

Red Inalámbrica de Sensores. Detección de Plazas Libres en Estacionamiento Vehicular

Jorge Ojeda Ramos, Jorge Arrúa, Katia Ayala

jorgeojeda90@gmail.com, jorgearrua@gmail.com, ktiaayala@gmail.com

Facultad Politécnica - Universidad Nacional del Este
Ciudad del Este – Paraguay

Resumen. En este proyecto, se aborda un trabajo de investigación en el campo de la Automatización y Control mediante la implementación de una red inalámbrica de sensores (Wireless Sensor Network, WSN), basada en el estándar IEEE 802.15.4 ZigBee y en la plataforma Arduino, y el diseño de una interfaz gráfica que permita la monitorización de las medidas recogidas por los nodos sensores en tiempo real. Se propone un sistema de detección de estado de las plazas de un local de estacionamiento vehicular. Para tal efecto, se dispone de una red que cuenta con dos nodos sensores y un nodo coordinador encargado de monitorear el estado de las plazas de estacionamiento. Cada nodo sensor cuenta con un sensor ultrasónico que recoge la distancia del objeto más cercano (vehículo), la analiza y envía una respuesta con el estado de la plaza al nodo coordinador. El coordinador se conecta a una PC mediante un gateway UART/USB, recoge la información recibida por radiofrecuencia de los diferentes nodos y la representa en la interfaz gráfica diseñada (desarrollada en NetBeans).

Palabras claves: Automatización y Control, WSN, IEEE 802.15.4 ZigBee, Arduino, Estacionamiento Vehicular, Ultrasonido.

Abstract. This work addresses a research in the field of Automation and Control by implementing a wireless sensor network (WSN) based on the IEEE 802.15.4 ZigBee and the Arduino platform, and the design of a graphical interface that allows the monitoring of the measures collected by the sensor nodes in real time. It proposes a system for detecting the state of parking spaces. To this end, there is a network that has two sensor nodes and a coordinator node to monitor the status of the parking spaces. Each sensor node has an ultrasonic sensor that collects the distance of closest object (vehicle), analyzes it and sends a response to the coordinator node. The coordinator connects to a PC via a gateway UART/USB, collects information received by radio from different nodes and represents it in the graphical interface designed (developed in NetBeans).

Keywords: Automation and Control, WSN, IEEE 802.15.4 ZigBee, Arduino, Vehicle Parking, Ultrasound.

1. Introducción

Hoy, encontrar una plaza libre en un estacionamiento se ha convertido en una tarea poco trivial, si no en un indiscutible desafío [1]. A medida que los valiosos minutos se escapan, los conductores dan vueltas por el estacionamiento, cada vez más frustrados ante la falta de espacios para estacionar. En el mejor de los casos, estas búsquedas son un inconveniente; en el peor, pueden causar una llegada tardía a una cita crítica o, incluso, la pérdida completa de la misma. En una situación como ésta, es necesario que alguien (o algo) sirva de guía a un espacio libre.

De ahí surge la idea de mejorar la gestión del espacio físico destinado al estacionamiento vehicular de las grandes empresas, industrias y establecimientos de cualquier índole, facilitando el acceso a las vacancias de los mismos a través del uso de tecnología eficiente y confiable.

Por otra parte, la continua miniaturización de los dispositivos, el aumento de la capacidad de computación y los avances en las técnicas de reducción de consumo de energía permiten la aparición de nuevos modelos de recopilación de información, como por ejemplo las redes inalámbricas de sensores (Wireless Sensor Networks, WSN) [2].

Las redes inalámbricas de sensores están actualmente en el punto de mira de muchos investigadores y empresas

tecnológicas. Su principal objetivo es la adquisición y el tratamiento de datos de forma rápida, flexible y autónoma con múltiples aplicaciones en distintos campos [3].

Se ha decidido introducirse en el mundo de la monitorización inalámbrica debido a la creciente importancia de las redes de sensores en las necesidades actuales, lo cual convierte esta tecnología en un interesante campo de desarrollo para la ingeniería de sistemas.

La detección automática del estado de plazas es algo bastante novedoso en el Paraguay ya que pocos establecimientos disponen de soluciones tecnológicas para resolver el gran problema que supone encontrar plazas libres en locales de estacionamiento [7].

En este trabajo, se propone un sistema de detección de plazas libres en estacionamiento vehicular mediante la implementación de una red inalámbrica de sensores basado en la plataforma Arduino y el estándar de comunicaciones ZigBee.

A continuación, se presenta la organización del paper y se describe brevemente el contenido de las secciones siguientes. En la sección 2, se especifican antecedentes de trabajos publicados que tienen directa relación con esta área de estudio. En la sección 3, se introduce la generalidad de las redes inalámbricas de sensores y las tecnologías específicas utilizadas en el presente trabajo: ZigBee y la plataforma Arduino. En la sección 4, se presenta el sistema a ser implementado, describiendo sus

elementos, su funcionamiento y configuración. En la sección 5, se exponen la implementación y pruebas del sistema, así como los resultados obtenidos. En la sección 6, se exhibe la discusión de los resultados obtenidos en cada una de las pruebas realizadas. Finalmente, en la sección 7, se mencionan las conclusiones del trabajo, recomendaciones y las líneas de futuros trabajos posibles.

2. Trabajos Previos

2.1. Redes Inalámbricas de Sensores Inteligentes. Aplicación a la Monitorización de Variables Fisiológicas.

En este trabajo, se propone un mecanismo basado en redes de sensores inteligentes para facilitar la monitorización de parámetros en humanos que evite su hospitalización y que les permita, en la medida de lo posible, seguir con su actividad cotidiana [2].

2.2. Estudio de una red de sensores sin hilos basadas en la tecnología *Arduino* bajo protocolos de comunicaciones *ZigBee*.

Este trabajo se enmarca en el proyecto AVANZA “Creación de un Sistema de Información Acústica para las playas de Ibiza y Formentera”. El propósito del mismo es diseñar una red cuya alimentación sea con baterías conectadas a paneles solares. La información de la red se trata remotamente en un ordenador que procesa y visualiza la información para, si es necesario, tomar las acciones correspondientes para reducir el ruido [3].

3. Conceptos Fundamentales

3.1. Redes Inalámbricas de Sensores (Wireless Sensor Networks, WSN)

Una red inalámbrica de sensores está conformada por un conjunto de elementos autónomos (nodos) interconectados de manera inalámbrica, que miden variables de ambiente [4].

Las redes de sensores tienen las siguientes tareas típicas [5]:

- Determinar un parámetro ambiental: calor, presión, luz, radiación, presencia de humo, humedad, ruido, fricción.
- Detectar eventos: presencia, llegada, movimiento, vibración, flujo.
- Estimar parámetros: velocidad, dirección.
- Clasificar los objetos detectados.
- Seguir la trayectoria de un objeto detectado.

La infraestructura de una red inalámbrica de sensores que puede ser apreciada en el Figura 1 comprende los siguientes componentes básicos [6]:

- Un conjunto de nodos sensores.
- Una red de interconexión inalámbrica.
- Un punto central de recolección de información o estación base.

- Un conjunto de recursos para procesar la información recolectada.

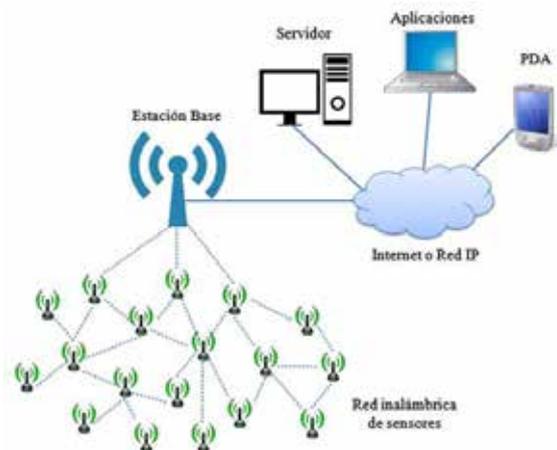


Figura 1. Infraestructura de una red inalámbrica de sensores

3.2. ZigBee

ZigBee es un estándar de comunicación basado en la norma IEEE 802.15.4 para redes inalámbricas, orientado a aplicaciones cuyos requerimientos principales son bajas tasas de transmisión, bajo coste y larga autonomía [8]. Algunos sistemas, para los que existen perfiles de aplicación, están definidos en la Figura 2.

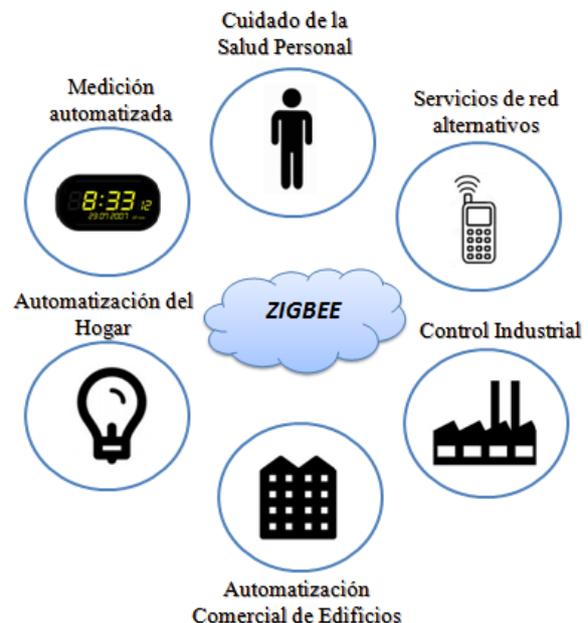


Figura 2. Perfiles ZigBee

Entre las necesidades que satisface el estándar ZigBee se encuentran [8]:

- Bajo coste.
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.

- Redes flexibles y extensibles.

Los roles de los nodos en ZigBee se corresponden con los del estándar IEEE 802.15.4, aunque tienen diferente denominación [8]:

- Coordinador ZigBee (coordinator) (Coordinador PAN IEEE 802.15.4)
- Encaminador ZigBee (router) (Coordinador IEEE 802.15.4)
- Dispositivo Final ZigBee (end points) (Dispositivo IEEE 802.15.4)

Se soportan las topologías de red estrella o punto a punto, como se especifica en IEEE 802.15.4 (Figura 3). La red punto a punto puede tomar diferentes formas: malla (*mesh*) y árbol.

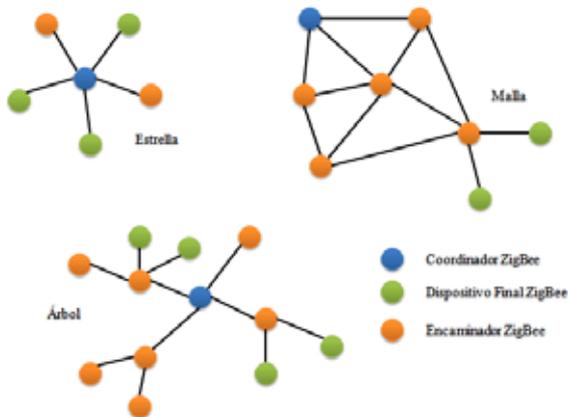


Figura 3. Topologías en ZigBee

Los módulos XBee (Figura 4) son dispositivos que integran un transmisor-receptor de ZigBee y un procesador en un mismo módulo, lo que le permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla [9]. Son económicos, potentes y fáciles de utilizar. Algunas de sus principales características son [6]:

- Buen alcance: hasta 100 metros en línea vista para los módulos XBee y hasta 1,6 Km para los módulos XBee PRO.
- 9 pines entrada/salida con entradas analógicas y digitales
- Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo *sleep*.
- Interfaz serial.
- Más de 65000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- Fáciles de integrar.



Figura 4. Módulo XBee

3.3. Arduino

Arduino es una plataforma *open-hardware* de diseño y distribución libre, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que implementa el lenguaje *Processing/Wiring* [12].

Arduino ofrece algunas ventajas sobre otros sistemas [10]:

- Barato.
- Multiplataforma.
- Entorno de programación simple y claro.
- Código abierto y software extensible.
- Código abierto y hardware extensible.

En la Figura 5, se puede observar algunas de las placas Arduino. Tal variedad responde a las diferentes extensiones que se han venido realizando de las primeras placas de acuerdo con las demandas específicas de los usuarios y son también resultado natural de tratarse de una tecnología abierta. El uso de una u otra placa dependerá del tamaño del proyecto a realizar, de las prestaciones (básicamente potencia, comunicación Bluetooth, otras), de la programación (USB o externo), o interacción con otros dispositivos [3].

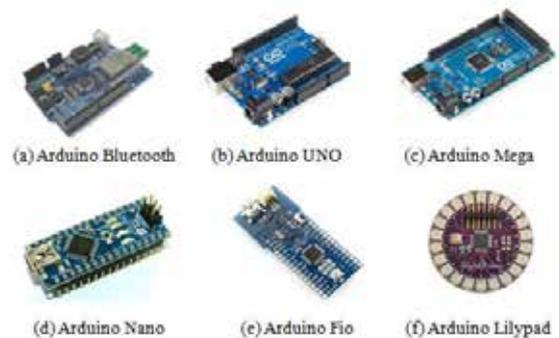


Figura 5. Gama de Arduino

4. Descripción del sistema propuesto

En esta sección, se da una visión general del diseño del sistema implementado en este proyecto.

4.1. Bloques y elementos del sistema

La Figura 6 muestra los diferentes bloques que forman la red inalámbrica de sensores propuesta y las conexiones entre ellos, existiendo una conexión inalámbrica bidireccional entre cada nodo y el coordinador y una conexión física cableada entre el coordinador y la PC.

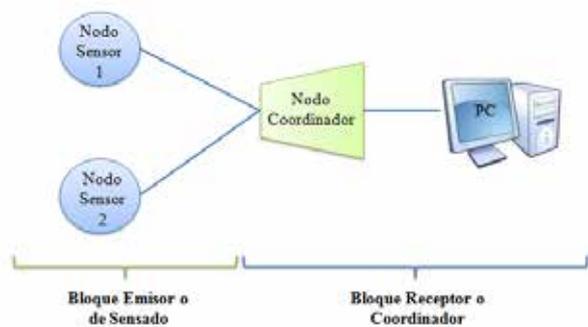


Figura 6. Esquema del sistema planteado

El sistema está dividido en dos bloques:

- Bloque Emisor o de Sensado. Cuenta con dos nodos sensores (*end devices*) encargados de transmitir y recibir datos. Cada dispositivo cuenta con un Kit Arduino UNO, un módulo de comunicación por RF XBee y un sensor ultrasónico que recoge información del medio (Figura 7).



Figura 7. Bloque Emisor - Nodo Sensor

- Bloque Receptor o Coordinador. Cuenta con un dispositivo coordinador (Figura 8) que recibe los datos del bloque emisor y los gestiona con la PC. También manda información a cualquiera de los nodos mediante el módulo de comunicación XBee que posee. La PC a la que va unido el equipo coordinador se encarga de monitorizar los datos mediante un entorno gráfico que será diseñado en NetBeans 7.3.1.



Figura 8. Bloque Receptor - Coordinador

4.2. Funcionamiento del Sistema

Cada nodo sensor se encuentra en una plaza de estacionamiento. Cuando el nodo coordinador lo solicita, los nodos sensores le envían los datos con el estado de las plazas de estacionamiento teniendo en cuenta la distancia del obstáculo más cercano (vehículo).

Los nodos sensores cuentan con un sensor ultrasónico, que mide la distancia del objeto más cercano. De acuerdo con el lugar de la aplicación del sistema y de la ubicación de los módulos, si se trata de un local de estacionamiento abierto o cerrado, se establecerá un rango para determinar el estado de cada plaza, es decir, se determinará si la plaza se encuentra libre u ocupada comparando la distancia

obtenida con la consulta con un valor específico preestablecido (Figura 9).

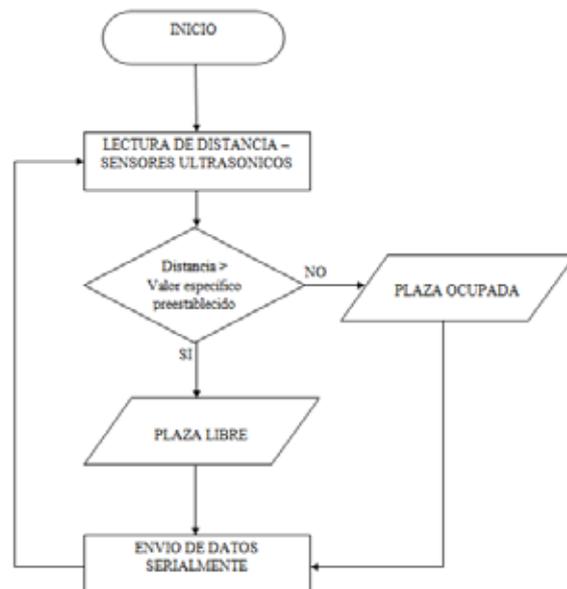


Figura 9. Diagrama de Flujo Simplificado de Bloque Emisor.

El microprocesador del Arduino emisor le envía mediante una comunicación serie al módulo XBee los datos que se desean transmitir al coordinador por radiofrecuencia, y éste cuando los recibe, se los pasa a la PC por el puerto USB, para que sean monitorizados por pantalla.

Una vez recibidos los datos en la PC, éstos son mostrados por pantalla mediante una interfaz gráfica que se ha creado para que resulte más fácil visualizar el resultado (Figura 10).

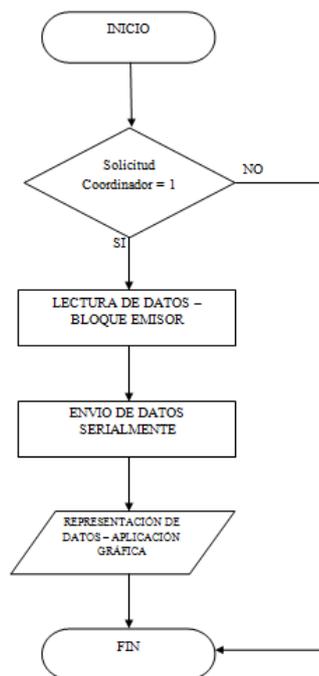


Figura 10. Diagrama de Flujo Simplificado de Bloque Receptor.

4.3. Configuración del sistema

En este proyecto, se ha optado por una conexión punto a multipunto. Los dispositivos terminales de la red son los módulos Arduino-XBee emisores, y el coordinador es el módulo USB-XBee receptor que se encuentra unido a la PC y es el encargado de transmitir toda la información que le llegue de los distintos módulos.

Para configurar una red punto a multipunto lo primero que se hizo fue configurar cada módulo XBee con su respectiva etiqueta, como coordinador o como *end device*. Para ello se utilizó el programa X-CTU [11], y dependiendo de la misión de cada dispositivo, los valores de los parámetros de configuración han sido diferentes, como se puede ver en la Tabla 1.

	COORDINADOR	ENDDEVICE 1	ENDDEVICE 2
PANID	3332	3332	3332
Destination Address(DH)(DL)	0/FFFF	0	0
Source Address(MY)	0	1	2
Coordinator Enable(CE)	1	0	0

Tabla 1. Parámetros de configuración de los módulos XBee

Una vez configurados los módulos XBee, el siguiente paso fue realizar el código fuente usando el software de Arduino y cargarlo al microprocesador de cada kit Arduino mediante USB.

Seguidamente, se procedió a crear la interfaz gráfica donde se analizan todos los datos recogidos por el sistema. Esta interfaz gráfica se realizó con el software NetBeans 7.3.1 [13] y permite monitorizar los resultados de medida de una forma sencilla y rápida (Figura 11).

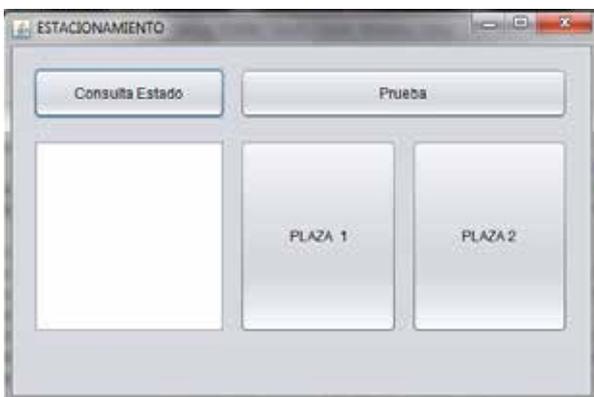


Figura 11. Aplicación Gráfica

5. Experimentos y resultados

En este apartado, se llega al diseño final de la red inalámbrica de sensores y se realizan las pruebas utilizando la interfaz gráfica diseñada en NetBeans.

En la Figura 12, se puede observar el diagrama final de bloques de la red desarrollada en este trabajo.

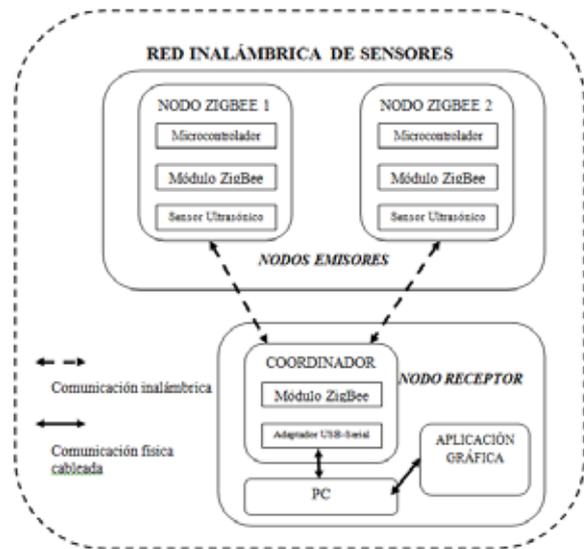


Figura 12. Diagrama final de bloques de los componentes de la red inalámbrica.

A continuación, se puede ver, en la Figura 13, el montaje realizado de la red sensorial. Aquí aparecen los dos nodos sensores, el coordinador y sus circuitos de acondicionamiento.

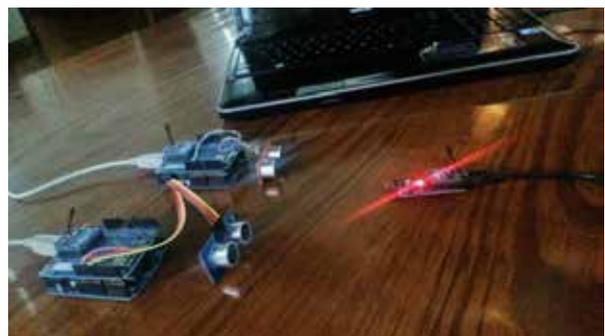


Figura 13. Montaje final realizado

La mayoría de las pruebas fueron realizadas en el escenario que se aprecia en la Figura 14. El mismo consiste en un local de estacionamiento abierto, con dos plazas de estacionamiento disponibles.



Figura 14. Escenario en donde se realizaron las pruebas del sistema.

Seguidamente, se muestran las diferentes pruebas realizadas y los resultados obtenidos en cada una de ellas.

5.1. Prueba 1: Diferentes combinaciones de estado de plazas en escenario de pruebas.

En esta prueba, se analizó el comportamiento del sistema con diferentes combinaciones de estado de plazas en el escenario de pruebas. Un ejemplo se puede observar en las Figuras 15 y 16.



Figura 15. Escenario Plaza 1 ocupada y Plaza 2 Libre



Figura 16. Resultado obtenido con el sistema

En la Tabla 2, se pueden apreciar los resultados obtenidos, en donde se detalla el estado real de la plaza 1 (P1), el estado real de la plaza 2 (P2) y el resultado obtenido a través de la aplicación gráfica (R).

#	P1	P2	R
1	Libre	Libre	Plaza 1 Libre Plaza 2 Libre
2	Ocupada	Libre	Plaza 1 Ocupada Plaza 2 Libre
3	Libre	Ocupada	Plaza 1 Libre Plaza 2 Ocupada
4	Ocupada	Ocupada	Plaza 1 Ocupada Plaza 2 Ocupada

Tabla 2. Prueba 1

5.2. Prueba 2: Distancia Vehículo – Nodo Sensor

En esta prueba, se analizó el comportamiento del sistema ubicando los obstáculos (vehículos) a distintas distancias de los nodos sensores, en busca de posibles resultados incorrectos por probables problemas de interferencia.

Cabe destacar que de acuerdo a lo mencionado en cuanto al funcionamiento del sistema, para decidir si una plaza de estacionamiento se encuentra libre o no, se tiene en cuenta la distancia del objeto más cercano capturada por el nodo sensor. Si esta medida se encuentra dentro de los límites establecidos para cada plaza, el sistema establece que la plaza se encuentra ocupada. Caso contrario, si la medida

supera los límites, el sistema establece que la plaza se encuentra libre.

Los resultados obtenidos en las diferentes pruebas se pueden observar en la Tabla 3, en donde se especifica la distancia del vehículo al nodo sensor de la plaza 1 (D1), la distancia del vehículo al nodo sensor de la plaza 2 (D2), el estado real de las plazas (ER) y el estado que arroja el sistema (ES).

#	D1	D2	ER	ES
1	10 cm	-	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
2	15 cm	-	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
3	30 cm	-	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
4	-	10 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada
5	-	15 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada
6	-	30 cm	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
7	30 cm	30 cm	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
8	10 cm	30 cm	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
9	30 cm	15 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada
10	-	-	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre

Tabla 3. Prueba 2

5.3. Prueba 3: Cambio de posición de nodos sensores

En esta prueba, se analizó el comportamiento del sistema mudando de lugar los nodos sensores y colocándolos en el centro superior de cada plaza de estacionamiento (Figura 17).



Figura 17. Ejemplo del escenario de la prueba 3

Para realizar esta prueba, se tuvo que modificar el *firmware* de los nodos sensores en el siguiente aspecto: si la altura medida por el sensor ultrasónico no varía y se mantiene constante, la plaza se encuentra libre. Ante cualquier variación de esa medida, el sistema determinará que la plaza se encuentra ocupada.

Los resultados obtenidos se pueden observar en la Tabla 4, en donde se especifica la distancia del vehículo a la pared de la plaza 1 (D1), la distancia del vehículo a la pared de la plaza 2 (D2), el estado real de las plazas (ER) y el estado que arroja el sistema (ES).

#	D1	D2	ER	ES
1	10 cm	-	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
2	15 cm	-	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Ocupada
3	30 cm	-	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
4	-	10 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada
5	-	15 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Ocupada Plaza 2 Ocupada
6	-	30 cm	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
7	30 cm	30 cm	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
8	10 cm	30 cm	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
9	30 cm	15 cm	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada
10	-	-	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre

Tabla 4. Prueba 3

5.4. Prueba 4: Diferentes disposiciones de las plazas de estacionamiento

En esta prueba, se analizó el comportamiento del sistema en caso de contar con un local de estacionamiento en donde las plazas se encuentran una frente a otra, o una detrás de otra como se puede apreciar en la Figura 18.



Figura 18. Ejemplo del escenario de la prueba 4

Para tal efecto, los nodos sensores de las plazas fueron acomodados uno frente a otro a una cierta distancia (Figura 19) y uno detrás del otro (Figura 20), respectivamente.



Figura 19. Disposición 1 de las plazas de estacionamiento

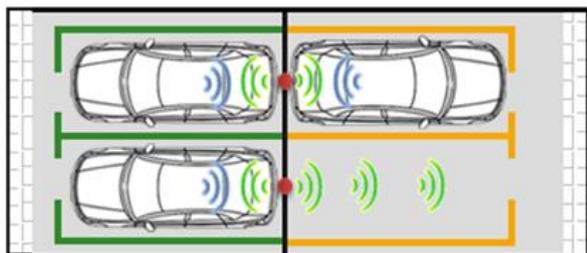


Figura 20. Disposición 2 de las plazas de estacionamiento

Los resultados obtenidos se observan en la Tabla 5, en donde se presenta la disposición de los nodos sensores (P), estado real de las plazas (ER), estado obtenido a través del sistema (ES).

#	P	ER	ES
1	Figura 19	Plaza 1 Ocupada Plaza 2 Libre	Plaza 1 Ocupada Plaza 2 Libre
2	Figura 19	Plaza 1 Libre Plaza 2 Libre	Plaza 1 Libre Plaza 2 Libre
3	Figura 20	Plaza 1 Ocupada Plaza 2 Ocupada	Plaza 1 Ocupada Plaza 2 Ocupada
4	Figura 20	Plaza 1 Libre Plaza 2 Ocupada	Plaza 1 Libre Plaza 2 Ocupada

Tabla 5. Prueba 4

5.5. Prueba 5: Alcance

En esta prueba, se analizó el alcance de la comunicación entre el nodo coordinador y los nodos sensores. Para tal efecto se alejaron los nodos sensores del nodo coordinador para verificar el rango de comunicación.

En la tabla 6, se recogen los resultados obtenidos, en donde se determina la distancia entre el nodo coordinador y los nodos sensores (D), ambiente de medición (A) y estado de la conexión (E).

#	D	A	E
1	5 metros	Indoor	Ok
2	10 metros	Indoor	Ok
3	15 metros	Indoor	Ok
4	20 metros	Indoor	No conecta
5	25 metros	Outdoor	Ok
6	30 metros	Outdoor	Ok
7	40 metros	Outdoor	Ok
8	45 metros	Outdoor	No conecta

Tabla 6. Prueba 5

Resulta importante mencionar que en un espacio real, una señal se ve afectada por otras señales y barreras que pueden presentarse en su camino. Según la hoja técnica de los módulos XBee Serie 1 teóricamente alcanzan los 30 metros en ambientes indoor (interiores) y 100 metros en ambientes outdoor (exteriores).

5.6. Prueba 6: Tiempo de vida de las baterías

Esta prueba consistió en averiguar el tiempo de duración de las baterías en los dos modos posibles: *sleep* (dormido) y continuo (sin interrupción).

Tanto el Arduino UNO como el módulo XBee pueden entrar en modo *sleep*. El consumo aproximado según los estados se establece en la Tabla 7.

#	Dispositivos	Consumo
1	Arduino UNO	23.9 mA
2	Arduino + Módulo XBee	77 mA
3	Arduino UNO dormido	9.81 mA
4	Arduino despierto + XBee dormido	26 mA
5	Arduino dormido + XBee despierto	70 mA
6	Arduino dormido + XBee dormido	21.5 mA

Tabla 7. Consumo aproximado según estados de Arduino y XBee

La Tabla 8 refleja los resultados obtenidos utilizando la combinación de la opción 2 (modo continuo) y la opción 4 (modo sleep) de la Tabla 7. En la misma, se especifican el modo de operación y la duración con baterías de Litio de 600 mAh.

Modo	Duración
Sleep	12:15:19
Continuo	07:32:27

Tabla 8. Prueba 6

6. Discusión de Resultados

6.1. Prueba 1: Diferentes combinaciones de estado de plazas en escenario de pruebas.

En todos los casos presentados en la Tabla 2, los resultados obtenidos a través del sistema coinciden con los estados reales de las plazas de estacionamiento en el escenario de pruebas.

6.2. Prueba 2: Distancia Vehículo – Nodo Sensor

En todos los casos de la Tabla 3, los resultados obtenidos concuerdan con la realidad planteada. Por lo tanto, los probables problemas de interferencia en el escenario de prueba establecido quedan prácticamente descartados.

Sin embargo, de acuerdo con las especificaciones técnicas de los sensores ultrasónicos, los mismos tienen un ángulo de medición de 30°, por lo que quizás, a la hora de implementar este sistema en el mundo real surjan algunos inconvenientes.

Estos inconvenientes pueden ser superados realizando un estudio de las dimensiones de cada plaza del estacionamiento y de esta manera ubicar los sensores en lugares convenientes.

6.3. Prueba 3: Cambio de posición de nodos sensores

Ocho de los casos presentados en la Tabla 4, arrojan resultados que concuerdan con la realidad planteada, sin embargo dos no. Esto se debe a que el sensor ultrasónico de la plaza 1, con un ángulo de medición de 30°, tiene bajo su campo de visión al automóvil estacionado en la plaza 2 y, por lo tanto, aunque en la plaza 1 no se encuentre estacionado ningún vehículo, el sistema determinará que ambas plazas se encuentran ocupadas.

Seguidamente, al comparar los resultados obtenidos en esta prueba y en la anterior, se puede concluir que la mejor disposición de los sensores para el escenario de pruebas montado es la ubicación original con un 100% de efectividad contra un 80% de la ubicación cambiada.

6.4. Prueba 4: Diferentes disposiciones de las plazas de estacionamiento

En todos los casos presentados en la Tabla 5, el estado real de las plazas de estacionamiento coincide con el estado obtenido a través del sistema. Cabe mencionar que esta prueba fue realizada manteniendo la posición original de los sensores para el escenario montado (Figura 14).

Por lo tanto, se pudo comprobar una vez más que los problemas de interferencia son prácticamente nulos, y que los estados de las plazas son correctamente captados por el sistema propuesto.

6.5. Prueba 5: Alcance

Las mediciones fueron realizadas en ambientes *indoor* y *outdoor*. El resultado obtenido no coincide con las especificaciones técnicas, ya que a partir de los 20 metros (traspasando tres paredes) en ambientes *indoor* ya no es posible establecer una conexión estable entre el coordinador y los nodos sensores, al igual que a partir de los 45 metros en ambientes *outdoor*. El alcance puede ser mejorado utilizando módulos de comunicación XBee de mayor potencia.

6.6. Prueba 6: Tiempo de vida de las baterías

Tanto el Arduino UNO, como los módulos XBee, consumen menos al estar en modo sleep. No se encontró la forma de sincronizarlos y se decidió escoger sólo uno de los dos para “dormirlo”.

Cabe destacar que cada prueba fue realizada en un solo modo, mientras que al implementar el sistema ambos modos entrarían en funcionamiento de acuerdo con los eventos que surjan, por lo cual es recomendado realizar pruebas más exhaustivas con el fin de determinar el consumo real.

7. Conclusiones

7.1. Principales Logros

Tras la ejecución de este trabajo, se puede decir que:

- Se ha diseñado e implementado un prototipo de red inalámbrica de sensores mediante comunicación ZigBee y la plataforma Arduino para la detección de estado de plazas de un local de estacionamiento vehicular.
- El sistema diseñado es viable y podría ser implementado en un local de estacionamiento real, indistintamente de la ubicación de las plazas y de los nodos sensores. Esto quedó demostrado con las pruebas 2, 3 y 4, cambiando de posición los nodos sensores o la ubicación de las plazas de estacionamiento. En, prácticamente, todas ellas el resultado obtenido por el sistema coincidió con la situación real planteada en los diferentes escenarios.

7.2. Recomendaciones

- Para un correcto funcionamiento de la red, es importante realizar un análisis detallado de los requerimientos que éste pueda necesitar como: cobertura y números de dispositivos que formarán la red. Si se requiere una mayor cobertura o mayor número de equipos terminales, será necesario utilizar los dispositivos Serie 2.
- La incorporación de un motor de paso en cada nodo sensor podría permitir que, con un mismo sensor ultrasónico, se pueda verificar el estado de varias plazas, reduciendo de esta manera el coste de la implementación del sistema.
- Para la comunicación de los dispositivos ZigBee fue utilizado el modo de comunicación serial transparente, en el cual la comunicación se asemeja a lo que sería una transmisión a través de un puerto serial. Sin embargo, para redes de gran tamaño con topología *mesh*, el modo de comunicación recomendado es el

API, en donde un microcontrolador externo se debe encargar de crear una trama específica al tipo de información que se va a enviar.

- Para la implementación del sistema en un estacionamiento real, se tendrán que acomodar los nodos sensores con todos sus componentes en cajas especiales e instalarlos en lugares apropiados en cada una de las plazas, teniendo en cuenta el ángulo de medición de cada sensor de tal manera a evitar posibles interferencias y lecturas fallidas.
- También se recomienda trabajar con fuentes o baterías recargables. Se pueden utilizar alternativas como la utilización de paneles solares.

7.3. Líneas de futuros trabajos posibles

- Analizar e implementar algoritmos de enrutamiento para redes inalámbricas de sensores, de tal manera a trabajar con módulos XBee Serie 2 y con una red *mesh*.
- Utilizar otros tipos de sensores, como por ejemplo sensores de movimiento, sensores de distancia a láser, sensores ópticos, sensores magnéticos, etc. para la red inalámbrica y comparar resultados.
- Complementar la aplicación presentada en este trabajo con un software integrado de gestión de estacionamiento vehicular.
- Desarrollar aplicativos para teléfonos inteligentes que permitan a los conductores ver en tiempo real la información obtenida gracias a la red inalámbrica de sensores.
- Realizar este trabajo en otras plataformas como Raspberry Pi, BeagleBone, inclusive para el nodo coordinador.
- Utilizar otro tipo de estándar de comunicación para la red inalámbrica como, por ejemplo, 6LoWPAN, BLE.

Referencias Bibliográficas

[1] Donoso, P. Análisis de problemas de estacionamiento en grandes instituciones. Aplicación al caso de un campus universitario [en línea] <<http://es.scribd.com/doc/5366642/02-Analisis-de-problemas-de-estacionamiento>> [Octubre de 2013]

[2] Ramos, H. Macia, F. Marcos, D. Redes Inalámbricas de Sensores Inteligentes. Aplicación a la Monitorización de Variables Fisiológicas [en línea] <www.dtic.ua.es/grupoM/recursos/articulos/JDARE-06-H.pdf> [Octubre de 2013]

[3] Aguirre, L. Estudio de una red de sensores sin hilos basadas en la tecnología Arduino bajo protocolos de comunicaciones ZigBee [en línea] <<http://upcommons.upc.edu/pfc/bitstream/2099.1/9511/1/memoria.pdf>> [Octubre de 2013]

[4] Gomez, F. Redes de sensores inalámbricas [en línea] <http://atc.ugr.es/~aprieto/TIC_socio_sanitario/A11_4_05_Redets_sensores.pdf> [Octubre de 2013]

[5] Schaeffer, E. Un Vistazo a los fundamentos de optimización de redes sensoras [en línea] <<http://elisa.dyndns-web.com/~elisa/presentations/optsensora.pdf>> [Octubre de 2013]

[6] Garbarino, J. Protocolos para redes inalámbricas de sensores [en línea] <<http://materias.fi.uba.ar/7500/Garbarino.pdf>> [Octubre de 2013]

[7] ABC Color. Buscan solución al problema de estacionamiento en el Poder Judicial [en línea] <<http://www.abc.com.py/nacionales/buscan-solucion-al-problema-de-estacionamiento-en-el-poder-judicial-322403.html>> [Octubre de 2013]

[8] ZigBee Alliance [en línea] <<http://www.zigbee.org/>> [Octubre de 2013]

[9] Mendez, G ZigBee y sus aplicaciones en el monitoreo y control de procesos a distancia [en línea] <<http://www.fcencias.uaslp.mx/archivos/premiosDES2009/LICENCIATURA-C.Aplicada-MENDEZ-DELGADO.pdf>> [Octubre de 2013]

[10] Ruiz, J. Arduino + XBee [en línea] <http://unicarlos.com/_ARDUINO/Arduino%20+%20XBee.pdf> [Octubre de 2013]

[11] Digi International [en línea] <<http://www.digi.com/>> [Octubre de 2013]

[12] Arduino [en línea] <<http://arduino.cc/>> [Octubre de 2013]

[13] NetBeans IDE [en línea] <<https://netbeans.org/>> [Octubre de 2013]