de los datos graficados por el software desarrollado en este proyecto de investigación.

Con la ejecución de esta prueba se pudo evidenciar que durante la comunicación de los módulos Xbee, no solo se intercambia información referente a la aplicación específica que se les ha configurado (capturar muestras de los canales ADC), sino

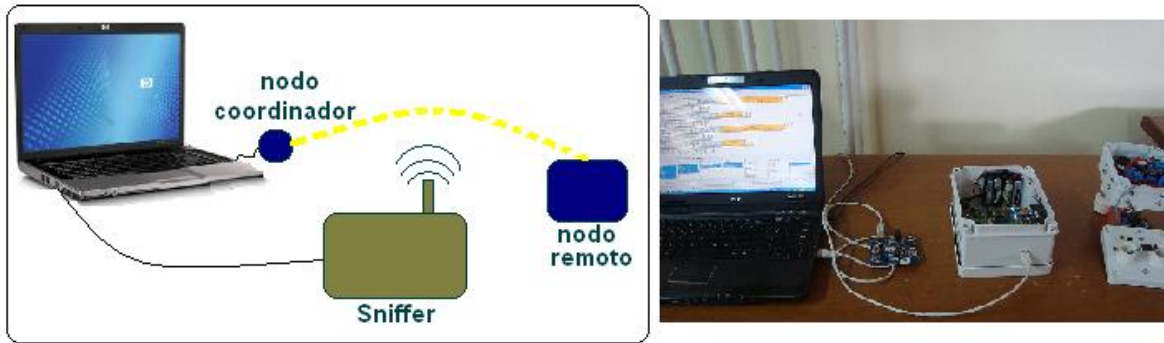
¹² Para más información sobre la trama API, referirse al capítulo 2.

que también intercambian información transparente para el software, relacionada con enrutamiento de paquetes y control flujo, propio del estándar IEEE802.15.4.

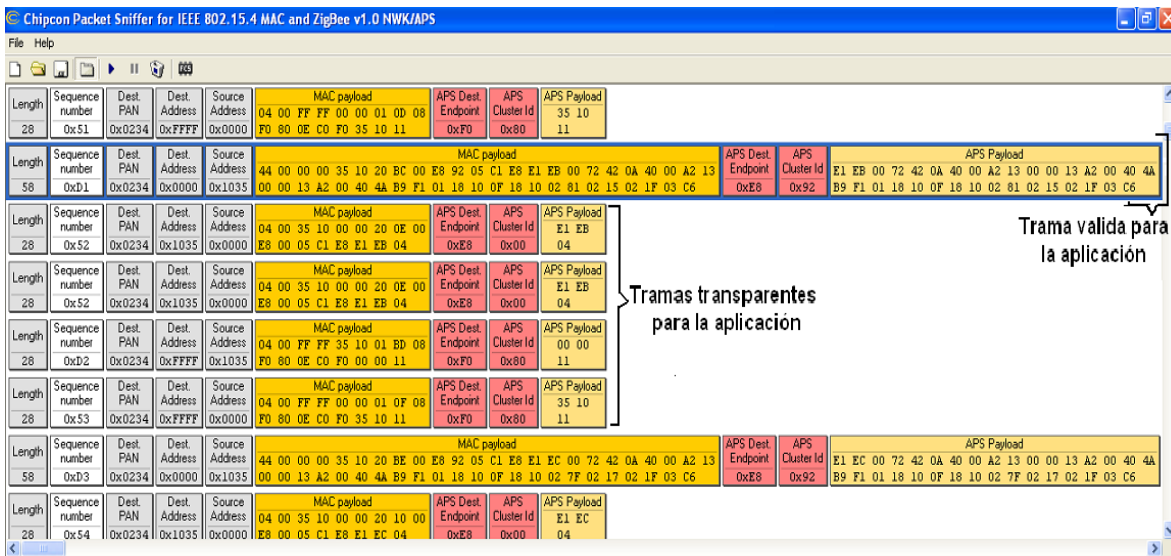
En la figura 4.16a se muestra un computador, un analizador de red y dos módulos Xbee; equipos que fueron utilizados en el desarrollo de la prueba anteriormente descrita.

En la figura 4.16b se presenta la ventana de captura utilizada por el analizador de red para mostrar la información recolectada. En esa figura se aprecian ocho líneas bien diferenciadas, cada una de esas líneas es una trama capturada. Los campos de las tramas están bien diferenciados por colores y nombres. En esta grafica se presentan los siguientes campos: longitud, número de secuencia, dirección de red, dirección destino, dirección fuente, campo MAC, direccionamiento ZigBee (dividió en clusterID y end points) y campo de aplicación.

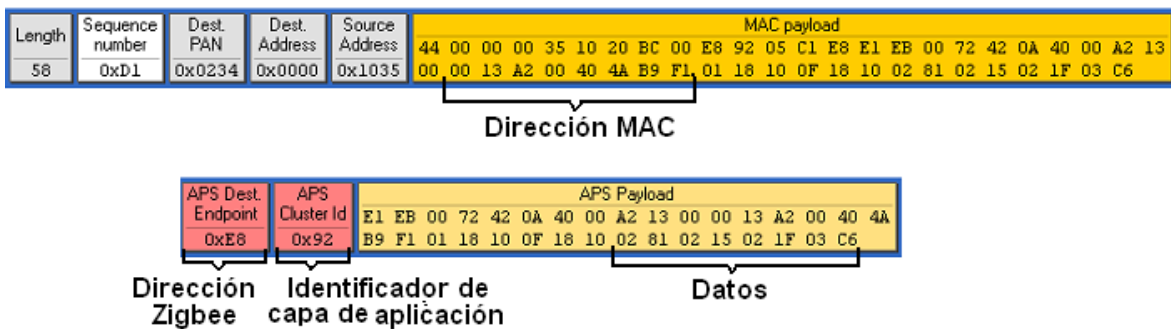
Como se aprecia la dirección de red, tiene el valor 0x234 para todas las tramas mostradas, indicando que el nodo remoto está asociado al nodo coordinador conectado al PC. Las direcciones de capa de control de acceso al medio (MAC), que hacen parte de la comunicación son la dirección 0x0000 que corresponde al coordinador (todo coordinador tiene esa dirección) y la dirección 0x1035 que corresponde al nodo remoto; mientras que las direcciones de capa de aplicación (realizadas a través de end points y cluster ID) que intervinieron en la comunicación fueron los end points F0 y E8 (correspondientes al end point reservado por el estándar y al end point de datos respectivamente) y a los cluster ID 0x92, 0x80 y 0x00 (correspondientes a datos recolectados de los ADC y a end points reservados por el estándar respectivamente). Como vemos, las tramas que contienen dirección de capa de aplicación 0x92, son las que contiene datos relativos a muestras de los puertos ADC y por tanto son las tramas de interés para el software. El resto de tramas hacen parte de la comunicación necesaria para mantener y establecer la conexión entre dos nodos [7].



a.



b.



c.

Figura 4.16 a. Computador, analizador de red y módulos Xbee utilizados en el desarrollo de la prueba b. Ventana de captura utilizada por el analizador de red para mostrar la información recolectada c. Trama válida para la aplicación Fuente: Autor

A manera de resumen se presenta la tabla 4.3. En esta se describen los logros asociados a cada una de las pruebas realizadas con cada una de las plataformas desarrolladas.

PRUEBA	Resultados	JAVA	MATLAB
Generador de señales	Se logró verificar que los datos medidos por un nodo remoto son visualizados de forma precisa en la interfaz grafica, denotando la precision y coherencia de la aplicación.	X	X
	El periodo de muestreo mínimo con el cual se deben configurar los nodos remotos es igual a 172ms.	X	X
Sensores	Se demostró la utilidad de la herramienta en aplicaciones relacionadas con monitorización de variables ambientales.	X	X
Varios nodos remotos	Se demostró que la interfaz gráfica es capaz de manejar y presentar información proveniente de tres fuentes de datos en forma simultánea y que además es posible ajustarla de acuerdo a la conveniencia de la aplicación	X	X
Analizador de red	Se verificó que los módulos Xbee utilizan el estándar IEEE 802.15.4 para realizar la comunicación y el establecimiento de las redes		
	A pesar que los módulos utilizan diversas tramas para el establecimiento de la comunicación inalámbrica, la aplicación es indiferente a las tramas que no contengan datos relacionados con muestras de los puertos ADC.		

Tabla 4.3 Resumen de los resultados de las pruebas

Fuente : Autor

Capítulo 5.

CONCLUSIONES Y OBSERVACIONES

Al inicio de la investigación se planteó como objetivo general “*Desarrollar una herramienta software (una interfaz Hombre-Máquina) que permita visualizar y almacenar datos provenientes de la comunicación entre plataformas hardware IEEE 802.15.4 X-Bee ZNet 2.5, en una base de datos local.*”

Al término de esta investigación se puede concluir que:

5.1 CONCLUSIONES

1. La herramienta de visualización desarrollada tanto en el lenguaje de programación orientada objetos Java, como en el entorno computacional Matlab; no solo permite almacenar y observar el comportamiento o tendencia de los datos capturados por los nodos remotos de una red conformada por dispositivos Xbee, sino que también cuentan con diversas funciones que brindan al usuario la oportunidad de interactuar con dicha herramienta y ajustarla a sus aplicaciones.
2. El software desarrollado, otorga mayor robustez a cualquier aplicación en redes de sensores inalámbricos que lo utilice, ya que cuenta con la habilidad de almacenar datos en tiempo real (tiempo de ejecución), con lo que se evita la pérdida de información valiosa.
3. Se logró desarrollar un producto que cumple con características importantes en cuanto a facilidad de manejo, gracias a la existencia de videos tutoriales

que suministran información útil para el aprovechamiento de toda la potencialidad que este ofrece.

4. Se justifica la implementación de la herramienta en el lenguaje de programación Java, porque este permite que el software desarrollado sea portable, es decir que pueda ser ejecutado en diferentes tipos de plataformas como Linux, Windows y MacOS. Además, Java tiene un modelo de objetos simple y elimina herramientas de bajo nivel que suelen inducir errores. Por último hay que resaltar que Java es un software libre que admite el desarrollo, modificación y distribución; de aplicaciones implementadas con él.
5. Se justifica la realización de la herramienta software en el entorno computacional Matlab, permitió contemplar de mejor manera las características requeridas para que la lógica de la herramienta funcionara perfectamente. Además se aprovechó el conocimiento previo en el uso de este entorno computacional adquirido durante todo el proceso de formación profesional. A pesar de ser un lenguaje de programación mundialmente utilizado, no es recomendable desarrollar aplicaciones en él, debido a las restricciones de licencia que posee.
6. El desempeño de la herramienta desarrollada es excelente si se tiene en cuenta que es capaz de procesar correctamente todos los datos provenientes del nodo concentrador de la red (de módulos Xbee) que la utiliza.
7. El sistema mostró un gran desempeño y robustez frente a errores de transmisión y recepción de datos a través de la comunicación serial, ya que es capaz de diferenciar entre tramas dañadas y tramas con todos sus campos completos.

8. La herramienta desarrollada contribuyó a cuantificar limitaciones de los módulos Xbee relacionadas con el proceso de encuesta de sus terminales de conversión analógica digital. Lo anterior establece un límite a la cantidad máxima de datos que serán procesados por el software en un periodo de tiempo determinado (periodo mínimo de muestreo (T_s) 172 [ms]).
9. Desafortunadamente, no fue posible lograr la interacción entre la herramienta software desarrollada y las plataformas hardware Xbee Sensor Adapter y Watch Port Sensors (fabricados por la empresa Digi). Estas plataformas, a pesar de usar módulos Xbee Znet 2.5 para la transmisión de datos, utilizan una configuración especial de firmware que impide que sus módulos sean compatibles con los demás Xbee diseñados para propósitos generales, sobre los cuales está basada toda la lógica de la herramienta desarrollada.

5.2 OBSERVACIONES

1. Es conveniente someter el sistema a una fase de diseño de software que agregue estética y confort visual al momento de utilizarlo.
2. Dado que la cantidad de nodos de una red conformada por módulos Xbee es una variable aleatoria, fue necesario desarrollar una lógica que permitiera adecuar su implementación (tanto en java como en Matlab) a un número indeterminado de nodos de una red, dándole a la herramienta una categoría de expansible frente a las necesidades de la aplicación.

3. La herramienta desarrollada se muestra como un valioso instrumento en el trabajo de calibración y ajuste de plataformas hardware que utilicen módulos Xbee para transmitir información. Tal es el caso de la plataforma hardware utilizada para la realización de las pruebas de desempeño.
4. Es posible utilizar otro tipo de librerías gráficas desarrolladas en el lenguaje de programación java, con los que se permita agregar mucha más interactividad al usuario en el uso de la plataforma.
5. Si la herramienta software hubiese sido diseñada para visualizar los datos generados en una red constituida por plataformas Watch Port-Sensor Adapter, se restringiría el uso de esta, a aplicaciones en donde se utilicen exclusivamente productos diseñados por la empresa Digi. Limitando a los usuarios, en el desarrollo de aplicaciones propias y a la medida de sus necesidades. A pesar de eso es recomendable, en caso de ser necesario, adaptar la interfaz gráfica desarrollada para poder utilizarla con este tipo de plataformas.
6. Es posible aprovechar los avances realizados con este proyecto de investigación (y demás proyectos relacionados del grupo de investigación CPS) para aplicarlos en el desarrollo de una herramienta que permita utilizar diferentes plataformas hardware basadas en el estándar IEEE802.15.4.
7. El desarrollo de esta herramienta, permitirá continuar avanzando en el diseño e implementación de aplicaciones que contemplen sistemas de información mucho más completos de los que hasta el momento se encuentran en el grupo de investigación CPS, debido a que este aporte permite que los datos recolectados en una red inalámbrica de sensores sea presentada a manera de información para el usuario final.

REFERENCIAS BIBLIOGRÁFICAS

[1] De La hoz Héctor, Castro Alexander “Diseño e implementación de una plataforma de red de sensores inalámbricos para la monitorización de variables físicas ambientales en un galpón avícola”. UIS 2008.

[2] Rugeles José. “Caracterización de una red ZigBee en ambientes abiertos y cerrados”. UIS 2007.

[3] Borja Edwin, Andrango Wílmer “Diseño e implementación de una red inalámbrica para sufragio electrónico basada en el estándar ZIGBEE (IEEE802.15.4)”. Escuela Politécnica Nacional. Julio 2009.

[4] Cepeda Gabriela, Guerra Jorge “Diseño e implementación de un sistema de seguridad contra intrusos para el laboratorio de informática utilizando el estándar IEEE 802.15.4 ZigBee como tecnología de comunicación”. Escuela Politécnica Nacional. Julio 2009.

[5] Khanh Tuan Le, Designing a ZigBee-ready IEEE 802.15.4-compliant radio transceiver.

[6] www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf

[7] Digi manual de usuario Xbee. www.digi.com

[8] Guía del usuario Xbee serie 1, Documento preliminar. www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf

[9] Maldonado Delgado Diego Armando “Caracterización de dispositivos de redes de sensores inalámbricos en ambientes indoor y outdoor” UIS

[10]<http://seclab.uiuc.edu/web/networked-sensors-and-control/amern-attested-metering-mergency-response-netwo.html>

[11] <http://www.faludi.com/code/xbee-api-library-for-processing/>

[12] <http://www.aisnet.org/award/bios/langefors.html>

[13] Langefors, Börje. *Theoretical Analysis of Information Systems*, Auerbach

[14] Sloan Career Cornerstone Center (2008). *Information Systems*. Alfred P. Sloan Foundation. Accessdate June 2, 2008.

[15] Ingeniería Eléctrica y Electrónica-DICCIONARIOS, México McGraw-Hill 2007

[16] Muñoz Luis, Jacome Erick, Plan de proyecto “Diseño, implementación y caracterización de una red de sensores Zigbee para la monitorización de contadores de energía digitales en una red de distribución de baja tensión empleando nodos Xbee”.UIS 2008.

[17] Murcia Barba José Manuel, “Puerto serie RS232”. ITIG

[18] www.digi.com/products/wireless/zigbee-mesh/xbee-series2-module.jsp

[20] Moreno Cynthia “Construcción de una red ZigBee prototipo para la adquisición de datos desde transmisores de corriente de dos hilos”. Escuela Politécnica Nacional. Marzo 2009.

[21] Reinoso Esteban “Diseño e implementación de un prototipo para una red de domótica y seguridad para un hogar utilizando el estándar IEEE 802.15.4 Zigbee”. Escuela Politécnica Nacional. Noviembre 2008.

[22] Caprile Sergio R. “CAN-088 Utilización del modo API en módulos Xbee 802.15.4”. Cika electrónica. Noviembre 2008.

[23] code.google.com/p/Xbee-api/wiki/Hardware

[24] Oyarce Andrés “Guía del Usuario”. Ingeniería MCI Ltda.

[25] <http://www.circuitsathome.com/mcu/programming/playing-xbee-part-4-api>

[26] www.ftdichip.com

[27] <http://qubits.wordpress.com/2009/04/04/esquematico-de-conexionado-y-montaje-de-modulos-xbee/>

[28] www.Digi.net “X-CTU Configuration & Test Utility Software”. Maxstream.

[29] http://es.wikipedia.org/wiki/Teorema_de_muestreo_de_Nyquist-Shannon

[30] <http://focus.ti.com/docs/toolsw/folders/print/packet-sniffer.html>.

ANEXO A

**HOJAS DE ESPECIFICACIONES TÉCNICAS DE LOS SENSORES Y
CIRCUITOS INTEGRADOS QUE HACEN PARTE DE LA PLATAFORMA
HARDWARE UTILIZADA EN LAS PRUEBAS Y DE LA PLATAFORMA XBEE
SENSOR ADAPTER Y WATCH PORT**

A1. Sensor de Temperatura de la plataforma hardware utilizada en las Pruebas



Low Voltage Temperature Sensors

TMP35/TMP36/TMP37

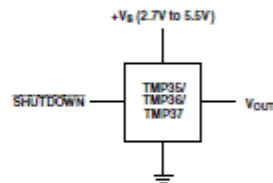
FEATURES

Low Voltage Operation (2.7 V to 5.5 V)
 Calibrated Directly in °C
 10 mV/°C Scale Factor (20 mV/°C on TMP37)
 ±2°C Accuracy over Temperature (Typ)
 ±0.5°C Linearity (Typ)
 Stable with Large Capacitive Loads
 Specified -40°C to +125°C, Operation to +150°C
 Less than 50 µA Quiescent Current
 Shutdown Current 0.5 µA Max
 Low Self-Heating

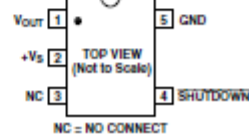
APPLICATIONS

Environmental Control Systems
 Thermal Protection
 Industrial Process Control
 Fire Alarms
 Power System Monitors
 CPU Thermal Management

FUNCTIONAL BLOCK DIAGRAM



**PACKAGE TYPES AVAILABLE
RT-5 (SOT-23)**



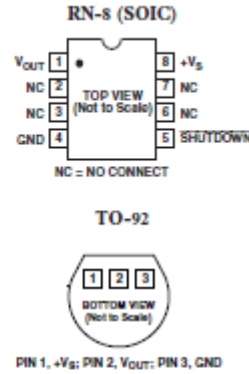
PRODUCT DESCRIPTION

The TMP35, TMP36, and TMP37 are low voltage, precision centigrade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (Centigrade) temperature. The TMP35/TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the -40°C to +125°C temperature range.

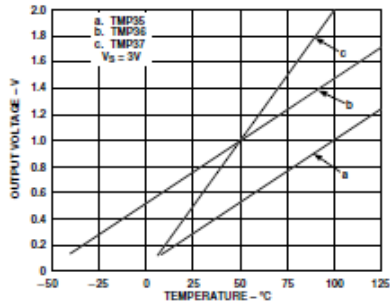
The low output impedance of the TMP35/TMP36/TMP37 and its linear output and precise calibration simplify interfacing to temperature control circuitry and A/D converters. All three devices are intended for single-supply operation from 2.7 V to 5.5 V maximum. Supply current runs well below 50 µA, providing very low self-heating—less than 0.1°C in still air. In addition, a shutdown function is provided to cut supply current to less than 0.5 µA.

The TMP35 is functionally compatible with the LM35/LM45 and provides a 250 mV output at 25°C. The TMP35 reads temperatures from 10°C to 125°C. The TMP36 is specified from -40°C to +125°C, provides a 750 mV output at 25°C, and operates to +125°C from a single 2.7 V supply. The TMP36 is functionally compatible with the LM50. Both the TMP35 and TMP36 have an output scale factor of 10 mV/°C. The TMP37 is intended for applications over the range 5°C to 100°C and provides an output scale factor of 20 mV/°C. The TMP37 provides a 500 mV output at 25°C. Operation extends to 150°C with reduced accuracy for all devices when operating from a 5 V supply.

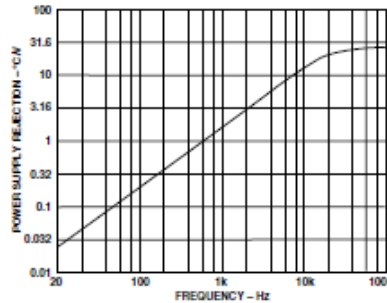
The TMP35/TMP36/TMP37 are all available in low cost 3-lead TO-92, SOIC-8, and 5-lead SOT-23 surface-mount packages.



TMP35/TMP36/TMP37 – Typical Performance Characteristics



TPC 1. Output Voltage vs. Temperature



TPC 4. Power Supply Rejection vs. Frequency

Ecuacion lineal que establece la relación existente entre la tensión entregada al un terminal ADC del modulo Xbee y la variable fisica medida por el sensor de temperatura:

$$T^{\circ}C = \underbrace{-45.45}_{A} V_{Xbee} + \underbrace{58.63}_{B}$$

Coeficientes

, donde VXbee es la tensión que llega al puerto ADC del modulo Xbee.

A2. Sensor de Humedad de la plataforma hardware utilizada en las Pruebas

Honeywell



Representative photograph, actual product appearance may vary.

Due to regional agency approval requirements, some products may not be available in your area. Please contact your regional Honeywell office regarding your product of choice.

HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Features

- Molded thermoset plastic housing with cover
- Linear voltage output vs %RH
- Laser trimmed interchangeability
- Low power design
- High accuracy
- Fast response time
- Stable, low drift performance
- Chemically resistant

Typical Applications

- Refrigeration
- Drying
- Meteorology
- Battery-powered systems
- OEM assemblies

Description

The HIH-4000 Series Humidity Sensors are designed specifically for high volume OEM (Original Equipment Manufacturer) users. Direct input to a controller or other device is made possible by this sensor's linear voltage output. With a typical current draw of only 200 μ A, the HIH-4000 Series is ideally suited for low drain, battery operated systems. Tight sensor interchangeability reduces or eliminates OEM production calibration costs. Individual sensor calibration data is available.

The HIH-4000 Series delivers instrumentation-quality RH (Relative Humidity) sensing performance in a low cost, solderable SIP (Single In-line Package). Available in two lead spacing configurations, the RH sensor is a laser trimmed, thermoset polymer capacitive sensing element with on-chip integrated signal conditioning. The sensing element's multilayer construction provides excellent resistance to most application hazards such as wetting, dust, dirt, oils and common environmental chemicals.

HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Product Specifications	
Package Style	Solderable SIP
Termination Details	2,54 mm [0.100 in] Lead Pitch
Series Name	HIH-4000 Series
RH Accuracy	± 3.5% RH, 0-100 % RH non-condensing, 25 °C, 5 Vdc supply
RH Interchangeability	± 5% RH, 0-60% RH; ± 8% @ 60-100% RH Typical
RH Hysteresis	± 3% of RH Span Maximum
RH Repeatability	± 0.5% RH
RH response time, 1/e	15 s in slowly moving air @ 25 °C
RH Stability	± 0.2% RH Typical at 50% RH in 1 Year
Supply Voltage	4.0 Vdc to 5.8 Vdc
Supply Current	500 µA Max.
Operating Humidity Range	0 to 100% RH, non-condensing
Operating Temperature Range	-40 °C to 85 °C (-40 °F to 185 °F)
Temperature Compensation	True RH = Sensor RH/(1.0305+0.000044T-0.0000011T ²) T in °C (True RH = Sensor RH/(0.9237-0.0041T+0.000040T ²) T in °C)
Availability	Global
Comment	Light sensitive, shield from bright light.
UNSPSC Code	411121
UNSPSC Commodity	411121 Transducers

Ecuacion lineal que establece la relación existente entre la tension entregada al un terminal ADC del modulo Xbee y la variable fisica medida por el sensor de humedad:

$$\%HR = \underbrace{-37.418}_{A} V_{Xbee} + \underbrace{94.0696}_{B}$$

donde V_{Xbee} es la tensión que llega al puerto ADC del modulo Xbee.

A3. Controlador de Línea serial 1-Wire™

DS2480

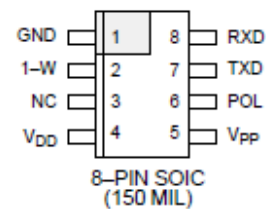
DALLAS
SEMICONDUCTOR

DS2480
Serial 1-Wire™ Line Driver

FEATURES

- Universal, common-ground serial port to 1-Wire™ line driver for MicroLAN™ applications
- Works with all jButtons and MicroLAN-compatible 1-Wire slave devices
- Communicates at regular and Overdrive 1-Wire speed and serial port data rates of 9600 (default), 19200, 57600 and 115200 bps
- Supports 12V EPROM programming and stiff 5V pull-up for Crypto jButton, sensors and EEPROM
- Self-calibrating time base with ±5% tolerance for serial and 1-Wire communication
- Slew rate controlled 1-Wire pull-down and active pull-up to accommodate long lines and reduce radiation
- User-selectable RXD/TXD polarity minimizes component count when interfacing to 5V based RS232 systems or directly to UARTs
- Programmable 1-Wire timing and driver characteristics accommodate a wide range of MicroLAN configurations at regular speed
- Smart protocol combines data and control information without requiring extra pins
- Compatible to optical, IR and RF to RS232 converters
- Low cost 8-pin SOIC surface mount package
- Operates over 4.5V to 5.5V from -40°C to +85°C

PIN ASSIGNMENT



PIN DESCRIPTION

GND	Ground
1-W	1-Wire Input/Output
NC	No Connection
VDD	4.5 to 5.5 Volts
VPP	Optional EPROM Programming Voltage
POL	RXD/TXD Polarity Select
TXD	Serial Data from UART
RXD	Serial Data to UART

ORDERING INFORMATION

DS2480S 8-pin SOIC

DESCRIPTION

The DS2480 is a serial port to 1-Wire interface chip that supports standard and Overdrive speeds. It connects directly to UARTs and 5V RS232 systems. Interfacing to RS232C ($\pm 12V$ levels) requires a passive clamping circuit and one 5V to $\pm 12V$ level translator. Internal timers relieve the host of the burden of generating the time-critical 1-Wire communication waveforms. In contrast to the DS9097(E) where a full character must be sent by the host for each 1-Wire time slot, the DS2480 can translate each character into eight 1-Wire time slots thereby increasing the data throughput significantly. In addition, the DS2480 can be set to communicate at four different data rates including 115.2 kbps, 57.6 kbps and 19.2 kbps with 9.6 kbps being the power-on default.

Command codes received from the host's crystal controlled UART serve as a reference to continuously calibrate the on-chip timing generator. The DS2480 uses a unique protocol that merges data and control information without requiring control pins. This approach maintains compatibility to off-the-shelf serial to wireless converters allowing easy realization of 1-Wire media jumpers. The various control functions of the DS2480 are optimized for MicroLAN 1-Wire networks and support the special needs of all current 1-Wire devices including the Crypto iButton, EPROM-based Add-Only Memories, EEPROM devices and 1-Wire Thermometers.

A4. Sensor de Temperatura de Watch Port



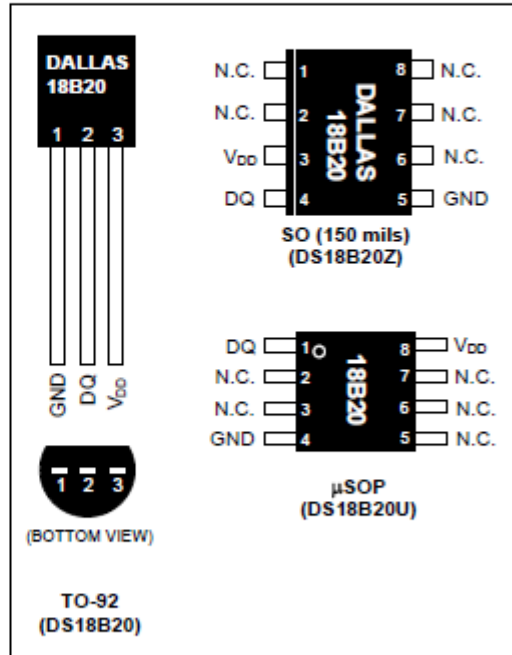
www.maxim-ic.com

DS18B20 Programmable Resolution 1-Wire Digital Thermometer

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ Accuracy from -10°C to $+85^{\circ}\text{C}$
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin μSOP , and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to $\pm 0.5^\circ\text{C}$ over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground.....	-0.5V to +6.0V
Operating Temperature Range.....	-55°C to +125°C
Storage Temperature Range	-55°C to +125°C
Solder Temperature	Refer to the IPC/JEDEC J-STD-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

A5. Controlador 1-Wire (Watchport / A sensor).

19-4675; Rev 7; 6/09



1024-Bit, 1-Wire EEPROM

General Description

The DS2431 is a 1024-bit, 1-Wire[®] EEPROM chip organized as four memory pages of 256 bits each. Data is written to an 8-byte scratchpad, verified, and then copied to the EEPROM memory. As a special feature, the four memory pages can individually be write protected or put in EPROM-emulation mode, where bits can only be changed from a 1 to a 0 state. The DS2431 communicates over the single-conductor 1-Wire bus. The communication follows the standard 1-Wire protocol. Each device has its own unalterable and unique 64-bit ROM registration number that is factory lasered into the chip. The registration number is used to address the device in a multidrop, 1-Wire net environment.

Applications

Accessory/PCB Identification
Medical Sensor Calibration Data Storage
Analog Sensor Calibration Including IEEE
P1451.4 Smart Sensors

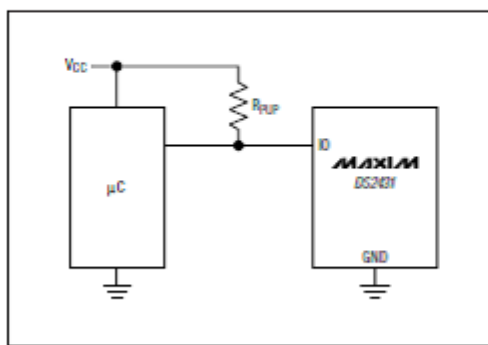
Features

- ◆ 1024 Bits of EEPROM Memory Partitioned Into Four Pages of 256 Bits
- ◆ Individual Memory Pages Can Be Permanently Write Protected or Put in EPROM-Emulation Mode ("Write to 0")
- ◆ Switchpoint Hysteresis and Filtering to Optimize Performance in the Presence of Noise
- ◆ IEC 1000-4-2 Level 4 ESD Protection ($\pm 8\text{kV}$ Contact, $\pm 15\text{kV}$ Air, Typical)
- ◆ Reads and Writes Over a Wide Voltage Range from 2.8V to 5.25V from -40°C to +85°C
- ◆ Communicates to Host with a Single Digital Signal at 15.4kbps or 125kbps Using 1-Wire Protocol
- ◆ Also Available as Automotive Version Meeting AEC-Q100 Grade 1 Qualification Requirements (DS2431-A1)

DS2431

Ordering Information

Typical Operating Circuit



PART	TEMP RANGE	PIN-PACKAGE
DS2431+	-40°C to +85°C	3 TO-92
DS2431+T&R	-40°C to +85°C	3 TO-92
DS2431P+	-40°C to +85°C	6 TSOC
DS2431P+T&R	-40°C to +85°C	6 TSOC
DS2431G+	-40°C to +85°C	2 SFN
DS2431G+T&R	-40°C to +85°C	2 SFN
DS2431Q+T&R	-40°C to +85°C	6 TDFN-EP* (2.5k pcs)
DS2431X+S	-40°C to +85°C	3x3 UCSPR (2.5k pcs)
DS2431X+	-40°C to +85°C	3x3 UCSPR (10k pcs)

Note: The leads of TO-92 packages on tape and reel are formed to approximately 100-mil (2.54mm) spacing. For details, refer to the package outline drawing.
+Denotes a lead(Pb)-free/roHS-compliant package.
T&R = Tape and reel.
*EP = Exposed pad.

Pin Configurations appear at end of data sheet.

ABSOLUTE MAXIMUM RATINGS

IO Voltage Range to GND-0.5V to +6V
IO Sink Current20mA
Operating Temperature Range-40°C to +85°C

Junction Temperature+150°C
Storage Temperature Range-55°C to +125°C
Soldering TemperatureRefer to the IPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

A6. Acelerometro Watch Port

Product Datasheet

Watchport® Sensors for Drop-in Networking

Environmental Monitoring Solutions

Watchport sensors provide a plug-and-play solution for Drop-in Networking applications. Each sensor connects to Digi's XBee sensor adapter (sold separately) for wireless connectivity to sensor data over a low power ZigBee network. Sensors are available to measure distance, proximity, temperature/humidity, acceleration/tilt, and the presence of water. Monitoring remote assets with Watchport sensors can preempt damage to facilities, reduce costly maintenance and warn of threatening environmental conditions.

When used in a Drop-in Network with Digi's ConnectPort X gateways, Watchport sensors can be monitored over a Wide Area Network (WAN) connection, providing centralized monitoring of geographically distributed sensor points. Plus, the Python® programming engine on the ConnectPort X provides the software tools to integrate multiple sensors and create custom applications using sensor readings.

Watchport sensors can be used in many Drop-in Networking applications. Monitor temperature and humidity levels in a refrigeration unit or grain silo; detect the presence of water in a livestock pen; activate a Watchport camera or an alarm when an access point is encroached; or reduce energy consumption by automatically powering down lights or other environmental systems when inhabitants are no longer detected.

Watchport/W Water Detector

- 2 gold plated pins detect wetness on a non-conductive surface
- Binary output with "Wet" or "Dry" readings
- Waterproof enclosure
- Maximum reading rate: 5 per second
- Power consumption: 15 mA (normal), 1.0 mA (suspend mode)
- Maximum CAT5 UTP cable length: 1,000 ft (304.8 m)

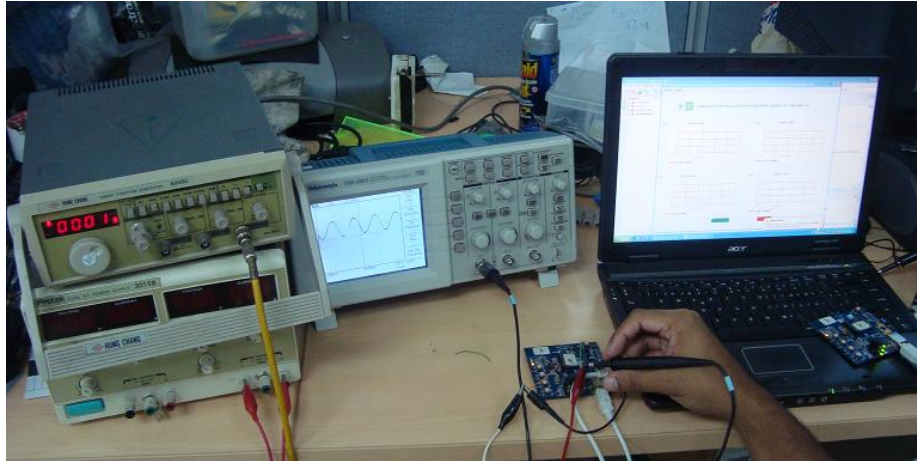
Watchport/A Accelerometer

- Dual axis accelerometer
- Measures dynamic acceleration such as vibration or static acceleration (gravity)
- Measures tilt to 0.1°
- Acceleration from -2g to +2g

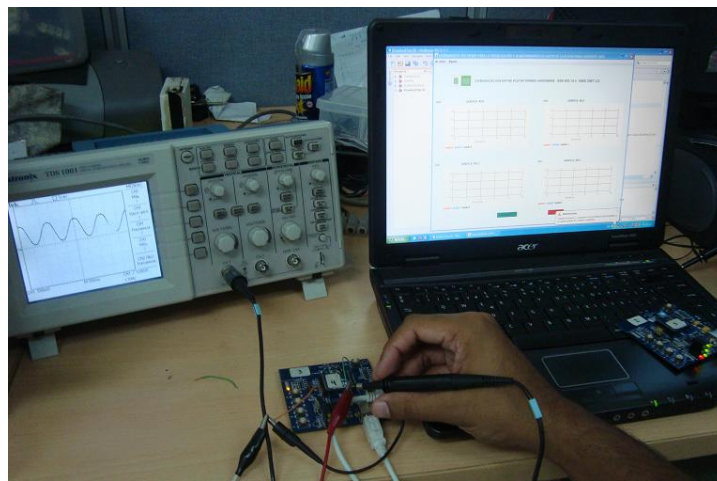


ANEXO B PRUEBAS DE DESEMPEÑO

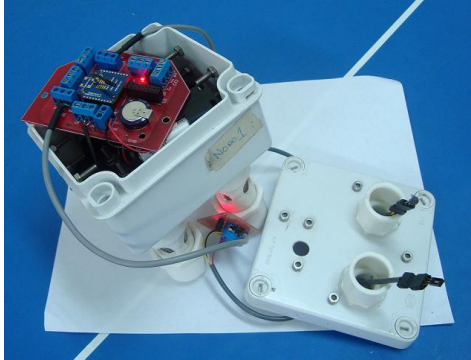
B1. Pruebas con el generador de señales.



Montaje para la prueba con el generador de señales



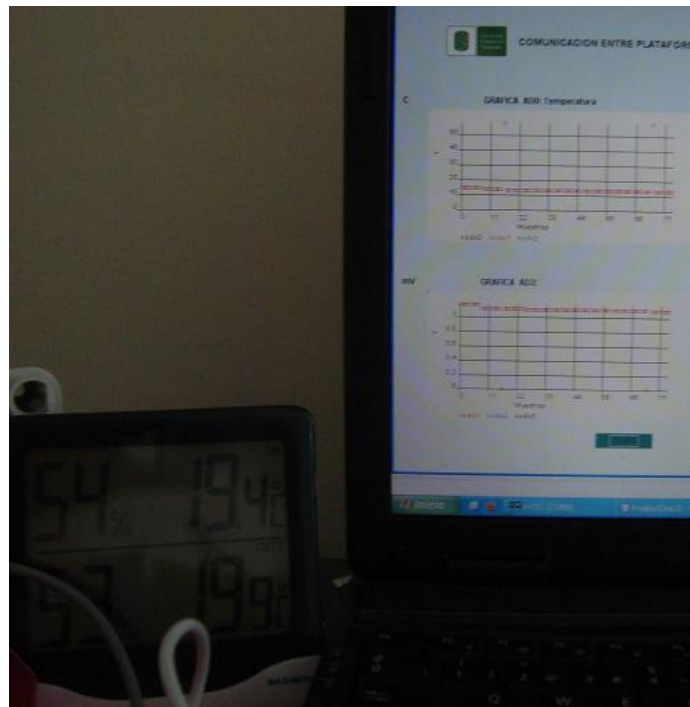
B2. Pruebas con los sensores de temperatura y humedad.



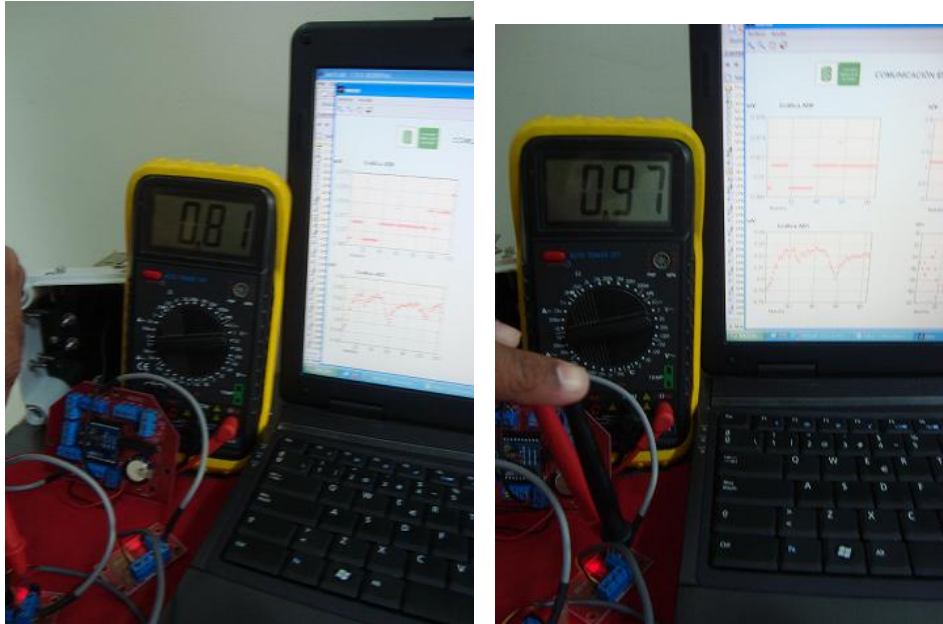
Plataforma hardware de la prueba



Lugar donde se desarrollo la prueba

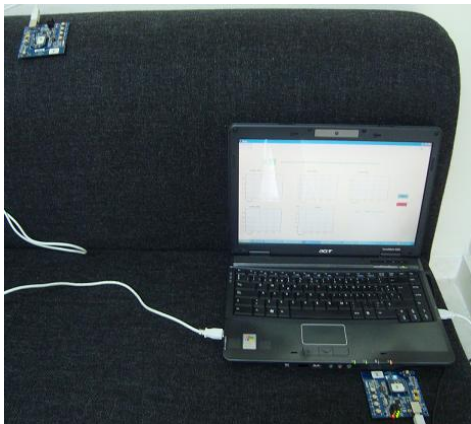


Medida de temperatura y humedad con el higrómetro



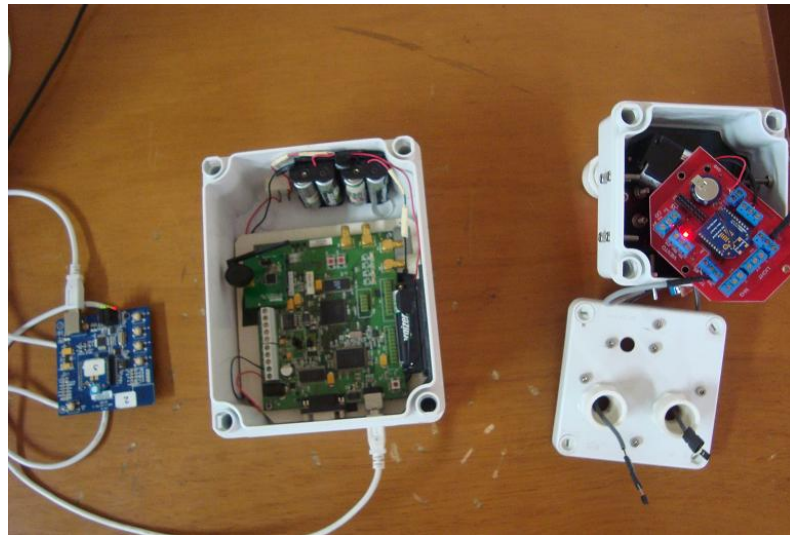
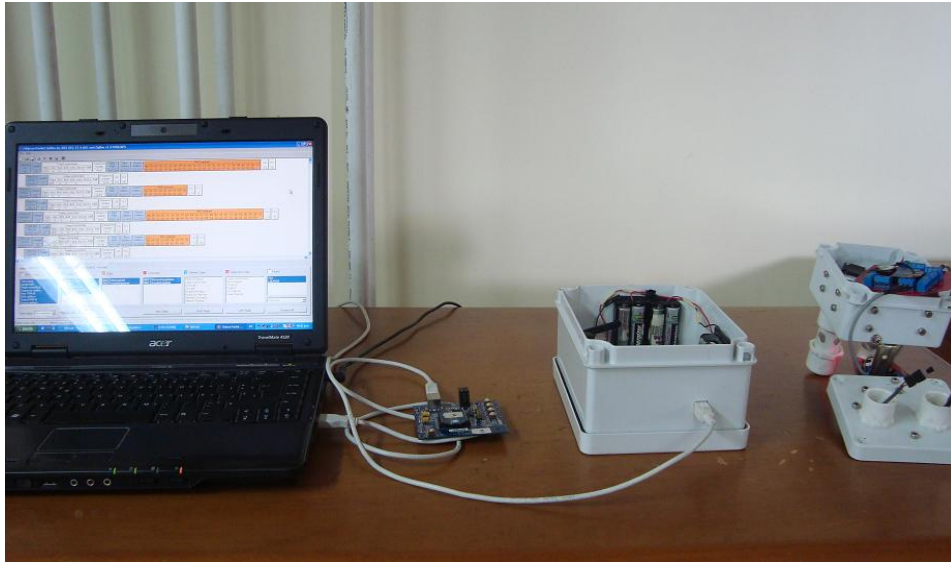
Medida con el multímetro de la tensión de salida del sensor de humedad y temperatura, respectivamente.

B3. Pruebas utilizando varios nodos



Red conformada por tres nodos remotos y un coordinador

B4. Pruebas con el sniffer.



PC, nodo coordinador, Sniffer y plataforma hardware con el nodo remoto

ANEXO C
DIAGRAMA DE FLUJO DEL PROGRAMA

