

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO



Estudio Exploratorio de dos Redes Inalámbricas de Sensores

POR:

GABRIEL HERNÁNDEZ SALCEDO

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

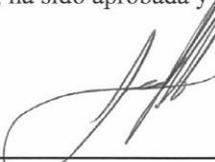
MAESTRO EN INGENIERÍA DE REDES MÓVILES

CHIHUAHUA, CHIH., MÉXICO

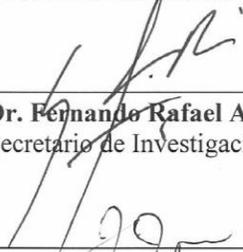
JUNIO 2018



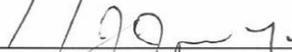
Estudio Exploratorio de dos Redes Inalámbricas de Sensores. Tesis presentada por Gabriel Hernández Salcedo como requisito parcial para obtener el grado de Maestro en Redes Móviles, ha sido aprobada y aceptada por:



M.I. Javier González Cantú
Director de la Facultad de Ingeniería



Dr. Fernando Rafael Astorga Bustillos
Secretario de Investigación y Posgrado



M.S.I. Karina Rocío Requena Yáñez
Coordinador(a) Académico



Dr. Fernando Martínez Reyes
Director de Tesis

Junio 2018

Comité:
Dr. Fernando Martínez Reyes
Dr. Luis Carlos González Gurrola
Dr. Raymundo Cornejo García
Dra. Graciela María Ramírez Alonso

© Derechos Reservados

Gabriel Hernández Salcedo
C. Jesús González Ortega # 5300 Granjas,
Chihuahua, Chih. México. C.P.3100

JUNIO 2018



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

22 de junio de 2018

L.S.C.A. GABRIEL HERNÁNDEZ SALCEDO

Presente

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Redes Móviles, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Fernando Martínez Reyes** para que lo desarrolle como tesis, con el título: **“ESTUDIO EXPLORATORIO DE DOS REDES INALÁBRICAS DE SENSORES”**.

ÍNDICE

Agradecimientos

Dedicatoria

Índice

Índice de Figuras

Resumen

I Introducción

II Revisión de literatura

III Metodología

3.1 Waspnote

3.2. Motes

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

IV Resultados

V Conclusiones

Referencias

Anexos

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

ATENTAMENTE
"Naturam subiecit aliis"

EL DIRECTOR

M.I. JAVIER GONZÁLEZ CANTÚ

EL SECRETARIO DE INVESTIGACIÓN
Y POSGRADO

FACULTAD DE
INGENIERÍA
U.A.CH.



DIRECCIÓN

DR. FERNANDO RAFAEL ASTORGA
BUSTILLOS

FACULTAD DE INGENIERÍA
Circuito No.1, Campus Universitario 2
Chihuahua, Chih., México. C.P. 31125
Tel. (614) 442-95-00
www.fing.uach.mx

Agradecimientos

Quiero agradecer primero, a quien ha sido clave fundamental, del privilegio que tengo hoy de ser profesionalista, sin tu apoyo no habría logrado entrar a la universidad, tu recuerdas aquellos días como eran las cosas, y no te decepcione, termine la carrera y considero que he cumplido hasta hoy como Licenciado en Sistemas. Pero nuevamente y después de muchos años de no vernos, vuelves a darme una segunda oportunidad y gracias a tu apoyo tengo la posibilidad de ingresar a la Maestría, la cual hoy con orgullo y satisfacción llega a feliz término.

Amigo eres un ejemplo a seguir, te exhorto a que cambies la vida de otros, como lo has hecho con la mía, en esta mitad que me resta por vivir, un enorme GRACIAS a ti Lic. Felipe de Jesús Fierro Alvídrez.

También deseo agradecer a mi Madre Gracia, y a mi Hermana Lucia, que sin saberlo han contribuido en muchas ocasiones sin siquiera saberlo, Madre, Hermana cada vez que se han preocupado por mí y mis hijas han logrado hacer de mi un mejor padre y sobre todo una mejor persona, gracias por siempre.

De igual manera deseo agradecer al Dr. Fernando Martínez Reyes, el haber transformado mi vida académica con sus enseñanzas, su confianza en mí, para retarme durante las clases de la maestría y gracias a ello lograr ser el estudiante que jamás antes fui, y por último su dedicación y paciencia para el logro de este trabajo que hoy culmina satisfactoriamente.

Por ultimo quiero agradecer a Dios por haber puesto a todos ellos en mi vida, pero principalmente por haberme dado dos hermosas creaturas que son los motivos que mueven y rigen mi vida, ellas son Laura y Daniela mis hijas.

Dedicatoria

Deseo dedicar este trabajo a la persona que en incontables ocasiones ha demostrado su amor así mi persona, a lo largo de mi andar por este mundo, pero que particularmente ha sido, fundamental en dos momentos trascendentes de mi vida, el primero en mi adolescencia acogiéndome en su casa, tras el divorcio de mis padres, y el segundo mucho más difícil, porque ya habían caído los años tanto en mí, como en ella, y a pesar de todo volvió a tomarme en su regazo y gracias a ello permitirme este logro que hoy se concreta, para mi tía Evangelina, quiero que sepas que no me alcanzara la vida para retribuirte todo lo que tú me has dado, te pido perdón por no estar a la altura de tu persona, pero quiero que sepas que siempre estarás en mi corazón, y que dios te premiara como tú te lo mereces.



Índice de Contenido

Agradecimientos	v
Dedicatoria.....	vi
Índice de Contenido.....	vii
Índice de Figuras.....	ix
Resumen	xi
Capítulo 1: Introducción	1
Capítulo 2: Revisión de literatura	4
Capítulo III: Metodología.....	14
3.1 WASPMOTE.....	14
3.1.1 Topologías de la red Zigbee	15
3.1.2 Configuración de los nodos de la red.....	17
3.1.3 Puesta en marcha de la red Waspmote	19
3.1.4 Programación de Módulos.....	22
3.1.5 Exploración de datos	25
3.2 MOTES.....	33
3.2.1 Introducción a MoteView	33
3.2.2 Monitoreo de datos con MoteView.....	38
3.2.3 Xserve como motor de MoteView	40
3.2.4 Uso independiente de Xserve	41
Capitulo IV: Resultados.....	48
V Conclusiones.....	55
Referencias	56
Anexos	59
Curriculum Vitae.....	86



Índice de Figuras

Figura 1.1 Tecnologías de la Banda de 2.4 GHz ⁴	2
Figura 2.0 Capas de Tecnologías de la Banda de 2.4 GHz ⁶	8
Figura 2.1 Aplicaciones ZigBee ⁷	10
Figura 2.2 Marco de Trabajo para una red inalámbrica de sensores. ⁸	11
Figura 3.0 Módulo Waspote con canal de comunicación Bluetooth y Zigbee.	14
Figura 3.1 Los modelos de topología de red ZigBee ¹⁰	16
Figura 3.2 Se muestra la tarjeta Gateway y un dispositivo ZigBee para su configuración. ¹¹	17
Figura 3.3 CTU configurado para acceder Módulo Xbee	18
Figura 3.4 Comunicación exitosa con el modem	18
Figura 3.5 MAC address Módulo Xbee ¹²	19
Figura 3.6 Configuración del Coordinador de la Red de Sensores	19
Figura 3.6.1 Modificación de valores de configuración nodo coordinador.....	20
Figura 3.7 Configuración Dispositivo final de la Red de Sensores	20
Figura 3.6.1 Modificación de valores de configuración nodo dispositivo final	21
Figura 3.8 Ensamble de XBee en la tarjeta Waspote	22
Figura 3.9 Imagen del IDE con programa abierto	23
Figura 3.9.0 Código cargado en el IDE para su ejecución en Waspote	24
Figura 3.10 Datos recibidos de la tarjeta waspote por medio de módulo zigBee	26
Figura 3.11 Meshlium ¹⁴	28
Figura 3.12 Dirección IP para acesar el System Manager del Meshlium ¹⁵	28
Figura 3.13 Pantalla de acceso al sistema de administración Meshlium ¹⁶	29
Figura 3.14 Pantalla principal del Administrador del Sistema Meshlium ¹⁷	30
Figura 3.15 Pantalla de configuración red de sensores. ¹⁸	31
Figura 3.16 Pantalla de captura de datos nodo 75ab ¹⁹	31
Figura 3.17 Pantalla de captura de datos nodo 7598 ²⁰	32
Figura 3.18 Captura de pantalla de la consola meshlium capturando datos varios nodos	32
Figura 3.19 Pantalla principal de la aplicación MoteView	33
Figura 3.20 Botón Acceso a Pantalla de Conexión de Sensores	34
Figura 3.21 Selector del modo de operación y adquisición de datos	35
Figura 3.22 Selector de la puerta de enlace y sus parámetros	36
Figura 3.23 Selección de la base de datos.	37
Figura 3.24 Selección de la aplicación y la tabla que almacenará los datos	38
Figura 3.24.1 Mensaje del servidor al hacer conexión en tiempo real	39
Figura 3.24.2 Mensajes del servidor actualizando la base de datos mts400_results.	39
Figura 3.25 Instalación de la Base de datos, PostgreSQL y su ODBC para windows	40
Figura 3.26 Capas del marco de trabajo de Red de Sensores Mote ²⁰	40
Figura 3.26.1 Conexión exitosa al xserve y a la base de datos por medio del puerto 9005	42
Figura 3.27 Muestra la Consola del Administrador de PostgreSQL, mostrando la B.D. task.	45
Figura 3.28. Cambio de tipo de TimeStamp a Date	46
Figura 3.29 Estructura de la tabla mts400_result con los nuevos campos.	46
Figura 3.30 Consulta (query) que inserta los datos a la tabla mts400_results.	47
Figura 3.31 Datos de la Tabla mts400_results, con la estructura y datos del proyecto.	47
Figura 4.0 Diagrama de Bloques de la Interfaz Gráfica	49
Figura 4.1 Seguimiento de los sensores activos de la red.	51
Figura 4.2 Ventana muestra la relación de datos filtrados el nodo 7818.....	52
Figura 4.3 Ventana de parámetros para la selección de la información.....	52
Figura 4.4 Reporte gráfico del promedio de temperaturas censado en el tiempo	53
Figura 4.5 Representación gráfica de nivel de voltaje a través del tiempo	54
Figura A.1 Tarjetas Waspote.....	59
Figura A.2 Características XBee.....	60
Figura A.3 Transmisión de Datos tipo emisión(Broadcast).....	64



Figura A.4 Tabla de Direcciones	66
Figura A.5 Transmisión XBee	68
Figura B.1 Instalación Modificada	70
Figura B.3 Venta de usuarios en el sistema operativo Windows 7.....	71
Figura B.4 Pestaña General Propiedades PostgreSQL.....	71
Figura B.5 Pestaña Miembros de PostgreSQL	71
Figura B.6 Usuario de la Base de datos task - tele.....	72
Figura B.7 Owner de la Base de datos task.....	73
Figura B.8 Usuario PostgreSQL30 ODBC	73
Figura B.9 Configuración del ODBC	74
Figura C.1 Menú Archivo nuevo solución, crear una solución nueva.	76
Figura C.2 Propiedades de la aplicación	77
Figura C.3 Selección de tipo de aplicación a generar.	77
Figura C.4 Árbol de la aplicación.	78
Figura C.5 Parte del código del procedimiento principal MAIN.	79
Figura C.6 Código embebido dentro de la aplicación.	81
Figura C.7 Llamada a la plantilla Power Run como evento de control al hacer clic en botón en línea	82
Figura C.8 Parámetros de la plantilla Power Run	82
Figura C.9 Código de recolección de datos de los nodos	84
Figura C.10 Configuración del temporizador de sensores en línea.....	84
Figura C.11 Eventos de ventana ejecución de temporizador	85
Figura C.12 Código embebido en el botón OK del Procedimiento Win Parameters.	85



Resumen

Con el rápido desarrollo tecnológico, las redes de sensores inalámbricas (WSN-wireless sensors network) se convertirán en la tecnología clave para la internet de las cosas (IoT-Internet of Things), las tecnologías de comunicación inalámbrica desempeñarán un papel importante y, en particular, las WSN harán proliferar muchas aplicaciones e industrias, estas WSN con una mejor flexibilidad de los dispositivos, a un costo razonable, gracias al incremento de la competencia en la fabricación de las mismas, lograran la integración del internet de las cosas.

El estudio y desarrollo de las WSN existentes, se pueden identificar y desarrollar nuevas aplicaciones potenciales para cumplir con la tecnología futura y las tendencias del mercado. Es posible hoy ejemplificar las redes de agua, de transporte, y hogares todas ellas inteligentes, generan una gran cantidad de datos, los cuales pueden ser explotados con diversos propósitos, quienes logren la combinación e integración de varias fuentes de información estarán en la posibilidad de nuevas oportunidades de negocios.

Debido a lo anteriormente expuesto, surge la motivación de investigar acerca de esta tecnología, el descubrimiento de la función básica de cada dispositivo, las configuraciones y parámetros óptimos para su operación, y de esta manera pasar a replicar este conocimiento particular, y así con cada individuo configurado, establecer la comunicación entre ellos, y así establecer una red de inalámbricos. Una vez alcanzados estos objetivos, se vuelve pertinente cumplir con el rigor de la investigación, y documentar los avances adquiridos, tanto para auto consulta, como para la comunidad universitaria, en particular, aquellos que desean profundizar en el conocimiento de esta tecnología.

La propuesta de este proyecto, nace precisamente del resultado de esta investigación con la que identificamos, que aunque la plataforma de lanzamiento sea la misma, existen diferentes fabricantes, cada uno con su propio diseño, y por supuesto la disponibilidad de los dispositivos para el desarrollo, así que al final se hizo una selección, sobre la cual se decidió hacer la propuesta de diseño e implementación de un software de apoyo para el registro y visualización de datos recolectados en una red inalámbrica de sensores. Esta red es desplegada en un área delimitada, la cual es monitoreada constantemente, para lograr identificar los cambios del medio ambiente. El uso de esta herramienta nos permitirá hacer un muestreo estadístico de las diferentes temperaturas, así como de la humedad relativa y la

presencia de luz, todo esto en conjunción con la hora de registro de estos eventos no dará la oportunidad de visualizar el comportamiento de algunas variables del ambiente. La interface de visualización permite también realizar diversos niveles de consulta por medio de presentaciones tabulares las cuales representan en forma exacta el ordenamiento lógico de la información numérica, así como la representación gráfica para una mayor rapidez en la comprensión de los datos. Esta herramienta justificara la toma decisiones con el soporte de los datos que producen información.

Los resultados de este proyecto pueden ser aprovechados por futuros trabajos, para reducir la curva de aprendizaje y permitir que estas investigaciones, tengan como resultado la aplicación práctica de esta tecnología y logren salir del ámbito universitario al público y comercial, permitiendo a los emprendedores generar oportunidades de negocios con esta tecnología.



Capítulo 1: Introducción

Actualmente para identificar dónde se encuentra alguna red industrial, agrícola, minera se requiere una cantidad importante de sensores los cuales permanentemente monitorean el estado que guarda la red en general, transmitiendo datos a la central de control para su procesamiento e interpretación y en base a la información obtenida al realizar alguna acción de control.¹

Cuando requerimos transmitir datos a una central de control se tienen un par de opciones: una es utilizar un tendido de cables, y la otra es utilizar un medio inalámbrico para realizar la transmisión. Entre las más importantes debemos hacer mención de: RFID y ZigBee para controlar sensores [1], Bluetooth para voz y datos y WIFI para redes inalámbricas^{2 3}.

Si la necesidad es transmitir una gran cantidad de datos un par de alternativas son WIFI o Bluetooth pero si solo hablamos de una velocidad baja de transmisión entonces tenemos las siguientes alternativas tecnológicas: RFID y 802.15.4. La primera es usada para el control de inventarios, la segunda es la llamada ZigBee, la cual es usada para transmisión inalámbrica a una tasa baja pero con alcance de media y larga distancia, una de las características importantes de esta tecnología es que opera en la bandas libres ISM (Industrial, Scientific & Medical) de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos) Tiene una velocidad de transmisión de 250 Kbps y un rango de cobertura de 10 a 75 metros. A pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado, esto debido a su baja tasa de transmisión y, a características propias del estándar IEEE 802.15.4. Cada red ZigBee tiene un identificador de red único, lo que permite que coexistan varias redes en un mismo canal de comunicación sin ningún problema. Teóricamente pueden existir hasta 16,000 redes diferentes en un mismo canal y cada red puede estar constituida por hasta 65,000 nodos, obviamente estos límites se ven truncados por algunas restricciones físicas (memoria disponible, ancho de banda, etc.).

¹ [URL1] <http://es.wikipedia.org/wiki/Automatizaci%C3%B3n>, "Automatización industrial" ² [URL2] <http://spanish.bluetooth.com/Bluetooth/Technology/>, "Página Principal de la organización bluetooth" de 2008. ³ [URL3] http://www.wi-fi.org/knowledge_center_overview.php, "Página Principal de la organización wi-fi".



En la figura 1.1 se muestra las diferentes tecnologías para comunicaciones inalámbricas, el eje X corresponde a la velocidad de transmisión y el eje Y a los diferentes tipos de redes. Estas son: Las (WPAN) área personal inalámbricas, (WLAN) redes de área local inalámbricas, (WWAN) redes de área amplia [2].

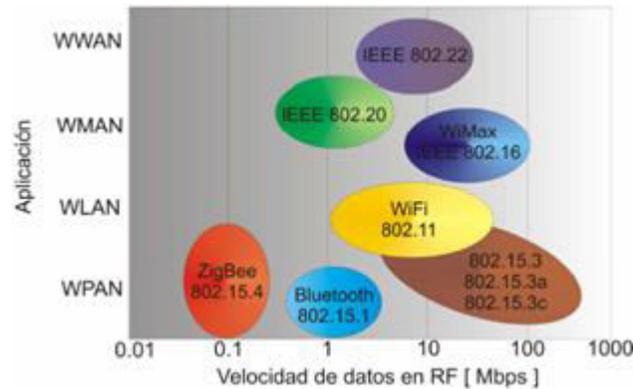


Figura 1.1 Tecnologías de la Banda de 2.4 GHz ⁴

Las redes tipo WPAN son las que se utilizan en el uso de aplicaciones de sensores, por esta razón son las que abordaremos en este trabajo.

Las redes de sensores tienen su origen como gran parte de la tecnología, en la milicia. Los Estados Unidos durante la guerra fría, se instaló una red de boyas sumergidas para detectar submarinos, utilizando sensores de sonido. La investigación de redes de sensores comenzó con el proyecto redes de sensores distribuidos (DSN, Distributed Sensor Networks), de la agencia militar de investigación avanzada de Estados Unidos (DARPA Defense Advanced Research Projects Agency). ⁵

Se ha dado un desarrollo en sensores inalámbricos gracias a los avances en la microelectrónica, la computación y las telecomunicaciones. El diseño de circuitos integrados de gran capacidad que pueden procesar señales digitales y análogas con la posibilidad de generar señales de transmisión en ondas de radiofrecuencia en módulos pequeños, que pueden ser colocados en espacios muy reducidos [3].

En los últimos años han surgido nuevas tecnologías encaminadas al monitoreo remoto de sensores, motivado por el bajo costo de los sistemas para la transmisión

⁵ [URL4] http://es.wikipedia.org/wiki/Red_de_sensores, "Red de sensores".



inalámbrica, ZigBee es una de ellas. Por tal motivo con el presente trabajo de tesis se pretende explorar su funcionalidad del protocolo, para poder utilizarlo por medio de una interfaz amigable para la interpretación de la información por el usuario, ya que es una tecnología que cada día viene teniendo una gran aceptación en el mercado.

En este trabajo se demuestra el uso de dos redes de sensores, en particular algunos elementos de configuración y se explora un prototipo para la visualización de datos recolectado desde una red de sensores.



Capítulo 2: Revisión de literatura

En virtud del gran desarrollo que se las redes de sensores han mostrado en los últimos años, este apartado realiza una revisión de algunos de los diferentes contextos de aplicación en los cuales estas han encontrado un nicho de aplicación.

Evaluación experimental de la tecnología ZigBee en ambientes interiores

La existencia en el mercado de diferentes dispositivos para redes inalámbricas tipo (WSN), obliga hacer la pregunta ¿En realidad funcionan lo que vendedor ofrece?, hasta el día de hoy existen muchos análisis de rendimiento para redes inalámbricas, como redes de topología estrella con ramificaciones (cluster-tree), para rendimiento industrial con aplicaciones OMNeT++ (componente de trabajo para construir simuladores de redes), para aplicaciones médicas se han realizado sistemáticas simulaciones. Sin embargo la mayoría del trabajo previo ha sido basado en simulación, en lugar de experimentos prácticos, como lo muestra este trabajo que evaluó el rendimiento de una red inalámbrica ZigBee de 51 nodos, por un extenso periodo de tiempo a lo largo de un pasillo y un cuarto.

Los resultados mostraron la eficiencia de estos dispositivos en tres aspectos, conectividad de los nodos, tasa de pérdida de paquetes, y tasa de transmisión de datos. Concluyendo que el desarrollo de plataformas ZigBee funciona muy bien en transmisiones multi-salto (multihop), por periodos de tiempo extendido [6].

Comunicaciones inalámbricas en la robótica

Un robot es una forma mecánica o virtual, estos necesitan una movilidad por lo tanto nos hace pensar en comunicaciones inalámbricas de tal manera que no se vea limitada la movilidad de las plataformas.

En la movilidad de los robots, se obliga al diseño sin cables para alimentación, comunicación y control, por lo que un factor importante al elegir tecnología de comunicación inalámbrica, es que sea de bajo consumo de potencia, además, si se emplea en grupos de investigación, es importante que sea de bajo costo.



Esta investigación le dio un valor agregado al proyecto, como la posibilidad de observar y almacenar la información en una base de datos, lo cual es muy útil para sensores que no son permanentemente monitoreados (ejemplo: Sensores ambientales) [7].

Monitoreo de variables ambientales en invernaderos

La necesidad de tener sistemas de ambiente controlados, es propiciada por la implementación de las redes de sensores inalámbricos, como lo demuestra este trabajo donde podemos encontrar una red de monitoreo de variables ambientales como la temperatura y la humedad relativa, con lo cual se puede reducir la influencia de los factores adversos, que limitan la producción y calidad de los cultivos. Otro aspecto importante, es el uso eficiente del agua, con una irrigación controlada y adecuada para cada cultivo. Por otro lado, tenemos que considerar la cobertura de varios cientos de metros cuadrados, que abarcan los invernaderos modernos, donde se elige con mucho cuidado la ubicación para las mediciones de temperatura, humedad y la iluminación, todo esto con el fin de mejorar la eficiencia de producción. Por todo esto se implementó el uso de una red inalámbrica de sensores (WSN).

El sistema de monitoreo basado en el estándar ZigBee, permitiendo una instalación casi ilimitada de sensores de todos tipo, que incrementa la robustez de la red y reduciendo considerablemente los costos de instalación. Al final ayudando a los productores a incrementar la cosecha, con una mejor calidad [8].

Despliegue de un sistema de control de luces en la vía pública basado en sensores inalámbricos

Los altos costos en el consumo de energía obligan a las instituciones públicas de gobierno a implementar mecanismos eficientes de ahorro de energía, en España el consumo de energía en la iluminación pública en algunas de sus municipalidades representa el 80% del total del consumo y el 60% del presupuesto de estas.

El objetivo de reducción a 75w por habitante por año, fue la meta planteada para la implementación de un sistema inteligente de control de luces públicas, utilizando



dispositivos ZigBee conectados a sus lámparas formado una red de malla (mesh), y agregando capacidad de sensor temperatura, humedad, luz directa, consumo de energía por lámpara, horas de vida etc., logrando medidas exitosas para dar formato a cambios ambientales y así el sistema controle de manera automática la operación y uso de la iluminación pública [9].

Vigilancia estructural de puentes

Podemos decir que su enfoque es hacia el diseño e implementación de una red para la adquisición de variables relacionadas con la vigilancia estructural de puentes, la evaluación se dio por 2 meses consecutivos llegando a unos resultados exitosos, desarrollando una plataforma de hardware y software escalable para contener decenas de sensores para realizar una evaluación mucho más compleja de la estructura [10].

Patologías en la construcción

Los autores expresan como objetivo la realización de un sistema sensorial inalámbrico para la detección de patologías en la construcción, buscando como resultado una red de detección de estas a partir de dispositivos embebidos, sensores y tecnologías Zigbee, con el aprovechamiento del bajo consumo de energía integrando el despertador en el propio módulo, con flexibilidad en la instalación y facilidad de mantenimiento [11].

Plataformas de Hardware

Los autores realizaron un estudio de plataformas hardware empleadas en redes de sensores inalámbricas con lo cual buscaba el crear redes de sensores que pudieran utilizar plataforma hardware de distintos modelos evaluando los parámetros más importantes como el consumo promedio, potencia de transmisión, sensibilidad, y protocolo implementado los cuales son determinantes para que el diseñador que plataforma se pueda adecuar a sus necesidades [12].

ZigBee



ZigBee es un conjunto establecido de especificaciones para el área de red inalámbrica personal (WPAN), es decir, las conexiones de radio digitales entre computadoras y dispositivos relacionados. El gran potencial de redes de sensores inalámbricos (WSN) que se está viendo en aplicaciones industriales, de consumo y de negocios.

Se trata de un protocolo inalámbrico que opera en la banda de 2,4 GHz, como Wi-Fi y Bluetooth, pero este opera a velocidades de datos mucho más bajas.

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital, de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (Wireless personal area network, WPAN) su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

Las principales ventajas de ZigBee inalámbrica son:

- Menos ancho de banda de canal de solamente 0.3-2Mhz Comparado Para 22MHz tomada por Wi-Fi
- Bajo consumo de energía
- Red muy robusta
- Hasta 65.645 nodos
- Muy fácil añadir o eliminar nodos de la red

Alianza IEEE y Zigbee

Como se ve en la Figura 2.0, el estándar IEEE 802.15.4 desarrolla la capa de MAC y la capa física, la cual direcciona cosas como las especificaciones de frecuencia y velocidad de datos. La capa física permite dos tipos de dispositivos, de función completa (FFDs) y los dispositivos para funciones reducidas (RFDs). Mientras tanto la alianza ZigBee desarrolla la capa de aplicación y la capa de red, que incluye el API y los servicios de seguridad. La capa de red y la capa de aplicación son más específicos que las otras capas e implican la forma como una



red ZigBee tiene que ser establecida y cómo los dispositivos de la red se conectan entre ellos.

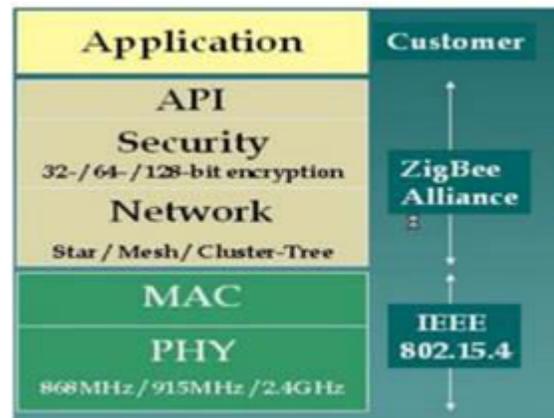


Figura 2.0 Capas de Tecnologías de la Banda de 2.4 GHz ⁶

Por lo tanto, ZigBee es de bajo costo, baja potencia, estándar de red de malla inalámbrica. El estándar ZigBee proporciona a la red, seguridad y los servicios de soporte de aplicaciones que operan en la parte superior del control IEEE 802.15.4 de acceso al medio (MAC) y la capa física (PHY) estándar inalámbrica figura 2.0.

Topologías de la red Zigbee

Los dispositivos ZigBee pueden transmitir los datos a través de largas distancias mediante el paso de datos a través de una red de malla de dispositivos intermedios para llegar a los más distantes. Módulos ZigBee se pueden utilizar para crear redes con diferentes topologías.

La Figura 2 indica las tres topologías principales en una WSN ZigBee.

El origen del nombre de ZigBee

Principio ZigBee:



- Los miembros de la colmena jerárquicamente estructuradas una reina, pocos zánganos, y miles de abejas obreras.

- ❑ La tecnología utilizada por las abejas para comunicar que se encontró una nueva fuente de alimento para Los demás miembros de la colonia.

⁶ <https://bit.ly/2IdLpev> .

- ❑ La abeja baila en un patrón de zigzag, el intercambio de información Tal
- ❑ como la ubicación, la distancia y la gestión de una fuente de alimento recién descubierta.

Organización ZigBee

1. No es estándar IEEE pero es un consorcio de industrias promoviendo Mercados de comunicaciones de velocidad de datos de baja transferencia.
2. 8 empresas promotoras.
Philips (que mantiene la marca), Ember, Freescale, Honeywell, Invensys, Mitsubishi, Motorola, Samsung.
3. Lista de participantes.
 - ❑ Rápido crecimiento (188 miembros hasta ahora)
 - ❑ Líderes de la industria en todo el mundo ha comprometido a proporcionar productos conformes ZigBee y soluciones.
 - ❑ Incluyen los fabricantes de semiconductores (STMicroelectronics), proveedores de IP inalámbricos, fabricantes de equipos originales y usuarios finales.
4. La versión 1.0 de la especificación ha sido ratificado-en diciembre de 2004.
5. Los conductores primarios.

Baja tasa de transmisión; Simplicidad y bajo costo; Batería de larga duración; Seguridad y fiabilidad; Las capacidades de red (estrella, árbol-estrella y topologías de malla) aplicación de perfiles interoperables [5].

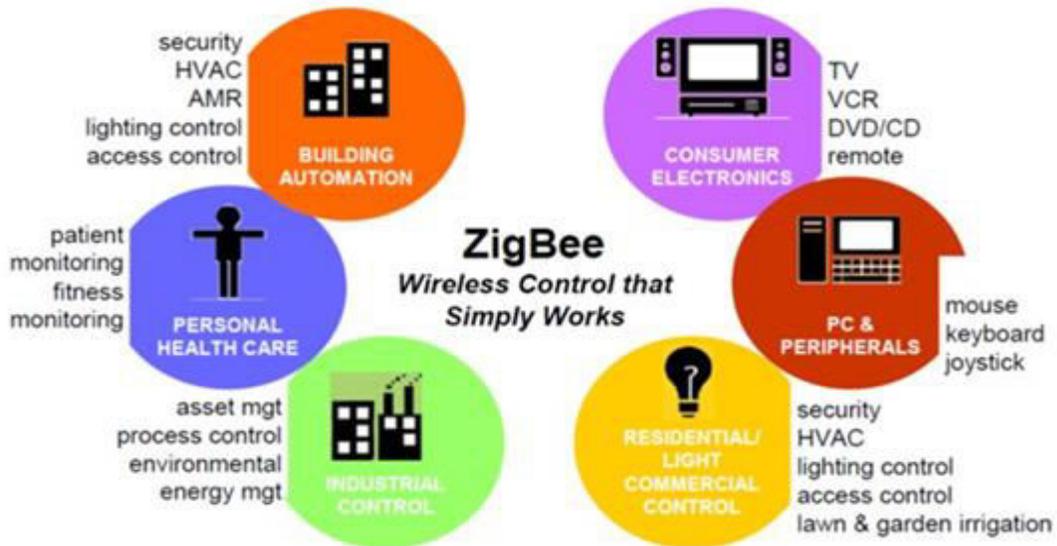


Figura 2.1 Aplicaciones ZigBee ⁷

Crossbow Mica Motes

Redes de sensores inalámbricos han atraído un amplio interés de la industria debido a su diversidad de aplicaciones. Una clave para el alcance de su potencial es la creación de redes de malla multi-salto que permite escalabilidad y fiabilidad. Una red de malla es un nombre genérico para una clase de sistemas integrados en red que comparten varias características que incluyen:

- Multi-Salto (Multi-Hop) la capacidad de enviar mensajes punto a punto una estación base, con lo que permite la extensión de un rango escalable.
- Auto-Configuración: Capaz de formación de la red sin la intervención humana;
- Auto Configurable (Self-Healing) Capaz de agregar y quitar nodos de la red de forma automática sin tener que restablecer la red.



- Enrutamiento dinámico, capaz de determinar de forma adaptativa la ruta basada en una red dinámica, condiciones (por ejemplo, la calidad del enlace, el conteo de saltos, u otra métrica).

Cuando se combina con la administración de energía de la batería, estas características permiten a las redes de sensores sean de larga duración, fáciles de

7 <http://slideplayer.com/slide/5926588/>

implementar, y resistentes a los cambios impredecibles en el canal inalámbrico. Con la creación de redes de malla, la visión de la percepción general y al detalle se hace realidad.

Un despliegue de red inalámbrica se compone de los tres niveles de software distintos:

- El nivel de cliente proporciona el software de visualización de usuario y la interfaz gráfica para gestión de la red. MEMSIC ofrece software cliente gratuito llamado MoteView que agrupa software de los tres niveles para proporcionar una solución de extremo a extremo.
- El servidor de etapa es una instalación siempre activa que se encarga de la traducción y de amortiguación de los datos de la red inalámbrica y proporciona el puente entre las motas inalámbricas y los clientes de Internet Xserve y XOtap son aplicaciones de nivel de servidor que se pueden ejecutar en un PC.
- El Nivel Mote, donde reside XMesh, es el software se ejecuta en la nube de nodos de sensores la formación de una red de malla. El software XMesh proporciona los algoritmos de red requeridos para formar una cadena principal de comunicación fiable que conecta todos los nodos dentro de la nube de malla para el servidor figura 2.2.

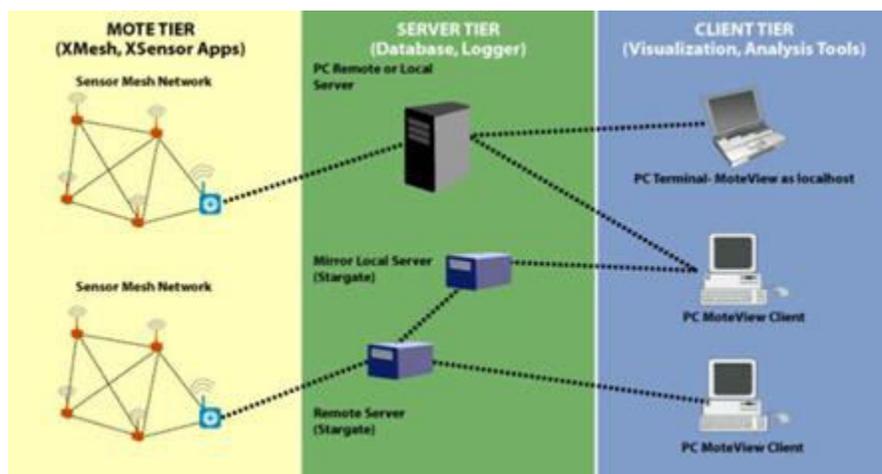




Figura 2.2 Marco de Trabajo para una red inalámbrica de sensores.⁸

⁸ <https://bit.ly/2txeIn8>

Revisión de Literatura Crossbow

Sistema inteligente para entrenamiento deportivo

La retroalimentación directa de la condición del atleta, del medio ambiente y la configuración de la pista. El principal logro es el diseño del método de clasificación, el cual caracteriza una tasa de éxito que es cerca al 80% y se demostró que este éxito depende de manera crítica de los datos ambientales y la retroalimentación positivamente mejora los resultados desde la decisión del equipo. Ambos infraestructura y el equipo de usuario los cuales se desarrollaron con tecnologías del mercado funcionaron bien [13].

Rendimiento en el análisis de la evolución del fuego

Aunque el sistema en un primer ejercicio, no le fue posible seguir la evolución del fuego, los resultados demuestran que las variaciones de temperatura y la humedad permiten con una gran relevancia la presencia de fuego, el tiempo de respuesta es adecuado y más corto a la escala del fuego. En un segundo ejercicio se reforzó el aislamiento de los sensores para evitar fueran destruidos por el fuego, demostrando que el sistema detecta la temperatura y la humedad en el aire con precisión [14].

Sistema de detección de intrusos humanos

La seguridad en el mundo de hoy obliga al monitoreo de áreas de gran valor estratégico, requiere de cordones de seguridad más estrechos que deben ser establecidos en grandes campamentos para protegerse de intrusos, ataques maliciosos o saboteadores, esto las 24 horas los 7 días de la semana, los cuales se presentan con un alto grado furtivo. En adición a esto deben contar con un alto grado de confiabilidad, longevidad y costo, requisitos fundamentales para tal



implementación. Los autores demostraron una confiabilidad del 85% en la tasa de detección humana exitosa, y debe ser desplegado tomando en cuenta los factores ambientales [15].

Monitoreo de cables de tensión en puentes colgantes

Los cables como elementos principales del soporte de la carga de puentes colgantes, pueden proveer valiosa información acerca de la solidez estructural. La tensión puede ser medida utilizando celdas de carga, sensores de estrés electromagnéticos, extensómetros y métodos basados en la vibración son usados en la práctica. La forma tradicional cableada de adquirir datos, es impráctica para un monitoreo de largo plazo. Gracias a las redes (WSN) tenemos gran alternativa para el monitoreo autónomo de largo plazo en los cables de tensión. Los autores comentan que el sistema fue validado en el puente colgante Jindo en Corea del Sur sostenido por cables de tensión, demostrando la sustentabilidad del sistema, su autonomía, la dependencia de la fuerza de tensión con la temperatura, aunque con problemas en la pérdida de datos [16].



Capítulo III: Metodología

El presente trabajo se centra en la exploración de dos plataformas tecnológicas: Wasmote de la compañía Libelium y Motes de la compañía Crossbow. En las siguientes secciones para cada caso en particular se describe cómo se prepara el entorno de trabajo para poder trabajar con estas tecnologías así como las pruebas preliminares desarrolladas para la captura y recolección de los datos previo a la implementación de un sistema de visualización del comportamiento de estas redes de sensores.

3.1 WASPMOTE

Wasmote es una red de sensores inalámbricos de código abierto. Utiliza procesadores AT mega, con diferentes configuraciones de memoria RAM/ROM, usa tecnología de bajo consumo de energía y cuenta con una arquitectura modular que le permite habilitar su canal de comunicación vía Bluetooth, WiFi, 3G, además a su capacidad de comunicación ZigBee⁹. Wasmote es una plataforma diseñada para programadores, es decir se requiere conocer el lenguaje de programación C, así como conocimiento en el área de redes de computadoras. Una descripción más detallada de las características técnicas de la tecnología waspmote se ofrece en el apéndice A Figura 3.0.





Figura 3.0 Módulo Waspote con canal de comunicación Bluetooth y Zigbee.⁹

Para poder hacer uso de los sistemas Waspote es necesario, además de descargar e instalar el IDE para el desarrollo de programas y programación de las tarjetas, definir el modo de operación de la red y configurar adecuadamente el nodo de comunicación Zigbee. ⁹ <https://bit.ly/2ttMb1s>

3.1.1 Topologías de la red Zigbee

La topología se refiere a la configuración de los componentes hardware y como los datos son transmitidos a través de esa configuración. Cada topología es apropiada bajo ciertas circunstancias y puede ser inapropiada en otras. Las topologías que se pueden utilizar en las RSI son: estrella, malla o una híbrida entre ellas dos.

Topología en estrella.- Es un sistema donde la información enviada solo da un salto y donde todos los nodos sensores están en comunicación directa con la puerta de enlace. Todos los nodos sensores son idénticos, nodos finales, y la puerta de enlace capta la información de todos ellos. La puerta de enlace también es usada para transmitir datos al exterior y permitir la monitorización de la red. Los nodos finales no intercambian información entre ellos, sino que usan la puerta de enlace para ello, si es necesario. Esta topología es la que menor gasto de energía desarrolla, pero por el contrario está limitada por la distancia de transmisión vía radio entre cada nodo y la puerta de enlace.

Topología en malla.- En esta configuración cada nodo puede enviar y recibir información de otro nodo y de la puerta de enlace. A diferencia de la topología en estrella, donde los nodos solo pueden hablar con la puerta de enlace, en ésta los nodos pueden enviarse mensajes entre ellos. Esta topología es altamente tolerante a fallos ya que cada nodo tiene diferentes caminos para comunicarse con la puerta de enlace. Si un nodo falla, la red se reconfiguran alrededor del nodo fallido automáticamente.

Topología híbrida estrella-malla.- Busca combinar las ventajas de los otros dos tipos, la simplicidad y el bajo consumo de una topología en estrella, así como la posibilidad de cubrir una gran extensión y de reorganizarse ante fallos de la topología en malla. Este tipo crea una red en estrella alrededor de nodos



intermedios pertenecientes a una red en malla. Los nodos intermedios dan la posibilidad de ampliar la red y de corregir fallos en estos nodos y los nodos finales se conectan con los nodos intermedios cercanos ahorrando energía.

La Figura 3.1 muestra gráficamente las tres configuraciones de red mencionadas anteriormente.

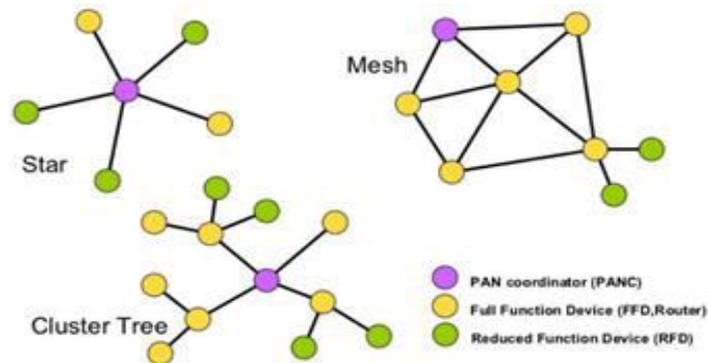


Figura 3.1 Los modelos de topología de red ZigBee ¹⁰

Los módulos ZigBee de la redes Waspnote soportan las topologías antes mencionadas.

Para cualquiera de las topologías para la red se requiere realizar la configuración de los nodos Zigbee ya sea como, Ruteador, Coordinador o Dispositivo final, ya que estos son los puntos a través de los cuales se realiza el intercambio de información dentro de la red.

- Coordinador ZigBee (ZC): Siendo el dispositivo más capaz, el coordinador forma la raíz. Almacena la información acerca de la red, incluyendo actuar como el Centro de Confianza y depósito de claves de seguridad
- ZigBee ruteador (ZR): Junto con el funcionamiento de una función de aplicación, puede actuar como un ZR un enrutador intermedio, la transmisión de datos desde demás dispositivos.
- Dispositivo Final ZigBee: El nodo cliente o nodo extremo (ZED) como también se le suele llamar, contiene sólo lo suficiente para hablar con las funciones en el nodo padre y no puede re-transmitir datos desde los demás dispositivos. Un ZED requiere la menor cantidad de memoria [4].



En nuestro caso, se usaron las configuraciones de fábrica, es decir, existe un solo nodo coordinador y cualquier nodo extremo puede conectarse y transmitir datos hacia el coordinador

¹⁰ <https://bit.ly/2loOaAB>

3.1.2 Configuración de los nodos de la red.

Como mencionamos antes de poner en operación la red de sensores Waspote es necesario configurar los nodos. La configuración de los nodos ZigBee se debe realizar por medio una herramienta conocida como X-CTU. Esta herramienta es proporcionada por DIGI International, fabricante de los dispositivos XBee/ZigBee. Antes de utilizar esta herramienta debe montarse el dispositivo ZigBee en la tarjeta Gateway provista por Waspote para la configuración de los dispositivos como para usarse como el nodo coordinador de la red, figura 3.2.



Figura 3.2 Se muestra la tarjeta Gateway y un dispositivo ZigBee para su configuración. ¹¹

Las figuras muestran los puertos de comunicación disponibles en X-CTU para la configuración de los dispositivos ZigBee figuras 3.3 y 3.4.



11 <https://bit.ly/2N237VL>

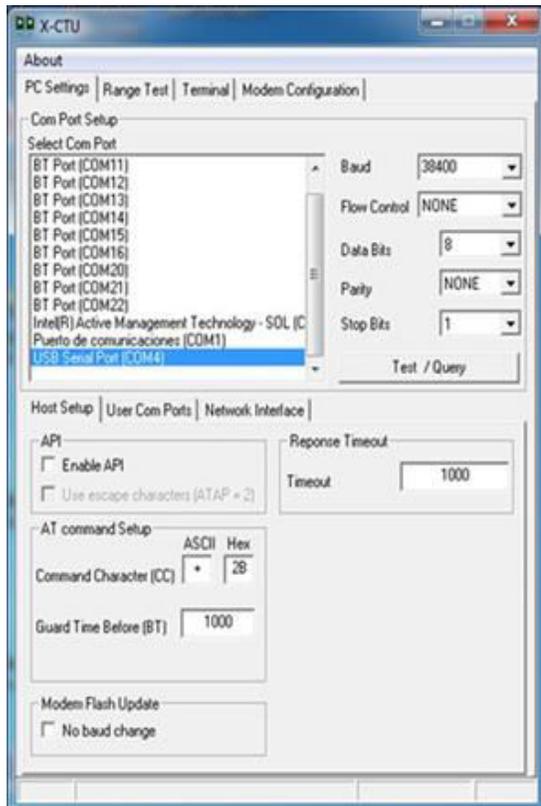


Figura 3.3 CTU configurado para acceder Módulo Xbee



Figura 3.4 Comunicación exitosa con el modem

Cada Xbee tiene un microcontrolador interno que ejecuta un programa, también conocido como firmware, el cual controla las funciones del Xbee como la comunicación y seguridad. Para cambiar el firmware se utiliza el programa X-CTU (Solo disponible para Windows), el cual se puede descargar desde el siguiente link: http://ftp1.digi.com/support/utilities/40003002_C.

Además del X-CTU, se deben instalar los controladores adecuados para Xbee Explorer, la mayoría de estos adaptadores utilizan drivers FTDI. Los controladores y las instrucciones de instalación se encuentran en:



<http://www.ftdichip.com/FTDrivers.htm>.

3.1.3 Puesta en marcha de la red Waspote

Antes que nada se deben anotar los valores SH y SL para cada Xbee (Figura 3.5). El valor superior es llamado SH (Serial Number High) y el inferior SL (Serial number Low).



Figura 3.5 MAC address Módulo Xbee ¹²

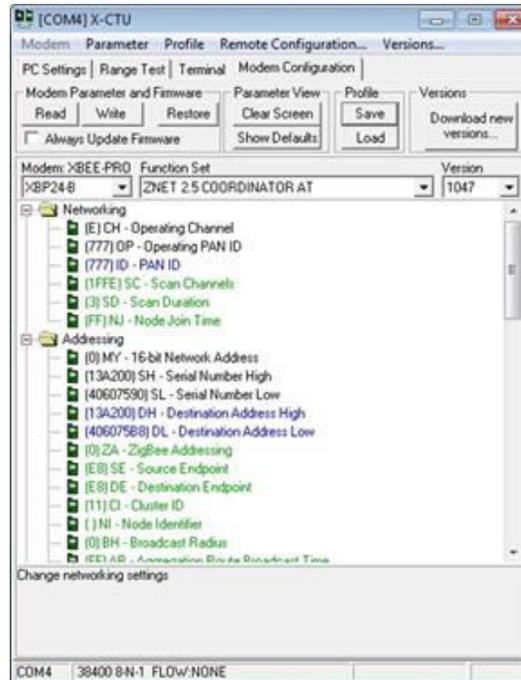


Figura 3.6 Configuración del Coordinador de la Red de Sensores

La configuración original y su modificación es la que se muestra a continuación, Figura 3.6.1

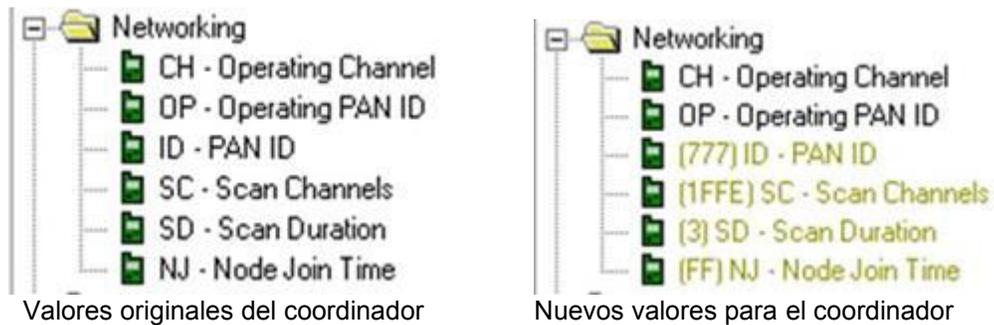


Figura 3.6.1 Modificación de valores de configuración nodo coordinador

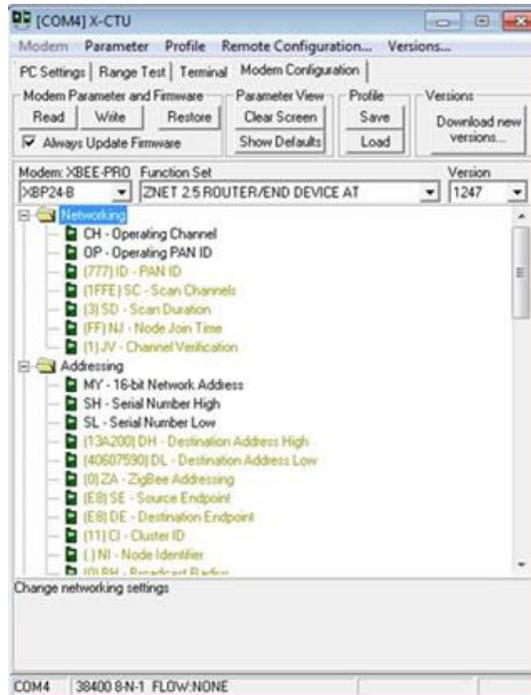


Figura 3.7 Configuración Dispositivo final de la Red de Sensores

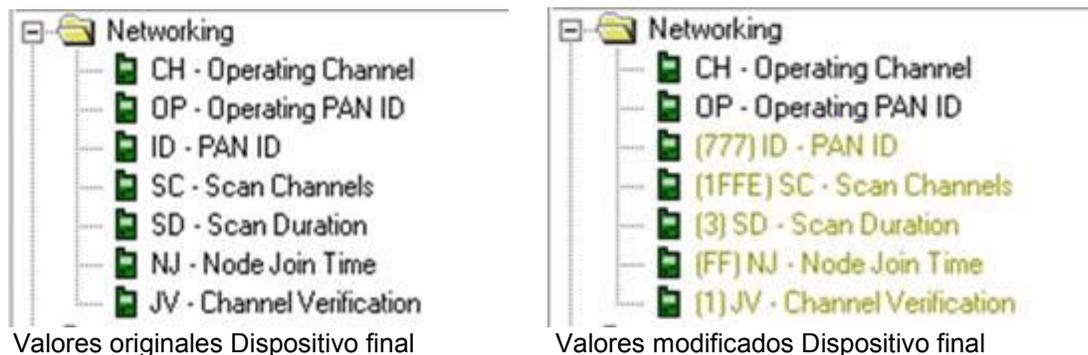


Figura 3.6.1 Modificación de valores de configuración nodo dispositivo final

En las figuras 3.6 y 3.7 anteriores se mostraron las configuraciones básicas para la establecer la comunicación entre el coordinador (gateway) y los nodos de la red, los valores que fueron modificados son el *PAN ID*, que es el mismo para ambos nodos, el canal de comunicación *SC*(scan channels), la duración del escaneo *SD*(scan duration), el *NJ*(node join time) y por último el *JV*(Channel verification) con lo que se establecen los parámetros que permiten la comunicación inalámbrica entre los dispositivos zigBee.



Configuración modo transparente

- Se forma una conexión punto a punto entre dos nodos
- Ambos mandan y reciben información por su puerto serial
- Funciona como un reemplazo de un cable serial físico
- Se reenvía al aire todo lo que se recibe por el puerto serial, tal cual.
 - Lo que el nodo 'A' recibe por su puerto serial, lo envía mediante la conexión inalámbrica al nodo 'B' y los datos de este último están disponibles por el puerto serial.

Configuración Modo de comando AT

- La información recibida por el puerto serial no es transmitida, sino interpretada por el módulo local.
- En este caso lo que el módulo recibe son comandos y no datos
- Es un modo de configuración/control del módulo.

Configuración Modo API

- La información se manda y se recibe en forma de paquetes de datos
- Los paquetes pueden contener datos o comandos AT
- Los paquetes de datos se transmiten a los módulos remotos
- Los paquetes de comandos AT pueden ser:
 - Locales
 - Remotos (destinados a otros módulos en la red). Esto permite reprogramar/reconfigurar módulos remotamente.

3.1.4 Programación de Módulos

En esta sección se aplican los conceptos revisados en las secciones anteriores para poner en ejecución la red Waspote y explorar los datos recibidos por el Gateway. Primeramente, se requiere tener instalado el IDE Waspote el cual permite cargar los programas a las tarjetas: <http://www.libelium.com/development/waspote>.

Instalación de los módulos en las tarjetas

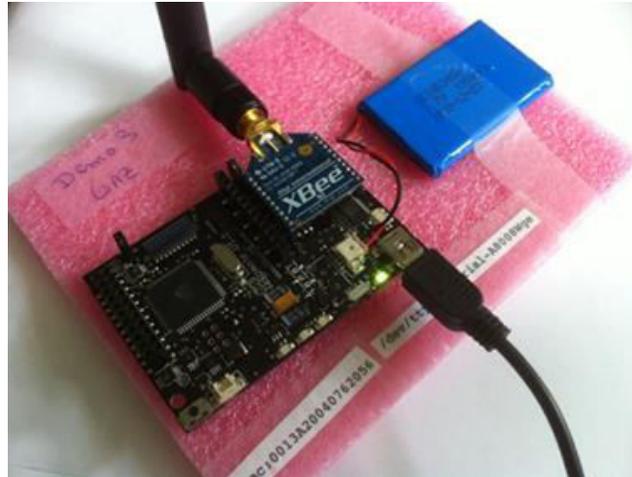


Figura 3.8 Ensamble de XBee en la tarjeta Waspmote

Una vez que se han configurado los dispositivos ZigBee por medio de la herramienta X-CTU se procede a colocar los módulos en las tarjetas waspmote, las cuales previamente y antes de colocar los módulos ZigBee fueron cargadas con el programa adecuado a las necesidades del entorno que se desea analizar. La figura 3.9 Muestra el programa abierto en el IDE para su descarga posterior a la tarjeta waspmote.

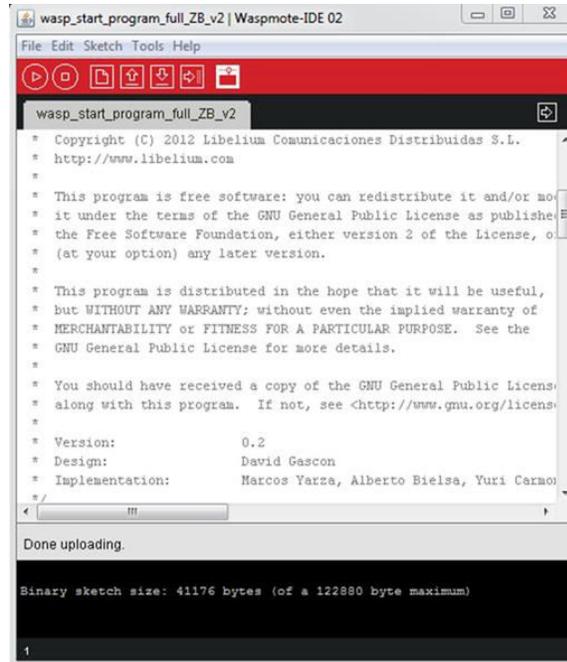


Figura 3.9 Imagen del IDE con programa abierto

Breve descripción del IDE

El ambiente está formado por un Menú de opciones como son

`File Edit Sketch Tools Help`

los cuales permiten la carga, edición y ejecución de los programas que interpretan las señales generadas por los diferentes módulos

insertados en el waspmote. La barra de herramientas  tiene botones que nos permiten ahorrar tiempo con acceso rápido para abrir y descargar códigos de programación. La pestaña que nos muestra el nombre del código que

se utiliza `wasp_start_program_full_ZB_v2 $` y que es mostrado bajo de esta misma

pestaña, como un editor de textos, el cual permite crear o modificar estas unidades llamadas “sketches” los cuales son guardados con la extensión “pde”. El área de mensajes brinda comentarios mientras se guarda o exporta el código, y también muestra errores si los hay al momento de la compilación del código

`Binary sketch size: 41176 bytes (of a 122880 byte maximum)`

Este IDE le permite administrar más de un archivo al mismo tiempo (en la zona de pestañas). Estos pueden ser



archivos de código Wasmote normales (.pde), archivos C (extensión .c), archivos C ++ (.cpp) o archivos de encabezado (.h).

3.1.5 Revisión lógica del código

Una vez que se ha revisado la lógica de programación del código y se satisfacen los parámetros, se procede a la descarga a la tarjeta waspmote, para esto es necesario seleccionar correctamente el puerto serial de comunicación el cual en Windows se denomina como COMX:. Una vez que se elige el puerto, se descarga el programa a la tarjeta y se realizan pruebas para confirmar el correcto funcionamiento del código.

```
10 void loop(){
11
12  sprintf(aux,"-mac:%s%s -x:%d,y:%d,z:%d -temp:%d -bat: %d%c%c%c",macHigh,
    macLow,ACC.getX(),ACC.getY(),ACC.getZ(),(int)RTC.getTemperature(),
    PWR.getBatteryLevel(),'%','\r','\n');
13
14  paq_sent=(packetXBee*) calloc(1,sizeof(packetXBee));
15  paq_sent->mode=UNICAST;
16  paq_sent->MY_known=0;
17  paq_sent->packetID=0x52;
18  paq_sent->opt=0;
19  xbeeZB.hops=0;
20  xbeeZB.setOriginParams(paq_sent,MAC_TYPE);
21  xbeeZB.setDestinationParams(paq_sent, direccion, aux, MAC_TYPE, DATA_ABSOLUTE);
22  xbeeZB.sendXBee(paq_sent);
23
24  free(paq_sent);
25  paq_sent = NULL;
26
27  delay(1000);
28 }
```

Figura 3.9.0 Código cargado en el IDE para su ejecución en Wasmote

A continuación se hace una descripción del código figura 3.9.0.

12 función que da formateo de impresión de los parámetros: Dirección MAC, valores xyz del acelerómetro, temperatura, el nivel de energía de la



batería, funciones que toman la dirección alta y baja de, la función `getX`, toma los valores del acelerómetro del waspmote, función que toma la temperatura, y la última función `PWR.getBatteryLevel`, que obtiene el valor del nivel de energía de la batería conectada al waspmote.

14 asigna dirección de memoria al paquete

15 asigna al paquete el modo de transmisión tipo UNICAST

16 asignación de valor cero a variable `MY_Known`

17 asignación del valor `0x52` al identificador del paquete

18 valor de cero para la variable `opt` del objeto `paq_sent`

19 asigna valor cero al parámetro `hops`

20 función del origen de la transmisión del paquete y su tipo de dirección MAC

21 la función que asigna los parámetros, paquete, dirección de destino, formato de datos

22 la función con el parámetro `paq_sent` envía el paquete previamente asignado con valor

24 Libera la variable de memoria `paq_sent`

25 Asigna el valor nulo a la variable `paq_sent`

27 Espera un segundo para continuar con el ciclo

3.1.5 Exploración de datos

Una vez realizadas las configuraciones de los módulos ZigBee y cargado el programa en la tarjeta Waspmote, se procede a verificar los datos que genera los diferentes sensores de la misma, produciendo los resultados que a continuación se presentan en la siguiente figura 3.10.



Figura 3.10 Datos recibidos de la tarjeta waspmote por medio de módulo zigBee

Se observa que la consola muestra los valores recibidos de los sensores iniciando con la dirección MAC de la tarjeta ZigBee, las posiciones x, y de posición, la temperatura y el porcentaje que resta a la energía de la batería. Si revisamos la figura 3.9.0 que muestra el código del programa cumple el orden de los parámetros que captura dicho código de la tarjeta waspmote. Debemos recordar que la transmisión se realizó de manera inalámbrica gracias a los módulos ZigBee. La dirección MAC nos permite identificar de cuál nodo de la red de sensores estamos capturando sus datos. De la investigación realizada hasta este momento se concluyó lo siguiente:

Se ha logrado desarrollar un sistema sencillo de comunicación inalámbrico para la medición de los sensores. Aunque se ha hecho uso de un solo de un par de nodos



para la captura de los datos, se han adquirido los conocimientos necesarios para la creación de una red de sensores, y este procedimiento se podría ampliar para una red más compleja.

Respecto al procesado de los datos, ha sido de gran utilidad el uso de los componentes integrados, la tarea se ha simplificado notablemente, el proporcionar ejemplos de código que para un principiante resultaría complejo, desarrollar la misma aplicación con las mismas características, desarrollando él mismo, el código completo.

Analizando los datos recabados, se confirma el correcto funcionamiento de los dispositivos así como del código implementado. Sin embargo, quedan varias líneas abiertas por investigar.

Desafortunadamente se presentaron problemas técnicos con los dispositivos, y no fue posible continuar con los objetivos planteados para con los dispositivos Waspote.

Introducción al Ruteador Meshlium

Al funcionar como una puerta de enlace de la red de sensores inalámbricos Libelium Waspote, Libelium Meshlium es un enrutador Linux que contiene 5 interfaces de radio diferentes, WIFI 2.4GHz, WIFI 5GHz, 3G / GPRS, Bluetooth y ZigBee. El Meshlium puede ser alimentado por panel solar, batería o ambos y puede integrarse con un módulo GPS para monitorear aplicaciones de vehículos y aplicaciones móviles. Debido a la armazón de aluminio IP-65, el Meshlium puede colocarse en el exterior.

Libelium proporciona un sistema de gestión que funciona como una aplicación web para que los desarrolladores controlen Meshlium de forma rápida y sencilla. El Administrador de sistema proporciona configuraciones gráficas del Módulo WIFI, ZigBee, Bluetooth y 3G / GPRS y opciones de almacenamiento del sensor y los datos recibidos. El Meshlium equipado con dos antenas WIFI y dos antenas ZigBee se muestra en la Figura 3.11

Como coordinador que recibe los parámetros ambientales recopilados, Meshlium se utiliza en comunicación con Waspote equipado con el módulo XBee ZigBee



Pro. En el Sistema de gestión, Meshlium puede configurarse para guardar los datos recopilados en archivos locales, bases locales o incluso bases de datos externas con una dirección específica y un puerto en Internet. El Administrador del sistema es fácil de usar, y nos permite verificar los datos almacenados en él, provenientes de los sensores de la red.



Figura 3.11 Meshlium ¹⁴

A continuación una descripción breve de las características del dispositivo Meshlium figura 3.12. Como se apuntó antes el manejo del administrador del sistema es de muy fácil uso, gracias a su interface tipo página de internet, la cual permite la configuración de los diferentes parámetros de conexión. Para acceder es necesario abrir el explorador y dar una IP en el rango de 10.10.10.10 - 10.10.10.250 de la manera como se muestra a continuación.



- ❑ URL: <http://10.10.10.1/ManagerSystem>
- ❑ user: admin
- ❑ password: libelium

Figura 3.12 Dirección IP para acesar el System Manager del Meshlium ¹⁵

^{14,15} <https://bit.ly/2Mf7Pyt>



Figura 3.13 Pantalla de acceso al sistema de administración Meshlium ¹⁶

Si su red no ofrece servicio DHCP, Meshlium comienza con una dirección IP predeterminada (192.168.1.100). En este caso, puede conectar Meshlium a través de la conexión WiFi (que siempre está disponible) o con el cable cruzado provisto con Meshlium.

Si desea acceder al sistema administrador utilizando el cable Ethernet cruzado, vaya a:

- ❑ URL: <http://192.168.1.100/ManagerSystem>
- ❑ user: admin
- ❑ password: libelium

Si lo hace, lo llevará al Sistema Administrador Meshlium, luego haga clic en el botón "Sensor Networks" figura 3.14.

¹⁶ <https://bit.ly/2Mf7Pyt>

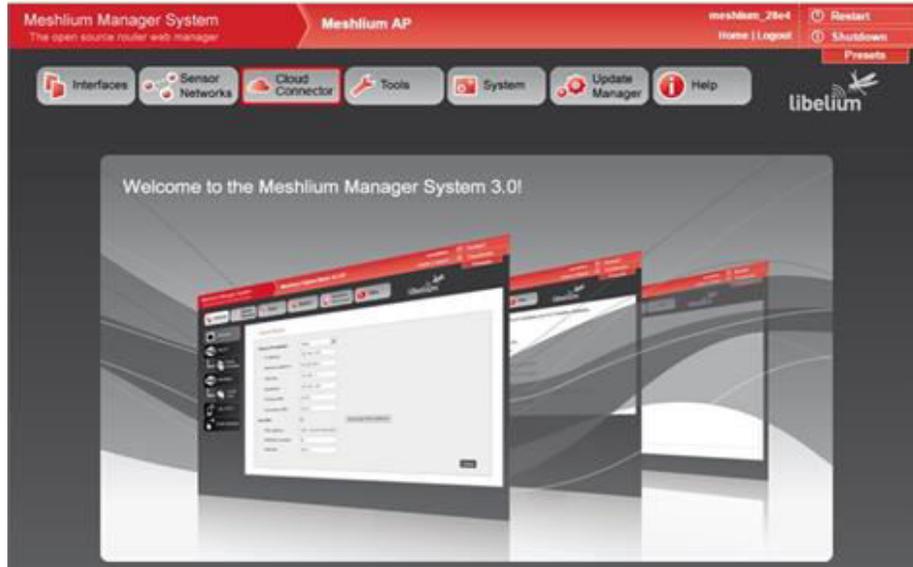
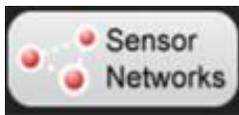


Figura 3.14 Pantalla principal del Administrador del Sistema Meshlium ¹⁷

Acceda a la configuración de la red de sensores haciendo clic en el botón "Sensor Networks":



Una de las aplicaciones de Meshlium es actuar como punto de entrada (GW) de las Redes de Sensores Waspote. Waspote es un dispositivo sensorial especialmente orientado a desarrolladores. Funciona con diferentes protocolos (ZigBee, Bluetooth, GPRS) y frecuencias (2,4GHz, 868MHz, 900MHz) siendo capaz de obtener enlaces de hasta 12km. Cuenta con un modo de hibernación de 0.7uA que permite ahorrar batería cuando no está transmitiendo.

¹⁷ <https://bit.ly/2Mf7Pyt>



Figura 3.15 Pantalla de configuración red de sensores. ¹⁸

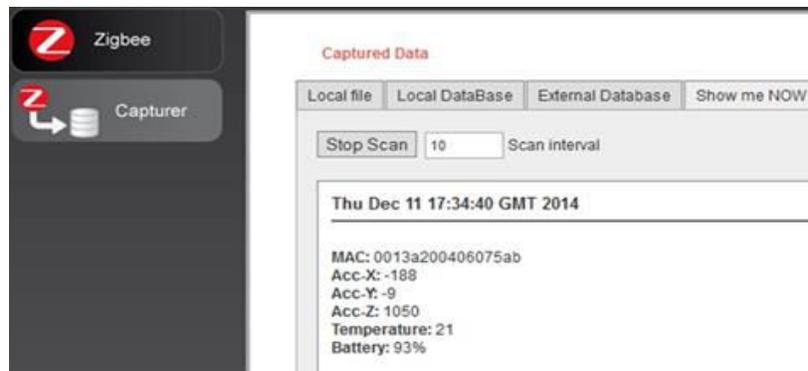


Figura 3.16 Pantalla de captura de datos nodo 75ab ¹⁹

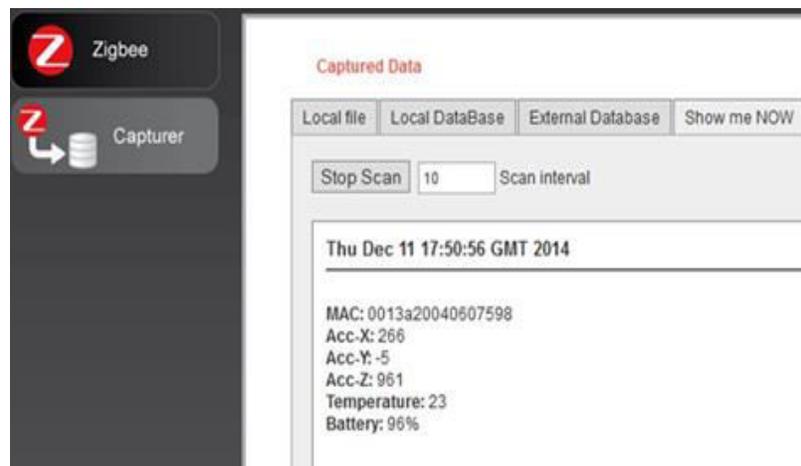




Figura 3.17 Pantalla de captura de datos nodo 7598 ²⁰

18,19,20 <https://bit.ly/2Mf7Pyt>

En las dos figuras 3.17 y 3.16 anteriores podemos ver los datos recibidos por el nodo, los cuales son consistentes con los mostrados en la *exploración de datos*, que muestran a su vez los parámetros descritos en la parte de *presentación lógica y descripción* de datos en las sección de los waspmote, la cual muestra Las tramas enviadas por Waspote contienen la siguiente secuencia:[Cabecera]-mac:0013a20040607598 -x:266,y:-5,z:961 -temp:23 -bat:96%. Para complementar la consistencia de los datos, enviados por las tarjetas Waspote por medio de los dispositivos ZigBee, se muestra a continuación la Figura 3.18 en la que se pueden comprobar las tramas recibidas de los diferentes nodos conectados a la red y que son capturadas, por medio de la consola de comandos del Meshlium [17].

```
meshlium:/etc/init.d# capturer S0 38400
setting speed 38400
ttyS0 chosen
...
Press Ctrl+C to close the program
+++ok
r#0 u-mac:0013a200406075ab -x:-215,y:16,z:1044 -temp:20 -bat: 93%
r#0 u-mac:0013a200406075ab -x:-228,y:-22,z:1042 -temp:20 -bat: 86%
r#0 u-mac:0013a200406075ab -x:-227,y:-23,z:1042 -temp:20 -bat: 92%
r#0 u-mac:0013a200406075ab -x:-229,y:-26,z:1041 -temp:20 -bat: 92%
r#0 u-mac:0013a200406075ab -x:-229,y:-26,z:1042 -temp:20 -bat: 93%
r#0 u-mac:0013a200406075ab -x:-228,y:-27,z:1041 -temp:20 -bat: 92%
r#0 u-mac:0013a200406075ab -x:-146,y:15,z:1056 -temp:20 -bat: 92%
r#0 u-mac:0013a20040607598 -x:-24,y:46,z:997 -temp:25 -bat: 96%
r#0 u-mac:0013a20040607598 -x:-24,y:45,z:997 -temp:24 -bat: 96%
r#0 u-mac:0013a20040607598 -x:-21,y:43,z:998 -temp:24 -bat: 96%
r#0 u-mac:0013a20040607598 -x:763,y:-158,z:609 -temp:24 -bat: 95%
r#0 u-mac:0013a200406075ab -x:456,y:447,z:829 -temp:20 -bat: 88%
r#0 u-mac:0013a200406075ab -x:551,y:312,z:-524 -temp:23 -bat: 92%
r#0 u-mac:0013a20040607598 -x:-326,y:377,z:825 -temp:27 -bat: 96%
r#0 u-mac:0013a200406075ab -x:720,y:633,z:181 -temp:23 -bat: 70%
r#0 u-mac:0013a200406075ab -x:796,y:309,z:-562 -temp:25 -bat: 92%
r#0 u-mac:0013a20040607598 -x:145,y:1000,z:163 -temp:28 -bat: 97%
r#0 u-mac:0013a200406075ab -x:270,y:854,z:-457 -temp:25 -bat: 93%
r#0 u-mac:0013a200406075ab -x:382,y:752,z:-560 -temp:27 -bat: 93%
r#0 u-mac:0013a200406075ab -x:374,y:767,z:-544 -temp:27 -bat: 47%
r#0 u-mac:0013a200406075ab -x:491,y:697,z:-560 -temp:27 -bat: 93%
r#0 u-mac:0013a200406075ab -x:119,y:-73,z:1046 -temp:28 -bat: 92%
```

Figura 3.18 Captura de pantalla de la consola meshlium capturando datos varios nodos

Para este momento en el que fue posible la captura los datos provenientes de los nodos figura 3.18, con la configuración lograda, de los ZigBee como de los Waspote, y que los problemas técnicos impidieron configurar más sensores, se decidió recopilar la cantidad más amplia posible con los dos nodos disponibles y



así generar los datos estadísticos necesarios, por cuestiones técnicas la investigación de los Wasmote se dio por concluida en este punto.

3.2 MOTES

3.2.1 Introducción a MoteView

MoteView está diseñado para ser una interfaz ("capa de cliente") figura 3.19, entre un usuario y una red desplegada de sensores inalámbricos. MoteView proporciona las herramientas para simplificar la implementación y la supervisión. También hace que sea fácil conectarse a una base de datos, analizar y graficar las lecturas de los sensores.

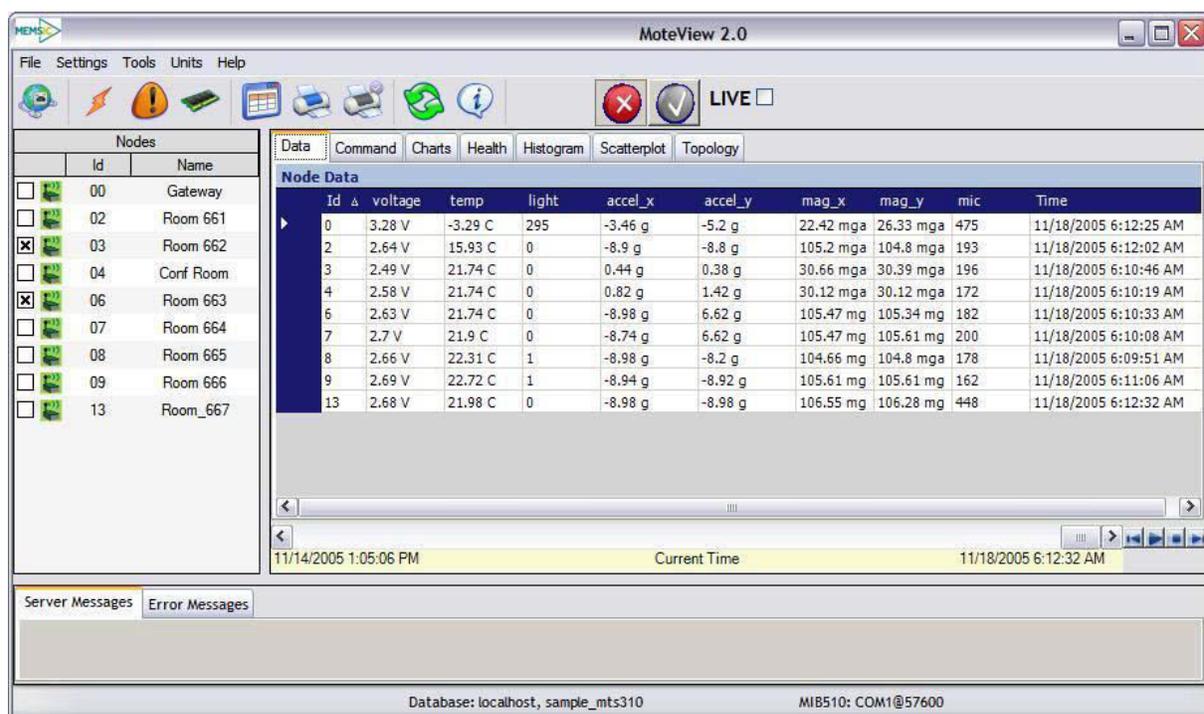


Figura 3.19 Pantalla principal de la aplicación MoteView.

En general lo que presenta la interfaz, son las opciones de Menús y la barra de herramientas, la primera nos permite acceder por medio de la navegación de los niveles de menús a las diferentes funciones que la aplicación proporciona, de igual manera pero con la facilidad de un clic del ratón acceder a la función propia de cada botón. También vamos a encontrar una lista de los nodos que están activos y de los cuales se está recibiendo datos. Por otra parte tenemos el panel de



pestañas de visualización, en la parte de abajo tenemos dos pestañas, una muestra los mensajes del servidor y la otra los mensajes de error de la aplicación. Aunque estos están en la barra de herramientas quise mencionarlos al final por su importancia dentro de esta aplicación son los botones de   LIVE  y la caja de opción live, son los que permiten a la aplicación conectar o desconectar el registro de datos en tiempo real, si el botón de  está encendido en color verde y la caja de opción Live está seleccionada es que se están recibiendo datos en tiempo real en la aplicación, por el contrario si el botón esta en rojo  quiere decir que no se está recibiendo datos en tiempo real.



Figura 3.20 Botón Acceso a Pantalla de Conexión de Sensores

Conectarse a la Red de Sensores, por medio de este botón llegamos a las pantallas de configuración que se muestran a continuación.

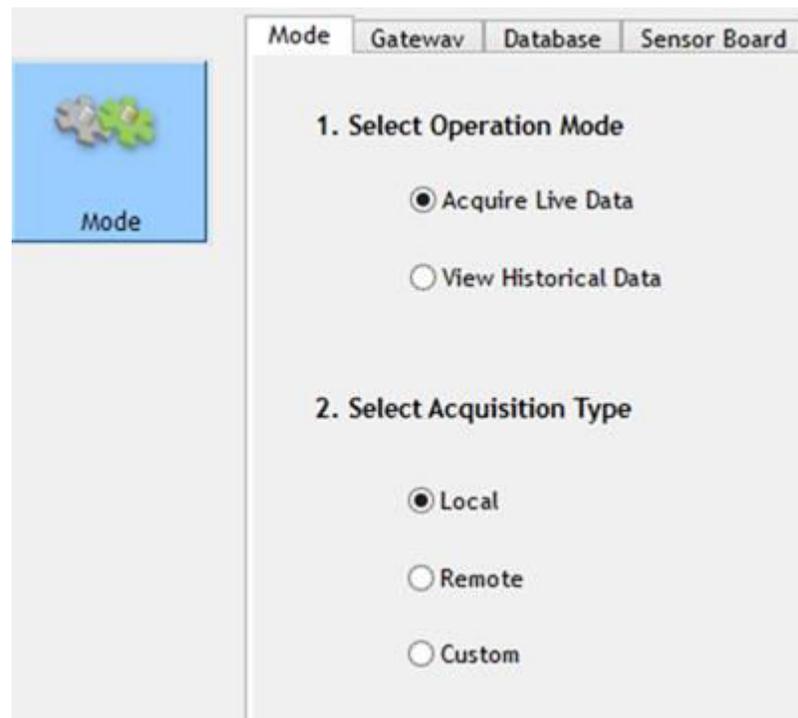




Figura 3.21 Selector del modo de operación y adquisición de datos

Ventana para seleccionar el modo de adquirir datos y la selección del origen de los mismos.

Por medio de un botón de selección excluyente que elige una u otra opción, las dos posibles en el modo de operación son, adquirir datos en línea y la otra el histórico de datos. Los otros dos botones de selección excluyente, permiten elegir entre forma de adquirir los datos, local, remoto, y personalizado por el usuario.

Mode:(Modo)

1.- *Modo de Operación:*

- Adquirir datos En Tiempo Real
- Consulta Datos históricos

2.- *Medio para Adquirir los Datos:*

- Servidor Local
- Servidor Remoto
- Personalizado por el usuario

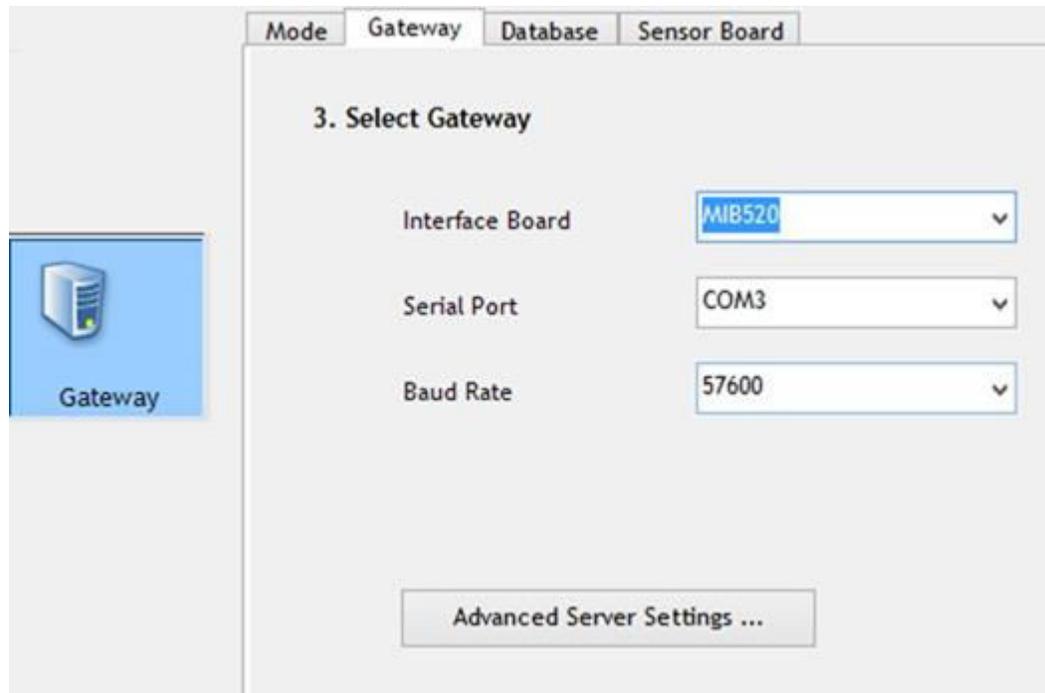


Figura 3.22 Selector de la puerta de enlace y sus parámetros

Ventana para la selección de la puerta de enlace, tipo de tarjeta, puerto de comunicación serial donde se hará la comunicación con la computadora, y por último la tasa de transferencia de los datos.

Gateway:(Puerta de enlace)

- | | |
|----------------------------|--|
| - La tarjeta de interface | <i>MIB520 Modelo de la tarjeta</i> |
| - El puerto serial | <i>COM3 Número de Puerto de comunicación</i> |
| - Velocidad de transmisión | <i>57600 Tasa de Transmisión</i> |

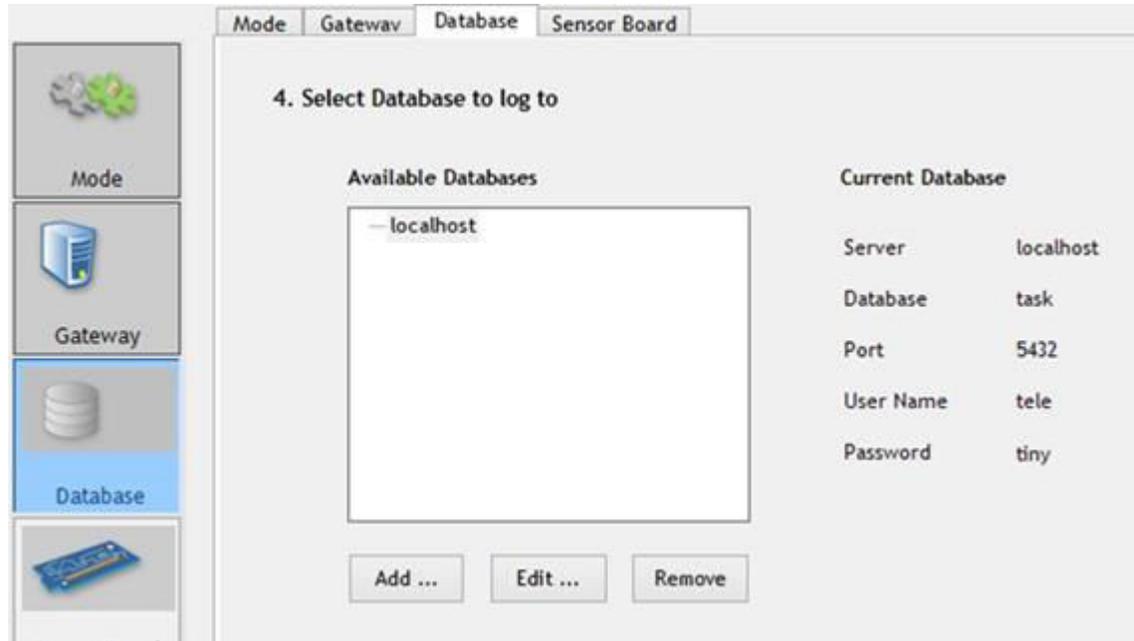


Figura 3.23 Selección de la base de datos.

Esta ventana nos muestra las base de datos actual, quiere decir la que está configurada por defecto desde la instalación, pero permite hacer cambios con los botones de agregar (Add), modificar (Edit) y por último eliminar la conexión a una base de datos existente.

Database: Selección de la base de datos

- Bases de datos disponibles *localhost (Actual)*
- Base de datos actual *task*
- Puerto *5432*
- Nombre del Usuario *tele*
- Password *tiny*

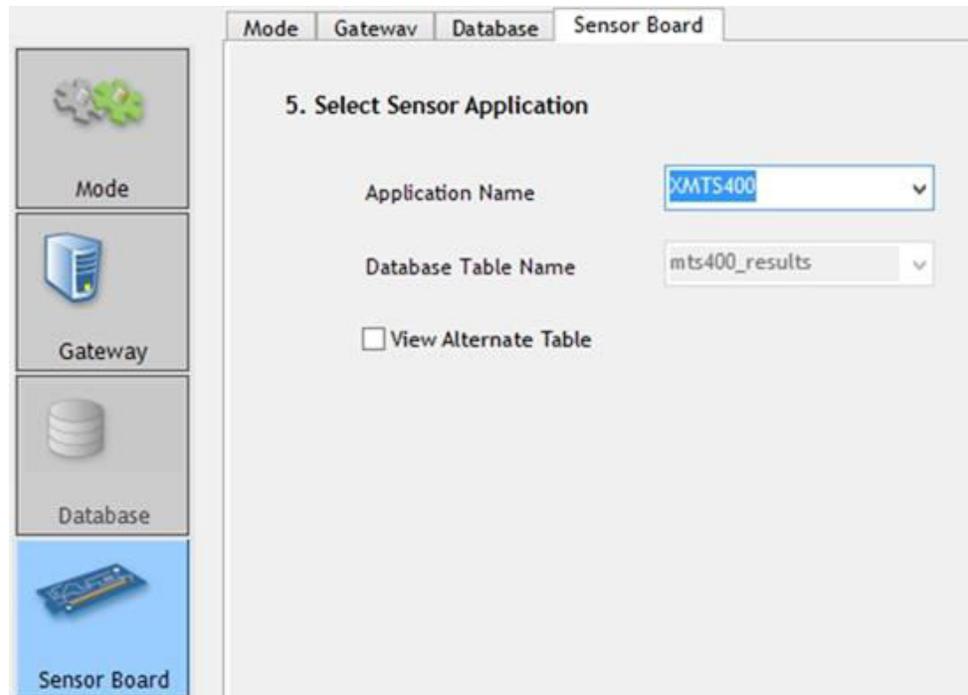


Figura 3.24 Selección de la aplicación y la tabla que almacenará los datos

En esta ventana se selecciona en el nombre de la aplicación, que interpretara los datos provenientes de la tarjeta que interpreta los datos de los sensores y la tabla donde se almacenarán los datos.

Sensord Board: Selección del tipo de tarjeta de sensor

- Nombre de la aplicación *XMTS400*
- Nombre de la Tabla en la Base *mts400_results*

3.2.2 Monitoreo de datos con MoteView

Una vez que se han confirmado las configuraciones por defecto o que se hayan realizado cambios, es posible realizar una consulta al estado que guarda la red de sensores, por medio del botón, *Done* de la sección de configuración, analizadas en las figuras 3.21, 3.22, 3.23 y 3.24, o también presionando el botón LIVE localizado en la barra de herramientas de la figura 3.19. Ambas opciones no



mostrarán en el área de mensajes, que se encuentra en la parte baja de la ventana principal de la aplicación MoteView, Fig. 3.19, nos debe mostrar los siguiente resultado en dicha área de mensajes, este resultado lo mostramos en la figura 3.24.1.

```
Server Messages Error Messages
Starting xserve:
xserve -s=COM10 -b=57600 -l -xmlfile=XmlStream.xml -xmpl -xmlport=9005
XSERVE 2.0.E: $Id: xserve.c,v 1.8.2.3 2007/02/02 17:45:01 rkapur Exp $
Warning: Converting Windows COM10 device to Cygwin device.
Using params: [db parsed] [xml parsed] [server port=9001]
Opening serial device: /dev/ttyS9 @ 57600
[2015/05/27 20:00:10] Serial Source Msg: sync
[MV]: Heartbeat received
```

Figura 3.24.1 Mensaje del servidor al hacer conexión en tiempo real

Una vez que este mensaje es enviado a la ventana de mensajes del servidor, inicia una secuencia de valores como los que presentamos en la siguiente figura 3.24.2.

```
Server Messages Error Messages
Using params: [db parsed] [xml parsed] [server port=9001]
Opening serial device: /dev/ttyS9 @ 57600
[2015/05/27 20:00:10] Serial Source Msg: sync
[MV]: Heartbeat received
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,priemp,press,taosch0,taosch1,accel_x,accel_y) values (now
0,7814,0,411,386,6736,27291,16407,65509,65454,449,449)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,priemp,press,taosch0,taosch1,accel_x,accel_y) values (now
0,7814,0,411,392,6738,27294,16408,65509,65454,450,451)
calibration info updated.
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,priemp,press,taosch0,taosch1,accel_x,accel_y) values (now
0,7814,0,411,394,6739,27298,16408,65509,65454,449,450)
calibration info updated.
Query: INSERT into node_health
(result_time,nodeid,health_pkts,node_pkts,forwarded,dropped,retries,battery,power_sum,board_id,parent,quality_tx,quality_rx,path_cost,parent_rssi) values (now
0,7814,1,7,0,3,0,30,0,133,0,15,15,0,61)
Query: INSERT into mts400_results (result_time,nodeid,parent,voltage,humid,humtemp,priemp,press,taosch0,taosch1,accel_x,accel_y) values (now
0,7814,0,411,390,6739,27300,16407,65509,65453,450,449)
calibration info updated.
```

Figura 3.24.2 Mensajes del servidor actualizando la base de datos mts400_results.

Por supuesto que llamó la atención, reconocer que los mensajes que enviaban en esta ventana, correspondían a una instrucción de inserción tipo SQL y que se puede observar, que los valores son almacenados en una tabla llamada *mts400_results*, exactamente como la definida en la ventana de configuración de



la figura, 3.24. Donde ciertamente fue posible corroborar que se asigna la tabla *mts4000_reusutls*.

3.2.3 Xserve como motor de MoteView



Figura 3.25 Instalación de la Base de datos, PostgreSQL y su ODBC para windows

Se realiza la instalación de las opciones seleccionadas en la Figura 3.25, las cuales son PostgreSQL 8.4, que es la versión de la base de datos que se instalará, además de el manejador tipo ODBC (open database connectivity), el cual manipula las operaciones entre la base de datos y la aplicación, la opción de limpiar la base de datos, la última opción es la de la instalación de la aplicación MoteView versión 2.1.

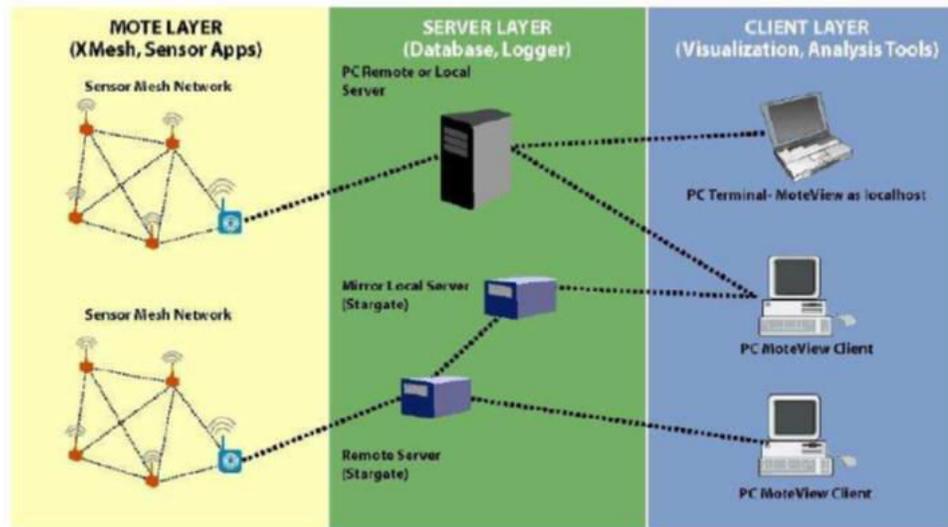


Figura 3.26 Capas del marco de trabajo de Red de Sensores Mote ²⁰

La capa del cliente se mostró en la parte *Descripción general del MoteView*. Ahora se describe las características de las otras dos capas.

²⁰ <https://bit.ly/2yJVj8c>

Descripción general de la capa Motes

La capa de motes está fundamentada en las capacidades del TinyOS, que es el software que ejecutan en la nube los sensores para formar una red de tipo malla. El TinyOS proporciona los algoritmos de red requeridos para formar una cadena de comunicación confiable que conecta todos los nodos dentro de la malla. Como se puede observar el sistema operativo le da vida a la nube como se aprecia en la figura 3.26, que además nos muestra la capa de servidor, que complementa a las otras dos, las tres conforman el marco de trabajo para desplegar una red de sensores tipo mote, cada uno representa la antesala del siguiente, de esta manera si no funciona correctamente cada nivel no podemos obtener los resultados deseados en el funcionamiento de esta red de sensores.

Como ya se dijo antes, la capa del Mote o la red tipo Mesh de sensores, los motes están programados con el sistema operativo llamado TinyOS, llamado firmware (“Aplicación”) para realizar tareas específicas, como monitoreo de microclima, seguimiento de activos, detección de intrusos etc.



La segunda capa o capa de servidor proporciona el registro de datos y servicios de base de datos. En este sensor de capa, las lecturas llegan a la estación base (por ejemplo, MIB510 o concentrador) y se almacenan en un servidor.

Descripción general de la capa Servidor

Durante el proceso de instalación del software, se configura el manejador de base de datos PostgreSQL, el cual se encarga de almacenar los datos que generada por los motes para más detalle ver anexo B.

3.2.4 Uso independiente de Xserve

El software del servidor XServe conecta la red de malla inalámbrica con las aplicaciones del cliente, incluido MOTE-VIEW de Crossbow, así como con los clientes de terceros. Como ejemplo, XServe puede unir una red Mote a un paquete de software empresarial ERP. XServe es compatible con múltiples plataformas y se ejecuta tanto en servidores de

PC con Windows estándar como en la puerta de enlace integrada independiente de Crossbow, Stargate. XServe registra y envía datos, genera alertas y crea informes de diagnóstico automáticamente. XServe contiene un servidor web incorporado para configuración / informes para más consultar el Anexo B.

La parte que genero interés del xserve es la de aplicación que corre por separado sin afectar la operación de otra aplicación y que de hecho es ejecutada dentro de esta misma que la contiene, que es la que nos permitirá tener acceso a los datos de los motes con independencia de la aplicación gráfica del MoteView.

Starting xserve:

```
xserve -s=COM10 -b=57600 -l -xmlfile=XmlStream.xml -xmlp -xmlport=9005
```

```
XSERVE 2.0 E: $Id:xserve.c,v 1.8.2.3 2007/02/02 17:45:01  
Warning: Converting Windows COM10 device to Cygwin device.  
Using params:[db parsed] [xml parsed] [server port =9001]  
Opening serial device:/dev/ttyS9 @ 57600  
[2015/05/27 20:00:10] Serial Source Msg:sync  
[MV]: Heartbeat received
```

Figura 3.26.1 Conexión exitosa al xserve y a la base de datos por medio del puerto 9005



La figura 3.26.1 muestra el mensaje de la iniciación del software Xserve, el cual permite abrir un canal de comunicación en este caso:

xserve	ejecutable de la aplicación
-s=COM10	puerto serial
-b=575600 bits	velocidad de transferencia
-l	sin nombre de tabla específico
-xmlfile=XmlStream.xml	nombre de archivo, el nombre donde se guardan los valores xml
-xmlp	parámetro para parsear los paquetes exportados a xml
-xmlport=9005	puerto de inicio del servidor xml
XSERVE 2.0 E: \$Id:xserve.c,v 1.8.2.3 2007/02/02 17:45:01	Versión del software y año de liberación
Warning:Converting Windows COM10 device to Cygwin device	Mensaje de advertencia
Using params:[db parsed] [xml parsed] [server port =9001]	Usando los parámetros [db, xml, server port]
Opening serial device:/dev/ttyS9 @ 57600	Abriendo el dispositivo serial S9 a velocidad de 57600
[2015/05/27 20:00:10] Serial Source Msg:sync	Fecha y hora de sincronía de los mensajes del serial fuente

La descripción más amplia de los parámetros del XServe se localiza en el Anexo B.

Con esta observación surgió la interrogante de si sería posible reproducir ese proceso fuera de la interfaz gráfica de MoteView.

Entonces se identificó el objetivo: Localizar el archivo xserve correrlo con los mismos parámetros mostrados en la ventana de mensajes del servidor de la interfaz gráfica del MoteView.

Al término de la instalación se observó que se creó dentro de archivos de programas (x86) la carpeta Memsic, que a su vez contiene a la carpeta Moteview, la cual dicho de paso tiene la aplicación MoteView y todos los archivos necesarios para ejecutar el administrador de la base de datos PostgreSQL, además de entre otras la carpeta xserve2.

La carpeta xserve2 contiene una estructura de carpetas que conforma el funcionamiento del xserve, de ahí localizamos el folder bin que es donde



regularmente en las aplicaciones se encuentran los ejecutables, y en efecto contiene el ejecutable `xserve.exe` y todas las librerías necesarias para su funcionamiento seguimos la ruta: `C:\Program Files (x86)\Memsic\MoteView\xserve2\bin`.

Se escribieron los parámetros correctos y se logró la conexión al `xserve` fuera de la aplicación de `MoteView`.

```
xserve -s=COM10 -b=57600 -l -xmlfile=XmlStream.xml -xmlp -xmlport=9005
```

Como se puede observar es la misma línea de comandos de la ventana de mensajes del servidor de la aplicación `MoteView`, ahora solo resta utilizarla dentro de nuestra propuesta de proyecto interfaz gráfica para más detalle de la implementación de nuestra interfaz gráfica ver Anexo B.

El `XSERVE` es un servicio que utiliza `MoteView` para leer los datos del puerto serial y transferirlos a la base de datos, en este caso `PostgreSQL`. En la sección siguiente se revisa `xserve` ya que es el mecanismo que se usa para atrapar los datos sin la necesidad de `MoteView`, pero antes se muestran las estructuras de información utilizadas para los sensores, así como el tipo de tablas para el almacenamiento de los mismos.

Gracias este descubrimiento, se tomó la decisión de utilizar al `xserve` para crear una interfaz independiente de la aplicación proporcionada por el fabricante, de esta manera poder desarrollar una interfaz que satisfaga nuestras necesidades, sobre todo en la manipulación de la gran cantidad de datos que las redes de sensores generan, es pertinente mencionar que para lograrlo será necesario hacer algunos cambios a la base de datos, la cual es descrita en la próxima sección, que explica más a fondo en qué consisten las modificaciones planteadas, sobre todo a la base de datos.

Las tablas se agrupan en bases de datos, y una colección de bases de datos administradas por una única instancia de servidor `PostgreSQL` constituye un clúster de base de datos.

A continuación se puede observar el administrador del manejador de `PostgreSQL` Figura 3.25, el cual nos muestra la estructura del servidor, el cual es llamado `task`, el cual a su vez contiene las diferentes bases de datos, en nuestro proyecto el



nombre de la base de datos creada por el proceso de instalación es **task**, y dentro de la cual utilizaremos la tabla llamada **mts400_results**.

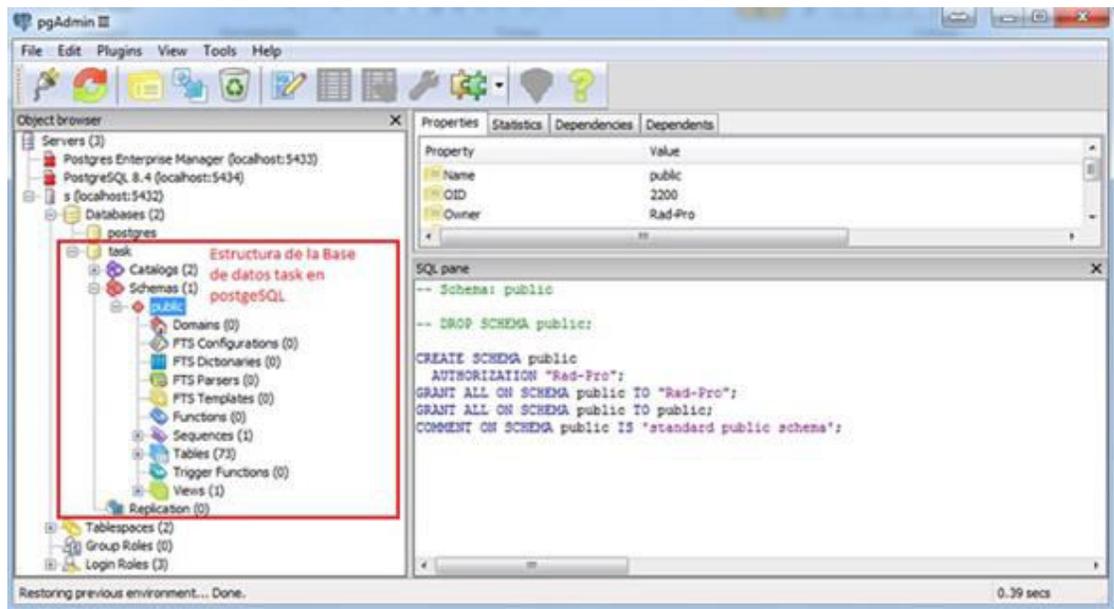


Figura 3.27 Muestra la Consola del Administrador de PostgreSQL, mostrando la B.D. task.

Para poder utilizar la tabla **mts400_results** originalmente instalada para ser usada por el MoteView, fue necesario realizar algunas modificaciones a su estructura para hacerla compatible con las necesidades del lenguaje de programación seleccionado para su acceso y manipulación.

La tabla **mts400_results**, que ubicamos en la parte “*Descripción general de la capa Servidor*” requirió de una modificación en su estructura para lograr la manipulación de los datos almacenados en ella con nuestra propuesta de interfaz, específicamente los cambios consistieron en la modificación del campo **result_time**, que tenía la condición de ser un campo de tipo *timestamp* (fecha y hora), que para efectos de indexación fue separada en dos campos, uno el mismo **result_time** pero como tipo *date* (fecha), y se agregó un campo hora para almacenar de manera separada la hora de registro del datos capturado, también se agregó el campo **conseid** de tipo integer para almacenar un consecutivo de cada registro en la tabla.

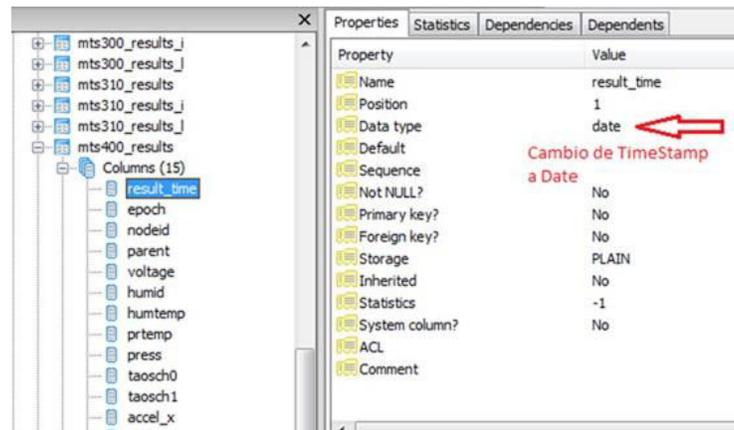


Figura 3.28. Cambio de tipo de TimeStamp a Date.

Se modificó la estructura de la tabla quedando de la siguiente manera:

```
File Edit Query Favourites Macros View Help
task on tele@localhost:5432
SQL Editor Graphical Query Builder
CREATE TABLE mts400_results
(
  result_time date,
  epoch integer,
  nodeid integer,
  parent integer,
  voltage integer,
  humid integer,
  humtemp integer,
  prtemp integer,
  press integer,
  taosch0 integer,
  taosch1 integer,
  accel_x integer,
  accel_y integer,
  hora time without time zone DEFAULT (now)::time without time zone,
  conseid integer DEFAULT nextval('conse'::regclass)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE mts400_results OWNER TO tele;
```

Figura 3.29 Estructura de la tabla mts400_result con los nuevos campos.

Se creó una secuencia para el campo conseid para generar el número consecutivo de cada registro.

Además de agregar los campos en la inserción de los datos. Proceso que es disparado por el XServe cuando recibe datos de los sensores.



```
CREATE OR REPLACE RULE cache_mts400_results AS
ON INSERT TO mts400_results DO { DELETE FROM mts400_results_l
WHERE mts400_results_l.nodeid = new.nodeid;
INSERT INTO mts400_results_l (result_time, epoch, nodeid, parent, voltage, humid, humtemp, prtemp, press, taosch0, taosch1, accel_x, accel_y)
VALUES (new.result_time, new.epoch, new.nodeid, new.parent, new.voltage, new.humid, new.humtemp, new.prtemp, new.press, new.taosch0, new.taosch1,
new.accel_x, new.accel_y, hora, conseid);
};
```

Figura 3.30 Consulta (query) que inserta los datos a la tabla mts400_results.

A continuación se muestra la estructura final de la tabla mts400_results, con datos recabados por los sensores utilizando el administrador de PostgreSQL.

	result_time date	epoch integer	nodeid integer	parent integer	voltage integer	humid integer	humtemp integer	prtemp integer	press integer	taosch0 integer	taosch1 integer	accel_x integer	accel_y integer	hora time without zone	conseid integer
1	2015-03-31		7814	0	381	730	6541	27074	16390	65533	0	278	489	14:13:12	170333
2	2015-03-31		7814	0	381	730	6541	27075	16390	65533	0	278	489	14:18:12	170339
3	2015-03-31		7814	0	381	730	6541	27075	16389	65533	0	277	489	14:23:12	170345
4	2015-03-31		7814	0	381	727	6542	27075	16391	65533	0	278	489	14:28:12	170351
5	2015-03-31		7814	0	381	726	6542	27075	16390	65533	0	278	489	14:33:12	170357

Figura 3.31 Datos de la Tabla mts400_results, con la estructura y datos del proyecto.

No es la intención en este trabajo abundar en el funcionamiento del manejador de bases de datos PostgreSQL, lo que se ha descrito en este apartado es importante para mostrar los cambios que se realizaron a la base de datos para lograr su funcionamiento para el uso en nuestro proyecto, en el Anexo B se puede encontrar más información de la instalación y modificación de la base de datos de la plataforma MoteView.



Capítulo IV: Resultados

Posterior a la configuración del ambiente de trabajo y de la exploración preliminar de las plataformas tecnológicas Wasmote y Motes, nuestro objetivo final fue el desarrollo de la interfaz de visualización del comportamiento de las redes. A pesar de haber explorado y recolectado información con la red Wasmote, debido a las cuestiones técnicas explicadas en la sección anterior, sin embargo, en la interfaz de visualización solo se consideró la plataforma Motes. En las siguientes subsecciones entonces se describe la implementación de esta interfaz.

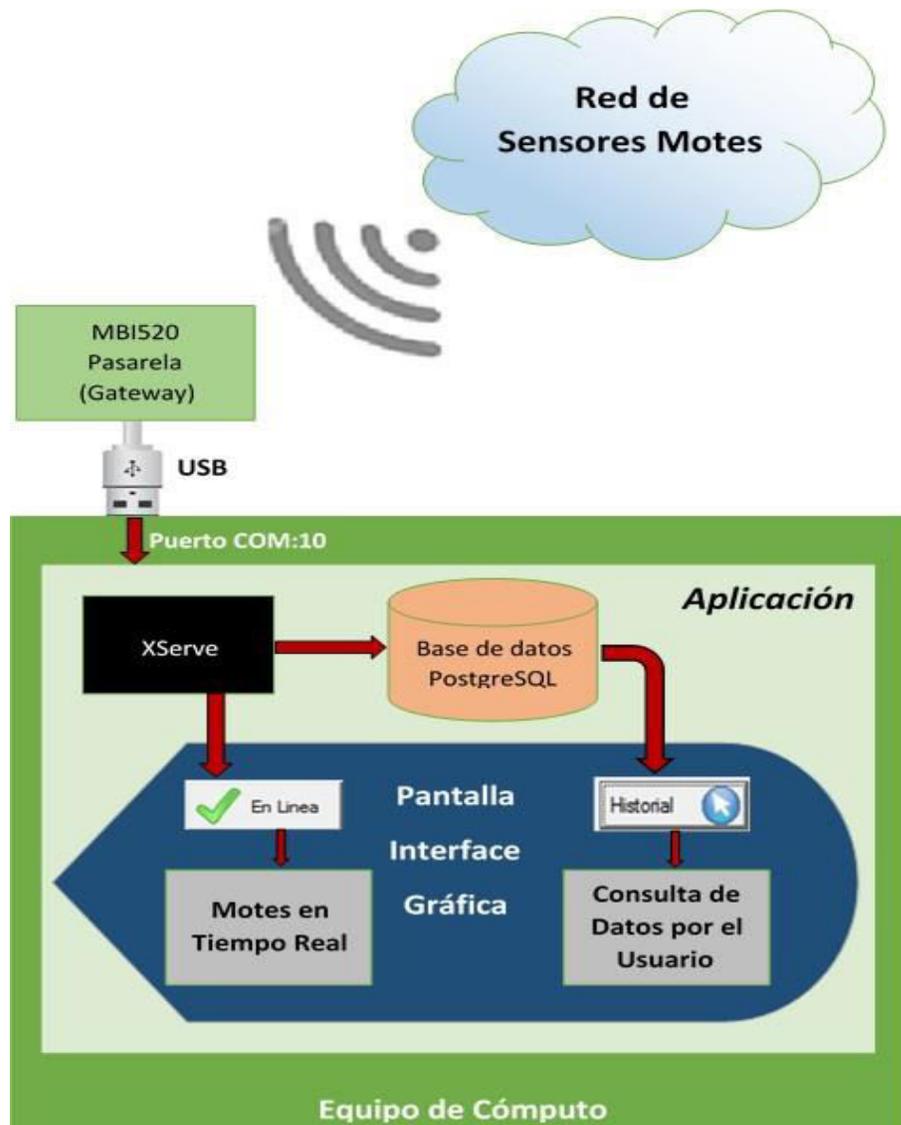




Figura 4.0 Diagrama de Bloques de la Interfaz Gráfica

La interfaz gráfica recoge toda la experiencia acumulada de la investigación del funcionamiento de los Motes, la cual es indispensable para poder conformar un conocimiento y de esta manera tener la posibilidad de realizar una aportación que ayude a facilitar el camino de quienes se interesen en esta tecnología, tanto en el ámbito académico como en el público y privado.

La figura 4.0 muestra el funcionamiento en bloques de la interfaz gráfica, para la captación de datos de una red de sensores inalámbricos tipo Motes.

Red de Sensores Motes: Está conformado por los N dispositivos inalámbricos que se pretendan utilizar en la red, estos constan de dos partes, primera el nodo sensor procesador / tarjeta de radio: módulos IRIS / MICA para habilitar el sistema de medición de redes de sensores inalámbricos de baja potencia. Disponible en 868/916 MHz o 2,4 GHz.

La segunda es la tarjeta (board) multi-sensor MTS400 que incluye capacidades de temperatura, humedad, presión barométrica, aceleración y detección de luz ambiental.

MBI520 Puerta de enlace a (Gateway): Este dispositivo es el encargado de recibir los datos enviados por los diferentes sensores distribuidos en la red, esta tarjeta tiene una conexión tipo USB que se conecta directamente al equipo de cómputo que contiene los programas necesarios para la recepción e interpretación de datos, en este caso al ser un dispositivo serial utiliza los puertos COM:X para establecer la comunicación.

XServe: Este es un programa residente en la memoria del computador que sirve como enlace entre el dispositivo MBI520 (Gateway), que recibe los datos de los sensores, para ejecutar dos acciones, las cuales son insertar los datos a la base de datos, así como interactuar con la aplicación para poder saber que sensor está conectado en tiempo real. En lo personal le llamé la caja negra ya que realiza una



cantidad importante de actividades las cuales no investigamos por no ser objeto de este proyecto.

Base de Datos de PostgreSQL: El manejador de base de datos PostgreSQL se encarga del almacenamiento de los datos generados por la red de sensores, ya en la sección de evaluación preliminar y exploración de datos hablamos de ella, por lo que sólo resta señalar que es de vital importancia ya que representa el almacenamiento de todos los registros generados por los sensores y nos permitirá darle forma a la información que permitirá al usuario final una adecuada toma de decisiones.

Aplicación

Esencialmente son dos las actividades principales que realiza la aplicación, la primera es permitir la lectura de los sensores en tiempo real, lo cual es importante ya que nos permite saber la condición o estado, como el de la batería, o si el sensor cambió de posición, además claro de las funciones esenciales de los sensores (Temperatura, humedad, luminosidad etc.).

La segunda actividad de la aplicación es la manipulación de los datos almacenados en la base de datos, la cual le da sentido a estos, de tal forma que la información generada puede ser presentada de diversas formas que permitan al usuario, tomar la mejor decisión posible, pero con la certeza que da el respaldo de un gran de datos históricos que la respalde.

Las dos actividades antes mencionadas representan son la propuesta de interfaz gráfica objeto de este proyecto.

A continuación se muestran varias pantallas que son parte de la interface visual de la red de sensores.



Figura 4.1 Seguimiento de los sensores activos de la red.

La figura 4.1 muestra como cada sensor, está enmarcado en el cuadro de color verde, que hace alusión a que esta encendido, lo cual representa que el nodo está transmitiendo en tiempo real, y a su vez se va generando un listado de estos nodos, los cuales son descritos por su id o identificador de cada uno de los que se encuentran activos, también muestra la presencia del dispositivo que permite la transmisión hacia la aplicación, la puerta de enlace (gateway) con una nomenclatura GW y un identificador como lo representa el número uno.

Además de lo antes mencionado el usuario puede hacer clic en el cuadro del sensor deseado una vez que hace el clic el cuadro muestra una línea punteada que denota la selección, al hacer doble clic mostrará una consulta de los datos específicos del nodo seleccionado en este caso el 7818.



Nodo ID	Fecha	Hora	Padre	Voltaje	Hum ID	Hum Temp	Pr Temp	Press	Accel x	Accel y
7818	28/03/2015	12:00 a	0	647	492	8.309	29.385	16.527	261	537
7818	28/03/2015	12:00 a	0	647	492	8.309	29.386	16.527	260	538
7818	28/03/2015	12:00 a	0	645	492	8.309	29.384	16.527	260	537
7818	28/03/2015	12:00 a	0	645	493	8.309	29.384	16.526	261	536
7818	28/03/2015	12:00 a	0	645	490	8.309	29.386	16.527	259	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.383	16.529	260	537
7818	28/03/2015	12:00 a	0	647	490	8.309	29.386	16.527	260	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.386	16.526	260	538
7818	28/03/2015	12:00 a	0	645	490	8.309	29.385	16.526	261	536
7818	28/03/2015	12:00 a	0	647	494	8.309	29.385	16.527	260	537
7818	28/03/2015	12:00 a	0	647	490	8.309	29.386	16.526	260	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.384	16.527	260	538
7818	28/03/2015	12:00 a	0	645	490	8.309	29.386	16.528	261	537
7818	28/03/2015	12:00 a	0	647	494	8.309	29.384	16.527	260	538

Figura 4.2 Ventana muestra la relación de datos filtrados el nodo 7818

Como se puede apreciar en la figura 4.2 nos muestra la selección que previamente se hizo al seleccionar el botón del nodo deseado, en la ventana de nodos en línea, este identifica cuál nodo es el que se está accedendo y muestra sólo sus registros (7818).

Nodo ID	Fecha	Hora	Padre	Voltaje	Hum ID	Hum Temp	Pr Temp	Press	Accel x	Accel y
7818	28/03/2015	12:00 a	0	647	492	8.309	29.385	16.527	261	537
7818	28/03/2015	12:00 a	0	647	492	8.309	29.386	16.527	260	538
7818	28/03/2015	12:00 a	0	645	492	8.309	29.384	16.527	260	537
7818	28/03/2015	12:00 a	0	645	493	8.309	29.384	16.526	261	536
7818	28/03/2015	12:00 a	0	645	490	8.309	29.386	16.527	259	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.383	16.529	260	537
7818	28/03/2015	12:00 a	0	647	490	8.309	29.386	16.527	260	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.386	16.526	260	538
7818	28/03/2015	12:00 a	0	645	490	8.309	29.385	16.526	261	536
7818	28/03/2015	12:00 a	0	647	494	8.309	29.385	16.527	260	537
7818	28/03/2015	12:00 a	0	647	490	8.309	29.386	16.526	260	538
7818	28/03/2015	12:00 a	0	647	494	8.309	29.384	16.527	260	538
7818	28/03/2015	12:00 a	0	645	490	8.309	29.386	16.528	261	537
7818	28/03/2015	12:00 a	0	647	494	8.309	29.384	16.527	260	538

Figura 4.3 Ventana de parámetros para la selección de la información.



Una vez que tenemos la selección del nodo que queremos consultar la información, tenemos los botones de Temp y Voltaje, que nos permite generar información de cada uno de ellos, para poder realizar esta tarea, presionamos el botón deseado y nos lo muestra la ventana de la figura 4.3, la cual permite ingresar los parámetros de un rango de fechas, una vez que determinamos que rango se desea consultar, se acepta la selección y podemos ver el resultado en la figura 4.4, donde se representa de manera gráfica los valores de temperatura en un periodo determinado por el rango de fechas previamente ingresados en la ventana de parámetros.

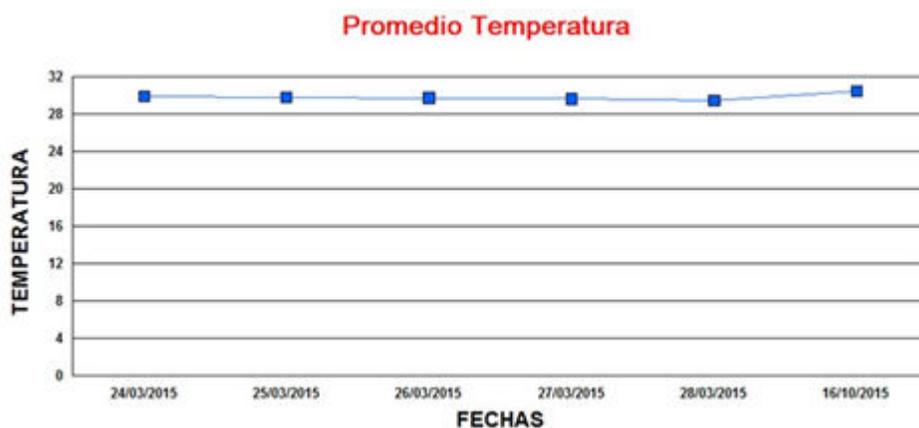


Figura 4.4 Reporte gráfico del promedio de temperaturas censado en el tiempo

Como podemos ver el gráfico nos ilustra un rango de temperaturas, por otro lado nos muestra las fechas en las que se registró el evento y por último las líneas y cuadros azules identifican la temperatura máxima en promedio que se alcanzó para la fecha específica.

Es importante señalar que este promedio es de todos los sensores distribuidos en el edificio seleccionado para estas mediciones.

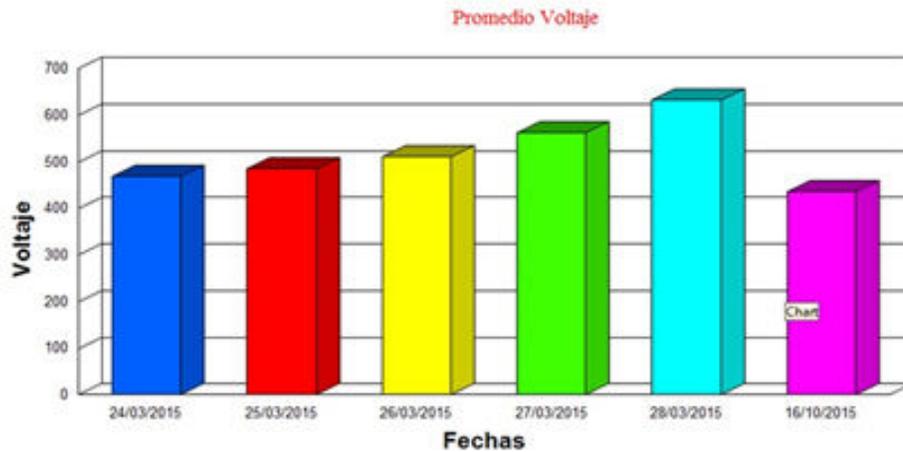


Figura 4.5 Representación gráfica de nivel de voltaje a través del tiempo

De igual manera la figura 4.5, nos permite identificar, cómo ha sido el comportamiento del voltaje de los sensores, a través del mismo periodo, es muy importante tener acceso a esta información ya que nos permite, dar seguimiento puntual de los niveles de energía de la batería de cada nodo, con lo que garantizamos evitar la pérdida de datos por motivo de que algún nodo se le haya agotado la batería.



V Conclusiones

5.1 Conclusiones

1. Se ha diseñado la interfaz gráfica de una red inalámbrica de sensores que permite monitorear el estado de los sensores, sus mediciones y almacenamiento para su presentación de manera sencilla para el usuario en forma de reportes gráficos.
2. El prototipo elaborado y su aplicación satisface las necesidades de medición y reporte requeridas en el diseño del sistema de monitoreo.
3. Bajo costo por concepto de consumo de energía.
4. Por lo tanto se concluye que las WSN son la mejor opción ante problemáticas donde la información debe ser captada desde diferentes puntos. Estas redes son relativamente económicas ante otras soluciones y permiten monitorear diferentes parámetros sin la necesidad de realizar grandes variaciones en la configuración de la misma.
5. Además se demostró que es posible generar interfaces propias, con esto eliminando posibles limitaciones de código cerrado que ofrecen los fabricantes de las redes de sensores inalámbricos.

5.2 Posibles líneas futuras de este Proyecto

- Se pueden generar oportunidades de negocio, ya que nuestra región cuenta con opciones productivas como el cultivo de la manzana, el cual tiene un gran potencial de uso esta tecnología, en la predicción del clima, en especial del granizo y de las heladas de temporada.
- Se podrían desarrollar aplicaciones basadas en sensores para el control de procesos industriales, las empresas maquiladoras, promueven el uso de la tecnología tanto en sus procesos productivos, como en administrativos para realizar ahorros cada ciclo fiscal, se puede desarrollar una redes inalámbricas para monitoreo y control de energía, de ambiente, inventarios etc., las posibilidades son muchas.





Referencias

- [1] Roger Smith, “RFID: A Brief Technology Analysis”, CTONet.org, 2004.
- [2] Jayro Santiago Paz, Diseño y construcción de un módulo transmisor receptor inalámbrico para el manejo de sensores con el protocolo estándar IEEE 802.15.4 (ZigBee) Tesis, Universidad Tecnológica de la Mixteca, 2008.
- [3] M. López Villaseñor, M. A. Gutiérrez Galindo, R. Marcelín Jiménez, “De sensores inalámbricos, mitigación de riesgos y sistemas auto-organizados”, Área de Redes y Comunicaciones, Depto. De Ing. Eléctrica, UAM–Iztapalapa, 2007.
- [4] Priya Asher, Ujala Daswani, “ An Overview of ZigBee Specifications and its Industrial Applications”, 1Department of Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering, Vidyavihar, Mumbai, India, 2013. 2 Department of Electronics and Telecommunication Engineering, Vivekanand Education Society’s Institute of Technology, Chembur, Mumbai, India. International Journal of Science and Research.
- [5] Laura Vanzago. Overview on 802.15.4/ZigBee, Advanced System Technology – R&D DICO. STMicroelectronics, May 2006.
- [6] Jin-Shyan Lee and Yuan-Ming Wang (2012). Experimental Evaluation of ZigBee-Based Wireless Networks in Indoor Environments. Journal of Engineering, 10.
- [7] Dora L. Castañeda Tibaquirá, Diana C. Bacca Quiroga, Gustavo A. Higuera Castro, Estudio De La Tecnología Zigbee Y La Implementación En La Aplicación De Sensores Remotos, Universidad Distrital Francisco José de Caldas Bogotá, Colombia 2010.
- [8] Juan Carlos Suárez Barón, Aplicación de tecnología Zigbee para el monitoreo de variables ambientales en invernaderos. Escuela de Ciencias Básicas, Tecnología e Ingeniería, Duitama, Colombia.
- [9] Pilar Elejoste, Ignacio Angulo, Asier Perallos, Aitor Chertudi, Ignacio Julio García Zuazola, Asier Moreno, Leire Azpilicueta, José Javier Astrain, Francisco Falcone 2, Jesús Villadangos, An Easy to Deploy Street Light Control System Based on Wireless Communication and LED Technology, Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain, Electrical and Electronic Engineering Department, Universidad Pública de Navarra, Pamplona 31006, Mathematics and Computer Engineering Department, Universidad Pública de Navarra, Pamplona 31006.



[10] David Magín Flórez-Rubio, Camilo Otálora-Sánchez, Carlos Iván Páez-Rueda, Diseño e implementación de una red de sensores para la adquisición de variables relacionadas con la vigilancia estructural de puentes, Ing. Univ. Bogotá (Colombia), 13, julio-diciembre de 2009.

[11] Santiago J. Barro Torres, Carlos J. Escudero Cascón. “Sistema Sensorial Inalámbrico para la Detección de Patologías en la Construcción”. Facultad de Informática. Universidad de la Coruña. 15071 Campus de Elviña.

[12] Lucas Iacono, Pablo Godoy, Osvaldo Marianetti, Carlos García Garino, Estudio de Plataformas de Hardware Empleadas en Redes de Sensores Inalámbricas, Instituto universitario para las Tecnologías y las Comunicaciones, UNCuyo, Facultad de ingeniería, Universidad Nacional de Cuyo, Facultad de Ingeniería, Universidad de Mendoza, LAPIC, Laboratorio de Investigación y Desarrollo para la Producción Integrada por Computadora, Carrera de Redes y Telecomunicaciones, ITU.

[13] Javier Vales-Alonso, Pablo López-Matencio, Francisco J. Gonzalez-Castaño 2, Honorio, Navarro-Hellín, Pedro J. Baños-Guirao, Francisco J. Pérez-Martínez, Rafael P. Martínez-Álvarez, Daniel González-Jiménez, Felipe Gil-Castiñira, Richard Duro-Fernández, Ambient Intelligence Systems for Personalized Sport Training, Published: 22 March 2010.

[14] Thierry Antoine-Santoni *, Jean-François Santucci, Emmanuelle de Gentili, Xavier Silvani and Frederic Morandini, Performance of a Protected Wireless Sensor Network in a Fire. Analysis of Fire Spread and Data Transmission, University of Corsica/UMR CNRS SPE, Quartier Grossetti 20250 Corte, France; Published: 24 July 2009.

[15] Absar-ul-Hasan¹, Ghalib A. Shah² & Ather Ali³, Intrusion Detection System using Wireless Sensor Networks, National University of Science and Technology, Islamabad, Pakistan, Center for Advanced Research in Engineering, Islamabad Pakistan, EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010).

[16] SungHan Sim, Jian Li, Hongki Jo, JongWoong Park, Soojin Cho, Billie F Spencer Jr, and HyungJo Jung, A wireless smart sensor network for automated monitoring of cable tension, School of Urban and Environmental Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan 689-798, Korea, Department of Civil, Environmental, and Architectural Engineering, University of Kansas, Lawrence, KS 66045, USA, Department of Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ 85721, USA, Department of Civil and Environmental engineering, KAIST, Daejeon 305-701, Korea, Department of Civil and Environmental Engineering,



University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA Published
13 December 2013.



Anexos

Anexo A

Introducción a Wasmote

Es preciso mostrar las características de los dispositivos waspmote, son circuitos inteligentes que son los encargados de realizar las diferentes mediciones en tiempo real que una ciudad, negocio, parcela, o un bosque necesitan.

Características físicas generales:

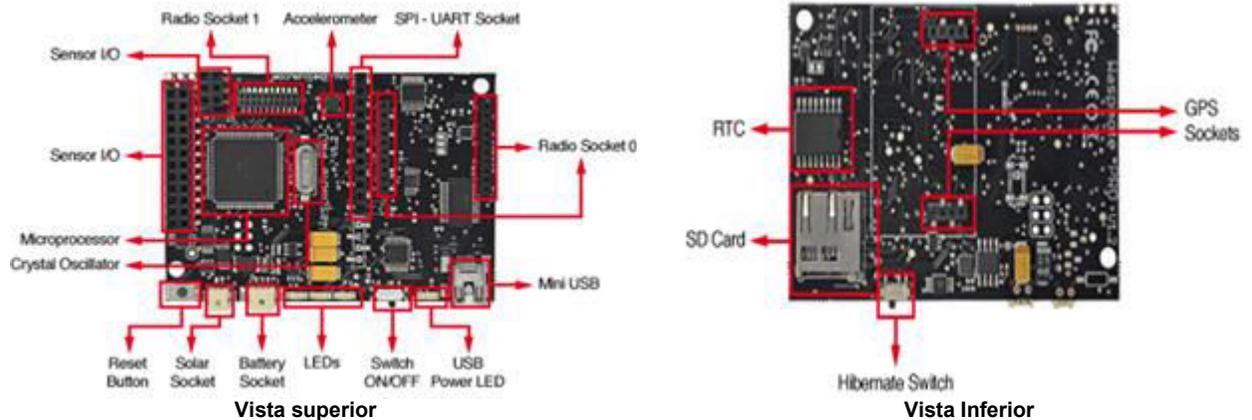


Figura A.1 Tarjetas Wasmote

Las imágenes no muestran una dimensión real de los dispositivos sus medidas son:

Peso: 20 gramos

Dimensiones: 73.5 x 51 x 13 mm

Rango de Temperatura: -10°C, +65°C

Consumo:

Encendido: 15 miliamperios

Dormido: 55 microamperios

Dormido profundo: 55 microamperios

Hibernación: 0.7 microamperios.

Sensores básicos:

Temperatura, aceleración, estado de la batería.



XBee



Model	Protocol	Frequency	Tx power	Sensitivity	Range*
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB-Pro	Zigbee-Pro	2.4GHz	50mW	-102dBm	7000m
XBee-868	RF	868MHz	315mW	-112dBm	12Km
XBee-900	RF	900MHz	50mW	-100dBm	10Km

Antenna
2.4GHz: 2dBi/5dBi
868/900MHz: 4.5dBi
Connector: RPSMA

Topologies



Tree Mesh Star

Figura A.2 Características XBee.

ZigBee define tres tipos diferentes de dispositivos: coordinador, el Ruteador y dispositivo final.

Tipos de nodo / Muestra de una topología de red ZigBee básico.

Un coordinador tiene las siguientes características:

- Selecciona un canal y el PAN ID (ambos de 64 bits y 16 bits) para iniciar la red
- Puede permitir que los ruteadores y dispositivos finales unirse a la red
- Puede ayudar en el encaminamiento de datos
- No puede dormir - debe ser con alimentación de red
- Puede hace buffer a los paquetes de datos RF para dispositivo END en modo sleep.

API Operación

Operación de API es una alternativa a la operación transparente. La API basada en tramas extiende el nivel al que una aplicación host puede interactuar con las capacidades de red del módulo. Cuando está en modo API, todos los datos de entrada y salida del módulo están contenidos en tramas que definen las operaciones o eventos dentro del módulo.

Transmitir tramas de datos (recibida a través del pin DIN (pin 3)) incluye:

- Trama de RF de transmisión de datos



- Trama Comando (equivalente a comandos AT) Recibir tramas de datos (enviado el pin DOUT (pin 2)) incluyen:
- Trama de datos RF-recibido
- Respuesta de Comando
- Las notificaciones de eventos tales como reset, asociado, disociar, etc.

La API proporciona medios alternativos de configuración de los módulos y el enrutamiento de datos en la sede de la capa de aplicación. A host aplicación puede enviar tramas de datos al módulo que contiene la dirección y la carga útil de información en lugar de utilizar el modo de comando para modificar direcciones. El módulo enviará tramas de datos a la aplicación que contiene paquetes de estado; así como la fuente, y la información de carga útil de los paquetes de datos recibidos.

La opción de operación de API facilita muchas operaciones, como los ejemplos que se citan a continuación:

- > La transmisión de datos a múltiples destinos, sin entrar en la modalidad de mandatos
- > Recibir éxito / fracaso de estado de cada paquete de RF transmitida
- > Identificar la dirección de origen de cada paquete recibido.

Tipos de Dispositivos

Un ruteador tiene las siguientes características:

- Debe unirse a un ZigBee PAN antes de que pueda transmitir, recibir o rutear datos
- Luego de ingresar, puede permitir que los ruteadores y dispositivos finales pueden unirse a la red
- Luego de ingresar, puede ayudar en el encaminamiento de datos
- No puede dormir - se deben alimentar la red eléctrica.
- Puede almacenar los paquetes de datos RF para los dispositivos finales.

Un dispositivo final tiene las siguientes características:

- Debe unirse a un ZigBee PAN antes de que pueda transmitir o recibir datos



- No permite que los dispositivos se unan a la red
- Siempre debe transmitir y recibir datos de RF a través de su padre. No puede enrutar datos.
- Puede entrar en modo de bajo consumo para ahorrar energía y puede ser alimentado por batería.

En las redes ZigBee, el coordinador debe seleccionar un PAN ID (64 bits y 16 bits) y el canal para iniciar una red.

Después de eso, se comporta esencialmente como un ruteador. El coordinador y los ruteadores pueden permitir que otros dispositivos se unan a red y puedan rutear datos.

Después si un dispositivo final se une a un router o coordinador, debe ser capaz de transmitir o recibir datos de RF a través de ese ruteador o coordinador. El ruteador o coordinador que permitió un dispositivo final para unirse se convierte en el "padre" del dispositivo final. Dado que el dispositivo final puede dormir, los padres deben ser capaces de amortiguar o retener los paquetes de datos entrantes destinados para el dispositivo final hasta que el dispositivo final es capaz de despertar y recibir los datos.

PAN ID

Redes ZigBee se denominan redes de área personal o PAN. Cada red se define con un identificador único PAN (PAN ID). Este identificador es común entre todos los dispositivos de la misma red. Los dispositivos ZigBee son pre configurados con un ID PAN para unirse, o pueden descubrir redes cercanas y seleccione un PAN ID para unirse.

ZigBee es compatible tanto con una de 64 bits y un PAN ID de 16 bits. Ambas identificaciones PAN se utilizan para identificar de forma exclusiva una red.

Los dispositivos de la misma red ZigBee deben compartir el mismo PAN ID de 64 bits y de 16 bits. Si hay varias redes ZigBee operando dentro de un rango entre sí, cada uno debe tener identificadores únicos del PAN.



El PAN ID de 16 bits se utiliza como capa MAC con un campo de direccionamiento en todas las transmisiones de datos RF entre dispositivos en una red. Sin embargo, debido al espacio limitado de abordar el PAN ID de 16 bits (65.535 posibilidades), hay una posibilidad de que múltiples redes ZigBee (dentro del alcance del otro) podría utilizar el mismo 16-bit ID PAN. Para resolver los posibles conflictos de ID PAN de 16 bits, la ZigBee Alliance creó un PAN ID de 64 bits.

El PAN ID de 64 bits (también llamado el extendido PAN ID), está destinado a ser un valor único, en duplicado. Cuando un coordinador comienza una red, se puede comenzar una red con un PAN ID pre-configurada de 64 bits, o puede seleccionar una PAN ID aleatorio. El PAN ID de 64 bits se utiliza durante la unión; si un dispositivo tiene una pre-configurada, sólo se unirá a una red con el mismo PAN ID 64-bit. De lo contrario, un dispositivo podría unirse a cualquier PAN detectado y heredar el PAN ID de la red cuando se une. El PAN ID de 64 bits se incluye en todos los dispositivos ZigBee y se utiliza la PAN ID de 16 bits para la resolución de conflictos.

Los ruteadores y dispositivos finales son típicamente configurados para unirse a una red con cualquier PAN ID de 16 bits siempre que el PAN ID de 64 bits es válido. Coordinadores suelen elegir un PAN ID de 16 bits al azar para su red.

Dado que el PAN ID de 16 bits sólo permite hasta 65.535 valores únicos, y puesto que el PAN ID de 16 bits se selecciona al azar, existen disposiciones en ZigBee para detectar si dos redes (con diferentes 64-bit PAN IDS) están operando en el mismo ID de PAN 16-bit. Si se detecta un conflicto, la pila ZigBee puede llevar a cabo la resolución de conflictos PAN ID para cambiar el PAN ID de 16 bits de la red con el fin de resolver el conflicto. Ver la especificación ZigBee para más detalles.

Para resumir, los ruteadores y dispositivos ZigBee finales deben estar configurados con el PAN ID de 64 bits de la red que quieren unirse. Por lo general adquieren el PAN ID de 16 bits cuando se unen a una red.

Transmisión de Datos



Paquetes de datos ZigBee pueden enviarse ya sea como transmisiones unicast o broadcast. Unicast rutea los datos de transmisiones desde el dispositivo de una fuente a un dispositivo de destino, mientras que las transmisiones broadcast se envían a muchos o todos los dispositivos en la red.

Transmisiones Broadcast.

Transmisiones Broadcast dentro del protocolo ZigBee están destinadas a ser propagadas a través de toda la red de tal manera que todos los nodos reciben la transmisión. Para lograr esto, el coordinador y todos los ruteadores que reciben una transmisión de broadcast se debe retransmitir el paquete tres veces.

Nota: cuando un Ruteador o coordinador entrega una transmisión de Broadcast a un dispositivo secundario final, la transmisión sólo se envía una vez (inmediatamente después de que el dispositivo final se despierta y jala todos los datos nuevos).

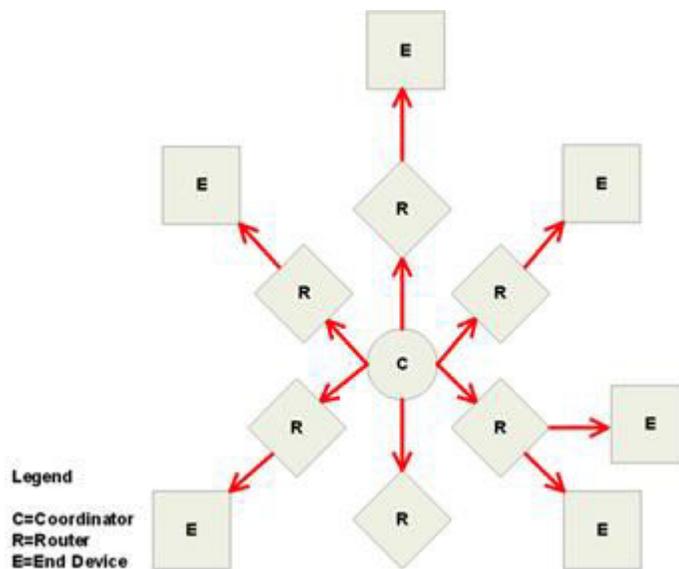


Figura A.3 Transmisión de Datos tipo emisión(Broadcast)

Transmisión de Datos Broadcast

Cada nodo que transmite la emisión también creará una entrada en una tabla de transmisión local. Esta entrada se utiliza para realizar un seguimiento de cada



paquete de broadcast recibido para asegurar que los paquetes no se transmiten sin cesar. Cada entrada persiste durante 8 segundos. La tabla de transmisión de broadcast tiene 8 entradas.

Para cada transmisión, la pila ZigBee debe reservar espacio de búfer para obtener una copia del paquete de datos. Esta copia se usa para retransmitir el paquete según sea necesario. Los paquetes grandes requerirán más espacio de almacenamiento intermedio. Esta información sobre el espacio de amortiguación proporciona conocimientos generales; el usuario no puede cambiar cualquier espacio del búfer. El espaciado en el buffer es manejado automáticamente por el módulo XBee.

Puesto que las transmisiones broadcast se retransmiten por cada dispositivo en la red, los mensajes de este tipo deberían usarse con moderación.

Transmisiones Unicast

Transmisiones unicast son enviadas de un dispositivo de origen a otro dispositivo de destino. El dispositivo de destino podría ser un vecino inmediato de la fuente, o podría estar a varios saltos de distancia. Transmisiones unicast que se envían a lo largo de un camino de saltos múltiples requieren algunos medios para establecer una ruta al dispositivo de destino.

Direccionamiento

Como se ha mencionado anteriormente, cada dispositivo en una red ZigBee tiene tanto una dirección de 16 bits (de red) y una dirección de 64 bits (extendido). La dirección de 64 bits es única y asignada al dispositivo durante la fabricación, y el Dirección de 16 bits se obtiene después de unirse a una red.

La dirección de 16 bits también puede cambiar bajo ciertas condiciones.

Cuando se envía una transmisión de unicast, la capa de red ZigBee utiliza la dirección de 16 bits del destino y cada salto para enrutar el paquete de datos. Si no se conoce la dirección de 16 bits del destino, la pila ZigBee incluye una



disposición para descubrir automáticamente la dirección de 16 bits del dispositivo de destino antes de enrutar los datos.

Para descubrir una dirección de 16 bits de un mando a distancia, el dispositivo de iniciación al descubrimiento envía una transmisión broadcast a la dirección de descubrimiento. La detección de una dirección broadcast incluye la dirección de 64 bits del dispositivo remoto cuya dirección de 16 bits se solicita. Todos los nodos que reciben esta transmisión deben comprobar la dirección de 64 bits en la carga útil y lo comparan con su propia dirección de 64 bits. Si las direcciones coinciden, el dispositivo envía un paquete de respuesta de vuelta al iniciador. Esta respuesta incluye la dirección de 16 bits. Cuando se recibe la respuesta de descubrimiento, el iniciador entonces transmitirá los datos.

Tabla de direcciones

Cada dispositivo ZigBee mantiene una tabla de direcciones que se asigna una dirección de 64 bits a una dirección de 16 bits. Cuando una transmisión se dirige a una dirección de 64 bits, la pila ZigBee busca en la tabla de direcciones para una entrada con una dirección de 64 bits a juego, con la esperanza de determinar la dirección de 16 bits del destino. Si no se encuentra una dirección que coincida de 16 bits, la pila ZigBee realiza un proceso de descubrimiento de direcciones para encontrar la dirección de 16 bits actual del dispositivo.

64-bit Address	16-bit Address
0013 A200 4000 0001	0x4414
0013 A200 400A 3568	0x1234
0013 A200 4004 1122	0xC200
0013 A200 4002 1123	0xFFFE (unknown)

Figura A.4 Tabla de Direcciones

Los módulos XBee pueden almacenar hasta 10 entradas de la tabla de direcciones. Para las aplicaciones en un solo dispositivo (por ejemplo,



coordinador) puede enviar transmisiones unicast a más de 10 dispositivos, la aplicación debe implementar una tabla de direcciones para almacenar las direcciones de 16 bits y 64 bits para cada dispositivo remoto. Cualquier XBee al que enviaremos los datos de más de 10 dispositivos a distancia también se debe usar el firmware API. La aplicación puede entonces enviar direcciones tanto la de 16 bits y de 64 bits para el XBee en modo API transmitir tramas que reducirán significativamente el número de descubrimientos de direcciones de 16 bits y mejorar el rendimiento de datos.

Si una aplicación apoyará una tabla de direcciones, el tamaño ideal debe ser mayor que el número máximo de dispositivos de destino con los que se comunicará. Cada entrada de la tabla de direcciones debe contener una dirección de destino de 64 bits y su última dirección conocida de 16 bits.

Al enviar una transmisión a una dirección de 64 bits de destino, la aplicación debe buscar en la tabla de direcciones para una dirección de 64 bits en juego. Si se encuentra una coincidencia, la dirección de 16 bits se debe rellenar en el campo de dirección de 16 bits de la trama de API. Si no se encuentra una coincidencia, la dirección de 16 bits se debe establecer en 0xFFFFE (desconocido) en la trama de transmisión API.

La API proporciona indicación de dirección de 16 bits de un dispositivo remoto en las siguientes tramas:

- Todos reciben secuencias de datos
 - Rx de datos (0x90)
 - Rx de datos explícitos (0x91)
 - IO datos de muestra (0x92)
 - Nodo Indicador de identificación (0x95)
 - Ruta Indicador de grabación (0xA1)
 - etcétera
- trama de estado de transmisión (0x8B)



La aplicación siempre debe actualizar la dirección de 16 bits en la tabla de direcciones cuando uno de estas tramas se recibe para asegurar que la tabla tiene la dirección más reciente conocida de 16 bits. Si se produce un fallo en la transmisión, la aplicación debe establecer la dirección de 16 bits de la tabla para 0xFFFE (desconocido).

Cada transmisión unicast puede soportar hasta 84 bytes de carga útil de RF. (Habilitación de la seguridad o el uso de enrutamiento de origen pueden reducir este número. Véase el comando NP para más detalles.) Sin embargo, el firmware XBee ZB admite una característica denominada ZigBee fragmentación que permite que un solo gran paquete de datos que se divide en múltiples transmisiones de RF y vuelto a montar por el receptor antes de enviar los datos a cabo su UART. Esto se muestra en la imagen de abajo.

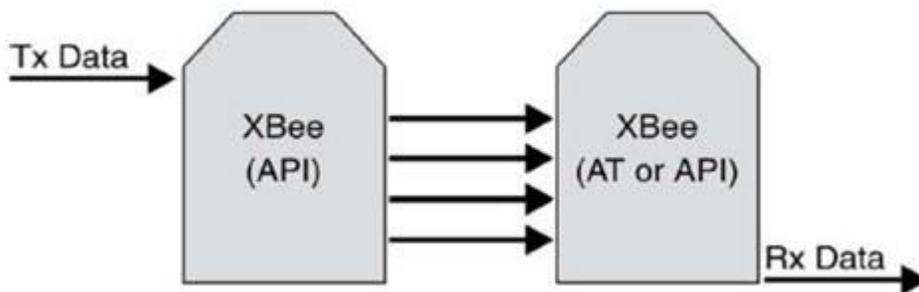


Figura A.5 Transmisión XBee

La trama de transmisión API puede incluir hasta 255 bytes de datos, que se divide en múltiples transmisiones y vuelo a montar en el lado receptor. Si uno o más de los mensajes fragmentados no son recibidos por el dispositivo receptor, del receptor descartará todo el mensaje, y el remitente indicará un fallo en la transmisión en el status de la trama de API Tx. El firmware ZB sólo puede admitir un paquete fragmentado en un momento en el nodo de recepción, debido a las limitaciones de memoria.



Anexo B

Introducción a Motes

Información General (MoteView)

MoteView está diseñado para ser una interfaz ("nivel de cliente") entre un usuario y una red desplegada de sensores inalámbricos. MoteView proporciona las herramientas para simplificar la implementación y seguimiento. También hace que sea fácil de conectar a una base de datos, para analizar, y para representar gráficamente las lecturas del sensor.

Figura 2.2 representa un marco de tres partes para la implementación de un sistema de redes de sensores. La primera parte es la capa o el sensor de red de malla Mote. Los Motes se programan con XMesh / TinyOS software embebido ("Aplicación") para realizar una tarea específica: por ejemplo, el seguimiento microclima, seguimiento de activos, detección de intrusos, etc. El segundo nivel o capa Server proporciona el registro de datos y servicios de base de datos. En esta capa lecturas de los sensores llegan a la estación base (por ejemplo, MIB510, MIB520, MIB600, compuerta de entrada) y se almacenan en un servidor. La tercera parte es el nivel de cliente en el que herramientas de software ofrecen visualización, monitoreo y herramientas de análisis para mostrar e interpretar datos del sensor. El propósito de este documento es explicar las características de MoteView y proporcionar información sobre las aplicaciones compatibles en la capa de aplicaciones Mote, las plataformas Mote y placas de sensores



Placas de sensores compatibles y Plataformas Mote.

MoteView soporta todas las tarjetas de adquisición de MEMSIC sensores y datos (Tabla 1.1), así como la MICA2, MICA2DOT y plataformas de procesador/radio MicaZ (Tabla 1-2) . Además, MoteView se puede utilizar para implementar y monitorear plataformas de sensores integrados, tales como el sistema seguridad e detección/intrusiones y el sistema de monitoreo ambiental MEP.

Las solicitudes admitidas Mote Software XMesh es un protocolo de multi-salto de red de malla de Memsic que tiene varias opciones, entre ellas de escucha de baja potencia, sincronización de tiempo, los modos de suspensión, cualquiera a la base y de base a cualquier enrutamiento. Todas nuestras placas de sensores y de adquisición de datos son compatibles con las aplicaciones habilitadas XMesh.

Instalación

Las plataformas para PC soportadas son:

- *Windows XP. Home*
- *Windows XP Professional*
- *Windows 2000 con sp4*
- *Windows 7 Professional*

Nota: El sistema de archivos debe ser NTFS, si se instala en Windows 7 necesita privilegios de supervisor porque la instalación crea un usuario para la base de datos PostgreSQL. La instalación pide el Microsoft .NET Framework 1.1, el cual ya está instalado en esta versión se sugiere omitir la instalación del mismo.

Instalación del software MoteView 2-1-4.



Figura B.1 Instalación Modificada

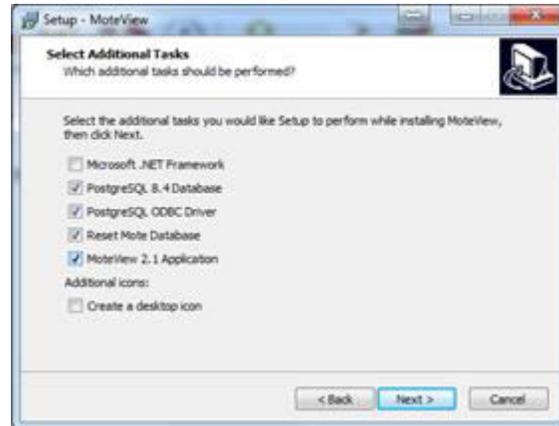


Figura B.2 Instalación Original

Si en el proceso de instalación tuvo algún problema hay que revisar si creo el usuario postgre en la ventana de administración de equipos en la sección usuarios y grupos locales, en la sección de usuarios aparece un usuario postgres, si no es así usted puede crear el usuario por su cuenta de la siguiente manera.

Revisión de la existencia de usuario postgres:

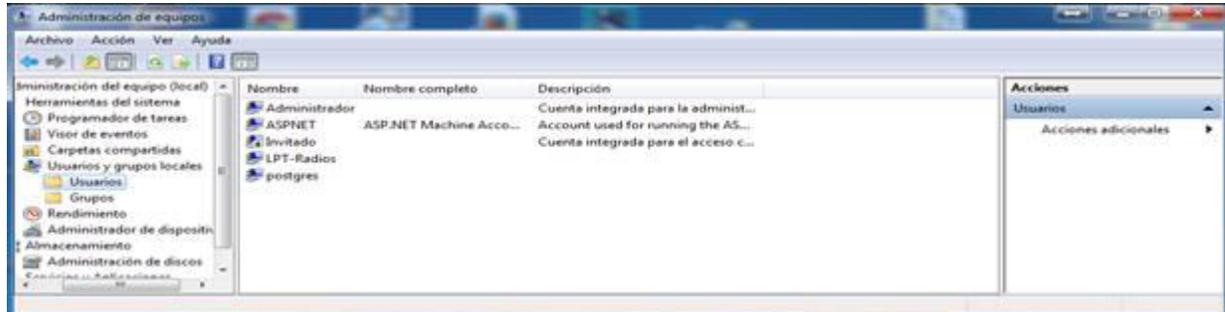


Figura B.3 Vista de usuarios en el sistema operativo Windows 7.

Si el usuario no fue creado durante la instalación lo podemos crear de manera manual como se muestra:

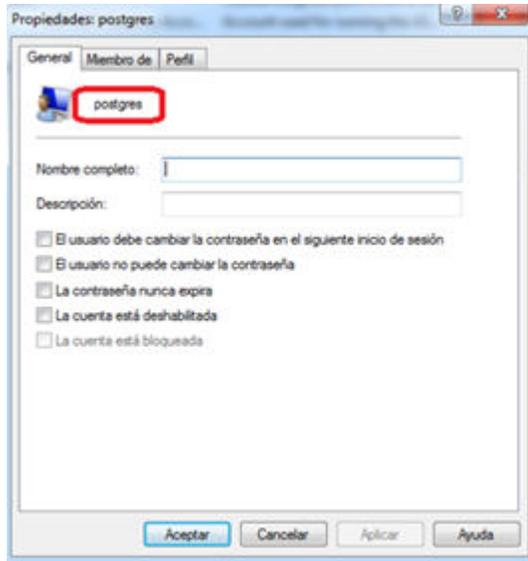


Figura B.4 Pestaña General Propiedades PostgreSQL

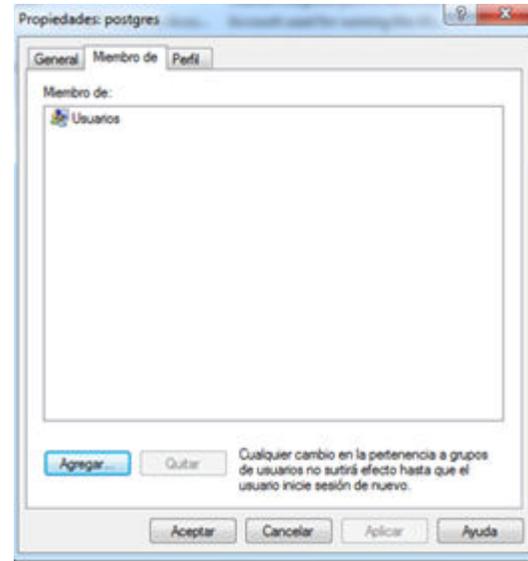


Figura B.5 Pestaña Miembros de PostgreSQL

Una vez creado el usuario corra de nuevo el proceso de instalación pero en esta ocasión solo elija la opción de limpiar la base de datos (Reset Mote Database) para que reinstale la parte de la base de datos.

Si el problema no fue con el usuario postgres sino con la creación de la base de datos "task" que es la que usa el MoteView, entonces debemos proceder de la siguiente manera:

1.- Verifique en el pgAdmin3 que la base de datos task no ha sido creada.

La ruta del pgAdmin3 es: C:\PostgreSQL\bin\pgAdmin3.exe

Una vez que lo ejecute debe tener una ventana como la que se muestra en la (Figura B.6).

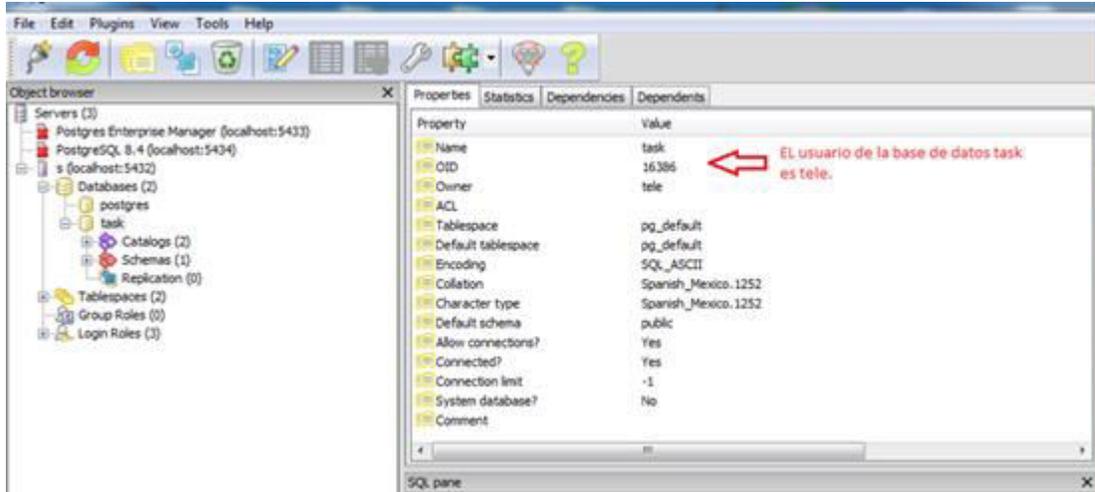


Figura B.6 Usuario de la Base de datos task - tele

Si la base de datos task no aparece puede crearla de la siguiente manera:
Solo debe dar el nombre de la base de datos “task” seleccionar el Owner “tele” y el Tablespace “pg_default” (Imagen B.7).

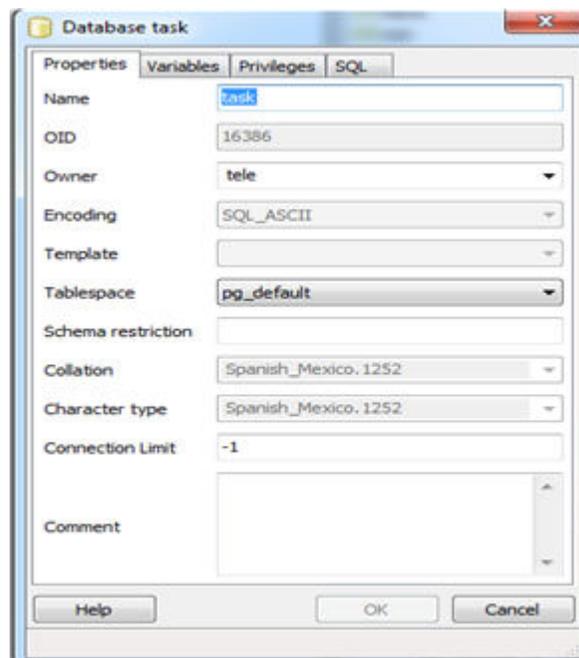


Figura B.7 Owner de la Base de datos task.



El usuario tele tuvo que haber sido creado en el proceso de instalación en la carga de PostgreSQL ODBC driver. El usuario tele puede ser localizado en la siguiente ventana: Admón. de Orígenes de datos PostgreSQL30 (Imagen B.8).



Figura B.8 Usuario PostgreSQL30 ODBC

Si por algún motivo el usuario tele no aparece este puede ser creado con el programa createuser.exe que está dentro del folder C:\PostgreSQL\bin\createuser.exe, este le pide un nombre y luego pregunta si el rol va ser super usuario le teclean la S para aceptarlo y listo se crea usuario tele con los privilegios de superusuario necesarios para la manipulación de la base de datos task.

Este usuario solo puede ser creado si se instaló correctamente el ODBC, esto permite la creación del usuario “tele”, al servidor “localhost”, en las base de datos “task”, previamente revisada y creada de ser necesario, usuario tele, el password “tiny” y por fin el port “5432” (Figura 2.9).

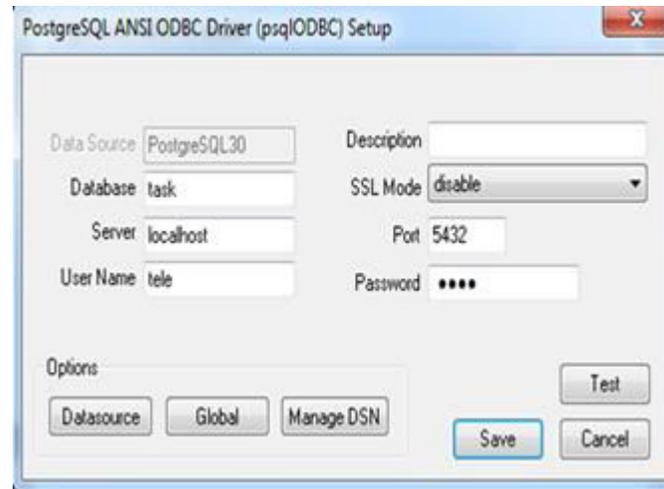


Figura B.9 Configuración del ODBC

Una vez hecho lo anterior volvemos a ejecutar la instalación con la opción de solo rehacer la BD del Mote (Reset Mote Data base).

Hasta aquí el proceso de instalación del programa y base de datos de MoteView, en futuras revisiones se modificara de ser necesario.

Anexo C

Introducción a Clarion

En general los lenguajes de programación tienen como objetivo el crear aplicaciones que al usuario le faciliten las tareas repetitivas de su trabajo o la manipulación de grandes cantidades de datos que de manera manual



simplemente resultaría imposible de lograr. Otra de las cosas en común que tienen los lenguajes entre sí, es la generalidad de su uso, esto es todos crean un programa que se ejecutara en una computadora, Tablet o teléfono inteligente, con regularmente cuatro capas de desarrollo, el piso que es la definición de los elementos que se desea almacenar, una interface de entrada o captura de estos elementos (datos), los procedimientos necesarios para convertirlos en información y por último una forma de expresar esta información que la conocemos como salida o resultados.

Partiendo de lo expresado en el párrafo anterior, se decidió utilizar como lenguaje de programación al Clarion 9.1, primero por el nivel de conocimiento que previamente se tenía del mismo, la importancia de esto radica en el tiempo necesario para alcanzar la curva de aprendizaje de un lenguaje de programación, con el nivel de conocimiento de este lenguaje evitamos esta curva de aprendizaje. Otro de los aspectos importantes era el acceso a la base de datos que de antemano sabíamos por las especificaciones del MoteView era PostgreSQL versión 8.0, con esto en mente era importante que el lenguaje manipulara dicha base de datos. Este fue otro motivo para elegir el Clarion 9.1, entre sus características está el acceso por medio del ODBC a PostgreSQL.

Con prácticamente el 50% de la aplicación resuelta esto es (definición de Datos y captura de los mismos), solo nos faltaba la generación de los procedimientos y la interface adecuada para la presentación de la información.

Claro que hubo la necesidad de pasar por el proceso de ensamble de la base de datos como los expusimos en el numeral 4 de este documento donde se explica claramente cómo logramos la conexión y manipulación de la base de datos de PostgreSQL, así como la interface en la que presentamos la información de los datos almacenados en la BD (base de datos).

Ahora es tiempo de pasar a la explicación de los procedimientos que nos permiten la manipulación de los datos y su presentación al usuario final.

A continuación mostraremos el código de programación en el Lenguaje Clarion 9.1, en los diferentes procedimientos de acceso de datos así como de su presentación visual. Cabe mencionar que aunque el código parece de uso general



son instrucciones propias del Clarion 9.1 y no pueden usarse en otro lenguaje, si es posible tomar la lógica de programación y hacer uso de ella para implementarla en un lenguaje de programación diferente a este y les permita superar la curva de aprendizaje del mismo.

Nota: Es imposible mostrar todo el código que la aplicación genera así que solo se mostrará la parte medular de los procedimientos y el resto se representará en diagrama de bloques para una mejor focalización de la explicación y así lograr un mejor entendimiento.

Generación de la aplicación por Clarion 9.1, conceptos generales.

1.- La aplicación se genera automáticamente por un generador de código en base a un diccionario de datos.

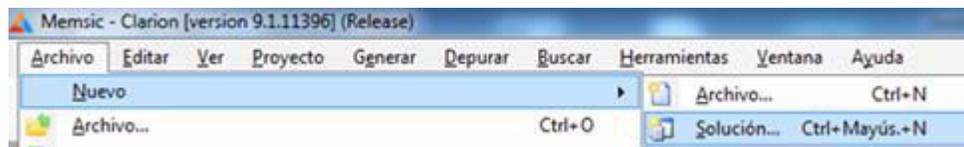


Figura C.1 Menú Archivo nuevo solución, crear una solución nueva.

Ventana de propiedades de la aplicación, Nombre físico del archivo de la aplicación, diccionario de datos con que se generará la aplicación, destino ejecutable.

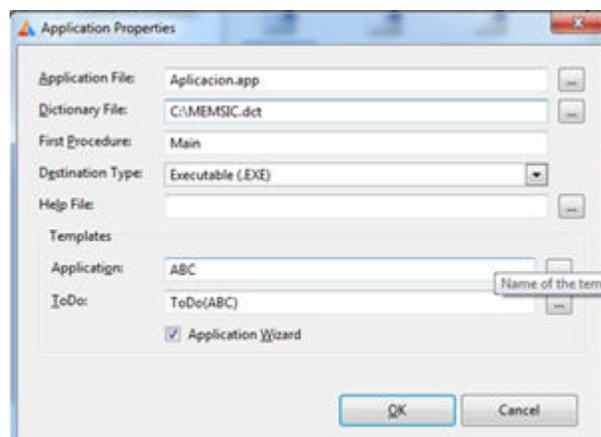




Figura C.2 Propiedades de la aplicación

Ventana que permite seleccionar que tipo de aplicación se desea crear, en este caso Application y damos los parámetros de nombre, ubicación.

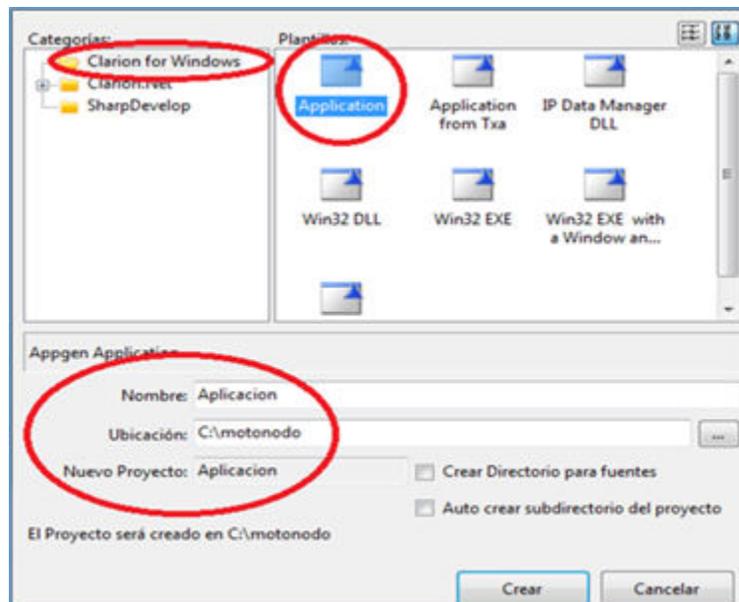


Figura C.3 Selección de tipo de aplicación a generar.

Una vez que se presiona el OK se genera automáticamente el código de la aplicación en base al diccionario de datos y muestra la siguiente pantalla.

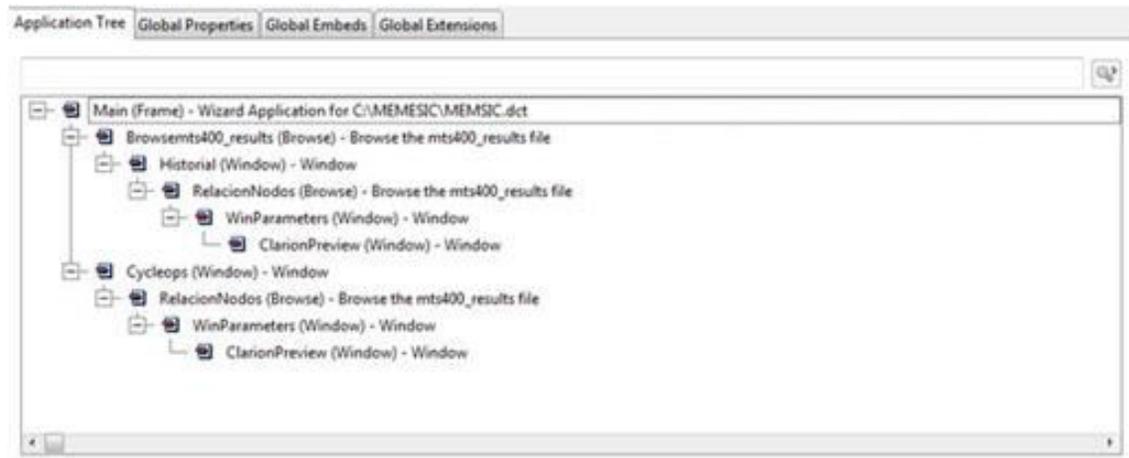


Figura C.4 Árbol de la aplicación.

La aplicación generada se muestra un árbol donde inicia con el (Main) que es el menú principal y del cual se desprenden las ramificaciones que son los diferentes procedimientos, que pueden ser relaciones, formas, procesos y reportes.

Como ya se mencionó, el Clarion tiene un generador de código, como las líneas del procedimiento Main son más de mil solo mostramos una parte pequeña de este código, cabe mencionar que no se ha tecleado ni una sola línea, la mayoría del código es generado automáticamente.

Main PROCEDURE

! Start of "Data for the procedure"

! [Priority 1300]

! [Priority 4000]

```
SQLOpenWindow      WINDOW('Initializing Database'),AT(,208,26),FONT('Microsoft Sans Serif',8,,FONT:regular),CENTER,GRAY,DOUBLE
                    STRING('This process could take several seconds. '),AT(27,12)
                    IMAGE(Icon:Connect),AT(4,4,23,17)
                    STRING('Please wait while the program connects to the database. '),AT(27,3)
                    END
```

! Window Structure

```
AppFrame           APPLICATION('Aplicación Motes'),AT(,523,318),FONT('Microsoft Sans Serif',8,,FONT:regular, |
CHARSET:DEFAULT),RESIZE,CENTER,ICON('WAFRAME.ICO'),MAX,STATUS(-1,80,120,45),SYSTEM,IMM
                    MENUBAR,USE(?Menubar)
                    MENU('&Archvo'),USE(?FileMenu)
                    ITEM('&Configura Impresora ...'),USE(?PrintSetup),MSG('Setup printer'),STD(STD:PrintSetup)
                    ITEM,USE(?SEPARATOR1),SEPARATOR
                    ITEM('E&xit'),USE(?Salir),MSG('Exit this application'),STD(STD:Close)
                    END
                    MENU('&Edita'),USE(?EditMenu)
                    ITEM('Cu&t'),USE(?Cortar),MSG('Cut Selection To Clipboard'),STD(STD:Cut)
                    ITEM('&Copy'),USE(?Copiar),MSG('Copy Selection To Clipboard'),STD(STD:Copy)
                    ITEM('&Paste'),USE(?Paegar),MSG('Paste From Clipboard'),STD(STD:Paste)
```



```
END
MENU('&Relaciones'),USE(?BrowseMenu)
    ITEM('Historico de Mediciones'),USE(?Browsemts400_results),MSG('Browse mts400_results')
    ITEM('Mediciones En Linea'),USE(?ITEM1)
END
MENU('&Ventanas'),USE(?WindowMenu),STD(STD:WindowList)
    ITEM('M&osaico'),USE(?Tile),MSG('Arrange multiple opened windows'),STD(STD:TileWindow)
    ITEM('&Cascada'),USE(?Cascade),MSG('Arrange multiple opened windows'),STD(STD:CascadeWindow)
    ITEM('&Agrupar Iconos'),USE(?Arrange),MSG('Arrange the icons for minimized windows'),STD(STD:Arrangelcons)
END
MENU('Ay&uda'),USE(?HelpMenu)
    ITEM('&Contenido'),USE(?Helpindex),MSG('View the contents of the help file'),STD(STD:Helpindex)
    ITEM('&Buscar Ayuda en...'),USE(?HelpSearch),MSG('Search for help on a subject'),STD(STD:HelpSearch)
    ITEM('&Como usar la Ayuda'),USE(?HelpOnHelp),MSG('Provides general instructions on using help'), |
STD(STD:HelpOnHelp)
    END
END
TOOLBAR,AT(0,0,523,25),USE(?TOOLBAR1)
BUTTON('En Linea'),AT(1,1,64,23),USE(?BUTTON1),LEFT,ICON(ICON:Tick)
END
END
```

Figura C.5 Parte del código del procedimiento principal MAIN.

El concepto general de la programación de los procedimientos del Clarion 9.1, es un Lenguaje ARAD (Advanced Rapid Application Development) además de ser un **entorno de desarrollo integrado** orientado a la programación de aplicaciones de bases de datos.

El núcleo del editor de Clarion está formado por un Editor del Diccionario de Datos (en donde se almacenan descripciones de tablas, atributos por defecto para las vistas de las columnas en las ventanas y reportes, reglas de negocios y opciones de uso), y el Generador de Aplicaciones Clarion (AppGen).

El generador de aplicaciones junto con una serie de **plantillas** predefinidas y personalizables y las Clases ABC (Application Builder Class), trabajan para producir código **POO**(Programación Orientada a Objetos) pre-testeado. El programador puede añadir su propio código que se mezcla con el de las plantillas, las cuales pueden generar su código una y otra vez sin perder el código introducido por el programador. Las plantillas pueden generar código a nivel puntual, relacionado con un control, o globales para el programa o sistema. Con cambios a las plantillas globales se pueden obtener cambios de estética o de funcionamiento globales, sin perder las funciones programadas en forma particular.



El sistema de manejo de datos es del tipo relacional, en el cual un conjunto de una o más tablas se vinculan entre sí con un campo en común, en las dos tablas posee las mismas características como el nombre, tipo y la longitud a este campo se le denomina comúnmente ID o campo llave, tomando este modelo Clarion genera los objetos Browse (Maestro), Update (Detalle) para cada una de las tablas del diccionario de datos.

El ejemplo de estas relaciones (Browses) lo podemos ver en la Figura 2.5 que muestra cada una de estas asociadas a una tabla y genera el código necesario para que los registros de la tabla puedan ser mostrados en la pantalla en una forma ordenada y legible para el usuario.

Se puede observar que no tenemos ningún procedimiento Update, esto se debe a que el sistema no está generando datos vía la captura del usuario, por el contrario los datos que ingresan a la base de datos son automáticamente almacenados por el Xserve mencionado en la Figura 3.6, por tanto esta aplicación es de lectura o extracción de datos almacenados.

A continuación se muestra un ejemplo de la instrucción que filtra los registros dentro de un Browse (Maestro).

Haciendo uso de la Tabla mts400_results, es posible filtrar el nodo que se desea consultar se muestra el código utilizado para visualizar sólo los datos que se requieren, el objeto BRW1 representa la tabla ya mencionada a la cual se le aplica un filtro en la columna mts:nodeid que es el identificador o campo donde están almacenados los valores de los nodos de nuestra red de sensores.

```
BRW1.SETFILTER('(mts:nodeid = GLO:NODO)')  
SELECT(?Browse:1)  
Display
```

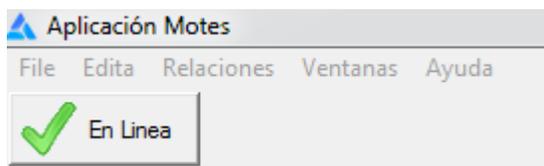


Figura C.6 Código embebido dentro de la aplicación.

En la figura anterior se muestra el código embebido dentro del procedimiento aplicar filtro de la clase BRW1 con uso de la variable global GLO:NODO que contiene el valor del nodo que se selecciona previamente permite realizar el filtro y mostrar solo los datos de este sensor.

Las figura C.6 muestran el resultado de la aplicación de este filtro, dando como resultado el despliegue de los datos que representan al nodo previamente seleccionado que son 7814 y 7818 respectivamente.

DATOS EN LINEA



Se muestra la operación en línea, al presionar este botón se activa el Xserve con la siguiente instrucción dentro del Clarion, utilizando la plantilla PowerRUN con el comando:

```
GBWinCommand='xserve -s=COM10 -b=57600 -l -xmlfile=XmlStream.xml -xmlp -xmlport=9005'
```

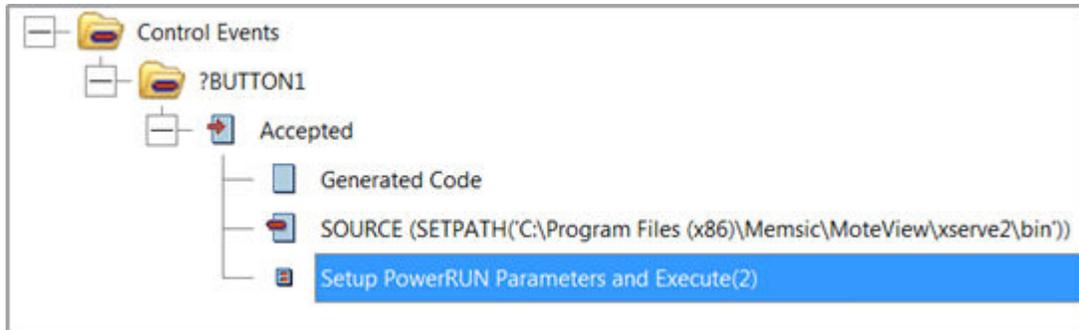


Figura C.7 Llamada a la plantilla Power Run como evento de control al hacer clic en botón en línea

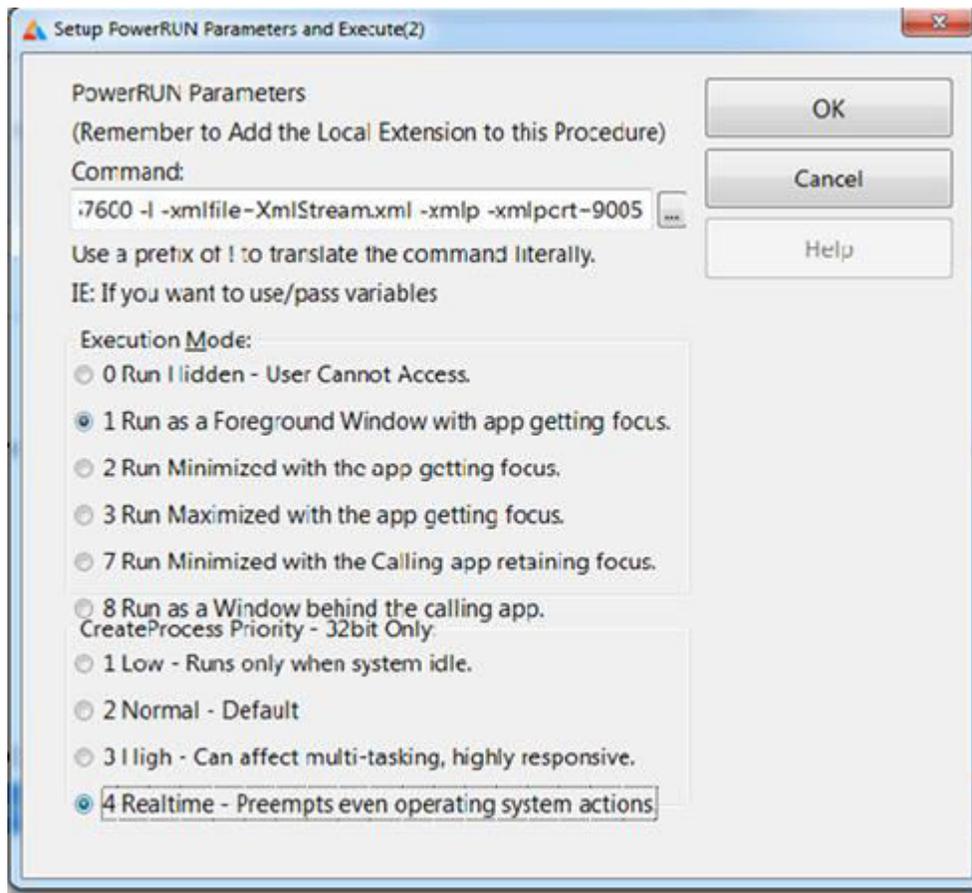


Figura C.8 Parámetros de la plantilla Power Run

Una vez generado el código de la plantilla se integra al código del programa y permite la ejecución del Xserve en la parte de atrás de la aplicación y en tiempo real, y así lograr el acceso a la más reciente lectura de cualquiera de los nodos de nuestra red de sensores.

Es hora de identificar el código que permite generar dinámicamente los nodos en la pantalla del Figura, C.9.



```
SELF.Reset
Encontrado = 0
LOOP X# = 2 to Records(Nodos)
    GET(Nodos, X#)
    Hora" = format(clock(),@t1)
    IF Sub(Hora",4,2) - SUB(Nodos.HoraReg,4,2) > 2
        Do btnhide
        DELETE(Nodos)
    END
END

IF BAN = 0
    IF GLO:Registros < RECORDS(mts400_results)
        GLO:Registros = RECORDS(mts400_results)
        mts:conseid = 99999999
        SET(mts:Conse,mts:Conse)
        PREVIOUS(mts400_results)
        Nodos.NodoNum = mts:nodeid
        SORT(Nodos,Nodos.NodoNum)
        GET(Nodos,Nodos.NodoNum)
        IF ERRORCODE()
            Nodos.nodonum = 1
            Nodos.HoraReg = format(clock(),@t1)
            Nodos.FechaReg = TODAY()
            Nodos.Horaproc = format(clock(),@t1)
            Nodos.FechaProc = TODAY()
            Nodos.ejeX = POINTER(Nodos) + RANDOM(1,655)
            Nodos.ejeY = POINTER(Nodos) + RANDOM(1,355)
            Nodos.Descripcion = 'GATEWAY'
            unhide(?BUTTON1)
            ?BUTTON1{PROP:text} = 'GW'
            ?BUTTON1{PROP:FontSize}= 13
            ?BUTTON1{FONT:bold} = 700
            ?BUTTON1{PROP:Color} = 0000FF00h
            SETPOSITION(?BUTTON1,Nodos.ejeX,Nodos.ejeY,26,20)
            GLO:XGW = Nodos.ejeX
            GLO:YGW = Nodos.ejeY
            Nodos.botond = 1
            ADD(Nodos,Nodos.nodonum)
            contador = 2
        END
    END
    BAN = 1
END

IF GLO:Registros < RECORDS(mts400_results)
    GLO:Registros = RECORDS(mts400_results)
    mts:conseid = 99999999
    SET(mts:Conse,mts:Conse)
    PREVIOUS(mts400_results)
    Nodos.NodoNum = mts:nodeid
    SORT(Nodos,Nodos.NodoNum)
    Nodos.NodoNum = mts:nodeid
    GET(Nodos,Nodos.NodoNum)
    IF ERRORCODE()
        Nodos.nodonum = mts:nodeid
        Nodos.Descripcion = 'Nodo'
        Nodos.HoraReg = format(clock(),@t1)
        Nodos.FechaReg = TODAY()
        Nodos.Horaproc = format(clock(),@t1)
        Nodos.FechaProc = TODAY()
        Nodos.ejeX = POINTER(Nodos) + RANDOM(1,655)
```



```
Nodos.ejey = POINTER(Nodos) + RANDOM(1,355)
Nodos.botond = contador
ADD(Nodos,Nodos.nodonom)
contador = contador + 1
ELSE
Nodos.Horaproc = format(clock(),@t1)
Nodos.HoraReg = format(clock(),@t1)
Nodos.FechaProc = TODAY()
PUT(Nodos)
END
END
DO ciclo
```

Figura C.9 Código de recolección de datos de los nodos

Se muestran los parámetros de configuración de la ventana donde se representan los nodos de sensores en tiempo real.

Se inicia con el parámetro de temporizador (Timer) de la ventana sensores en línea el cual permite definir el periodo en el cual se revisará de manera automática si hay un nuevo registro de alguno de los nodos de sensores de nuestra red, como lo muestra la figura C.10

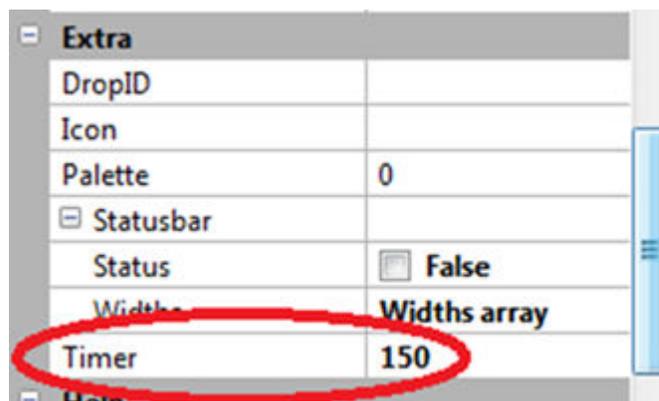


Figura C.10 Configuración del temporizador de sensores en línea

El temporizador (Timer) funciona en centésimas de segundo por lo tanto se puede variar el tiempo de refresco de la ventana aumentando o disminuyendo el temporizador con ello podemos controlar con qué frecuencia vamos a realizar las lecturas de la red de sensores. El valor que muestra la figura anterior apenas representa 1.5 segundos, así que se requiere incrementar el valor para tener un periodo más largo de revisión de datos.

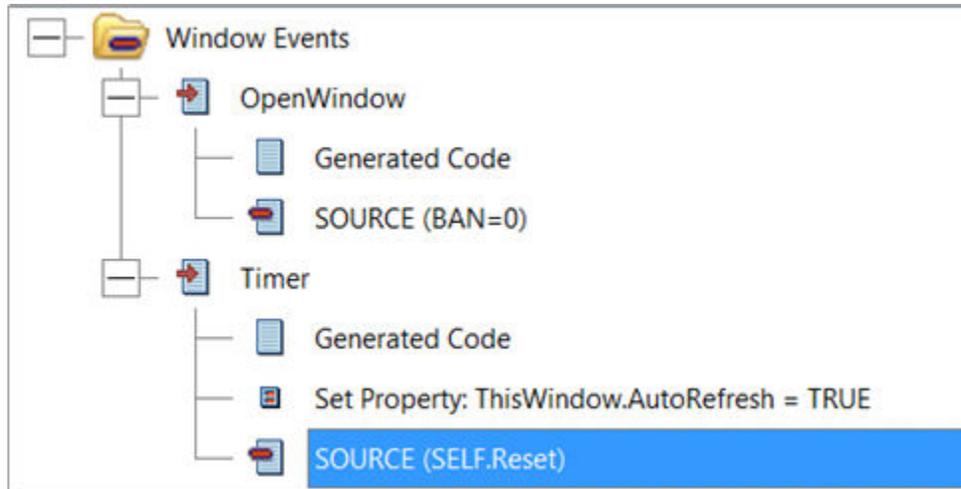


Figura C.11 Eventos de ventana ejecución de temporizador

La Figura C12 muestra el código embebido en el temporizador de la ventana donde se realiza la recolección de los datos de los nodos de sensores.

La última parte es la generación de las gráficas en base a los datos almacenados de cada nodo sensor, a continuación se muestra el código utilizado para representarlas.

```
DesdeS = "" & FORMAT(Desde,@D6) & ""
HastaS = "" & FORMAT(Hasta,@D6) & ""
!!!! imprime Recibo
clear(SelectionString)
CASE GLO:Procede
OF 1
    re = CrystalReport.init('Grafica1.rpt')
OF 2
    re = CrystalReport.init('Grafica2.rpt')
END
SelectionString = '{{mts400_results.nodeid} = '&GLO:NODO&' AND
{{mts400_results.result_time} >= CDate('&DesdeS')} and
{{mts400_results.result_time} <= CDate('&HastaS')}
re = CrystalReport.SelectionFormula(SelectionString)
UseRegion=1
CrystalReport.HasNavigationControls(1)
CrystalReport.ShowReportControls(1)
CrystalReport.HasPrintButton(1)
CrystalReport.HasExportButton(0)
CrystalReport.HasCloseButton(0)
ClarionPreview(CrystalReport,UseRegion)
```

Figura C.12 Código embebido en el botón OK del Procedimiento Win Parameters.



Curriculum Vitae

Grados Obtenidos:

Licenciatura en Sistemas de Computación Administrativa – Universidad Autónoma de Chihuahua (1992).

Experiencia laboral:

JEFE DE BIBLIOTECA Y PROFESOR
FACULTAD DE CIENCIAS POLÍTICAS Y SOCIALES UACH **CHIHUAHUA,**
CHIH.

Marzo de 2016 a Octubre 2017

Administración y servicio de biblioteca y Profesor de la materia de lenguaje y comunicación.

PROGRAMADOR ANALISTA
SEGURIDAD PÚBLICA MUNICIPAL **CHIHUAHUA,**
CHIH.

Octubre 2012 a Marzo 2016

Soporte a usuarios, reparación de equipo de cómputo, configuración de software, configuración, instalación y programación de radios portátiles y móviles Motorola.

ING. DE SERVICIO
JUÁREZ COMPUTACIÓN **CD. JUÁREZ,**
CHIH.

Noviembre 2010 - Septiembre 2011

Servicio al Cliente, Configuración de redes, reparación de equipo de cómputo, configuración de software, Programación en Vb6, C#, MS-SQL, desarrollo de páginas web, configuración de servidores Windows 2003,2008.

COORDINADOR DE SISTEMAS
DAVOLSURGICAL INNOVATIONS **CD. JUÁREZ,**
CHIH.

Agosto 2002 – Marzo 2010

Actividades: Administración de servidores Windows 2003, PBX Definity. Configuración de estaciones de trabajo, Soporte a Usuarios, Cableado estructurado, Mantenimiento y reparación de PC.

Davol Inc, BARD subsidiaria en Warwick Rhode Island: Configuración de estaciones de trabajo, MFG/PRO, administración de servidores Win 2000, NT, Migración de Lotus CC-mail a Microsoft Exchange 2000, Desarrollo de la Intranet, Operaciones de Respaldo, Configuración de Usuarios de Citrix. Software: Desarrollo del sistema de Control de Etiquetas de producción, Control de tickets de Desperdicio. Administración: Un Técnico a mi cargo. Dirigí un grupo multidisciplinario de ingenieros para mejorar el sistema de visión y etiquetado en producción. Otro proyecto en el que participe en coordinación con Nogales, Reynosa y Warwick fue estandarizar el uso del sistema Tress para el pago de la nómina en las plantas mexicanas de Bard, centralizando el software en la planta de at Reynosa.

Domicilio Permanente: Jesús González Ortega # 5300 Granjas

Chihuahua, Chihuahua, 31100

Esta tesis fue mecanografiada por Gabriel Hernández Salcedo.