

Sistema de Telemando Multiplataforma para Robot Móvil

Vargas Cabrera Conrado, Ramos Arreguín Carlos Alberto, Moya Morales Juan
Carlos y García López María del Carmen

Universidad Autónoma de Querétaro, Facultad de Informática
Cuerpo Académico de Cómputo Científico y Tecnológico

Resumen.

El mando a distancia o Teleoperación en robots móviles inalámbricos presenta grandes inconvenientes al decidir el protocolo de comunicación con el cual serán controlados. Los principales problemas son el tiempo de respuesta, distancia de transmisión, costo, y la interferencia entre canales. La aparición de nuevas tecnologías y protocolos de comunicación, ayudan a elegir un eficaz y eficiente sistema de telecontrol. Algunos módulos más utilizados son: xBee, Bluetooth, radio control RF y Wi-Fi. El objetivo de este trabajo es desarrollar e implementar un sistema de telemando inalámbrico para controlar un robot móvil, utilizando el protocolo de comunicación OSC sobre Wi-Fi, así mismo, se interactúa con distintos dispositivos móviles y una FPGA. El sistema consta de la interfaz de control en el dispositivo móvil, el vehículo robótico, y el módulo de comunicación sobre Wi-Fi.

Palabras clave: Telemando, OSC, Wi-Fi, FPGA, iOS, Android, iPhone, iPod, iPad, Robot Móvil.

1. Introducción.

En los últimos años, las comunicaciones inalámbricas se han convertido en líder de los sistemas de envío y recepción de datos a distancia. La tecnología inalámbrica ofrece capacidades de movilidad a los usuarios, sustituyendo las limitantes de una conexión física por cable. Hoy en día, existen en el mercado innumerables dispositivos con conexiones inalámbricas basados en distintos protocolos de comunicación como lo son: Wi-Fi, ZigBee, Bluetooth, GSM, etc. Debido a lo anterior, es necesario realizar una adecuada elección del protocolo, dependiendo de las necesidades de operación del usuario. El IEEE define los protocolos de comunicación entre redes de ordenadores bajo el estándar 802. Asignando a la tecnología Wi-Fi (WLAN) el protocolo 802.11 el cual opera en el rango de frecuencia de 2.4 GHz a 5 GHz. El protocolo 802.15.4 define las comunicaciones ZigBee (PAN) con frecuencia de trabajo de 900 MHz a 2.4GHz. El estándar de Red de Área Personal Inalámbrica (WPAN) define las comunicaciones inalámbricas para Bluetooth bajo el protocolo 802.15, trabaja en el rango de frecuencias 2.4 GHz a 2.48 GHz [1].

El presente artículo describe el desarrollo de aplicaciones para enviar y recibir datos, a través de los protocolos de comunicación inalámbrica Wi-Fi y ZigBee, y los servicios que ofrecen en distintos sistemas operativos, así como en hardware embebido y dispositivos móviles, implementado en el telemando de un robot móvil. Una de las ventajas al desarrollar un sistema de telemando, es que el

usuario tiene la posibilidad de elegir entre diversos dispositivos para operar el sistema, así como el protocolo de comunicación para la transferencia de datos. Actualmente existen dispositivos móviles que con recursos de hardware y software permiten comunicarse utilizando distintos protocolos de comunicación, así mismo, poseen sensores (acelerómetros, giroscopios, sensor de proximidad) disponibles para el desarrollador, basados en diferentes tecnologías. La potencia de tener un sistema operativo en teléfonos celulares y tabletas, permite adecuar controles de mando basados en interfaces gráficas, control de movimiento por medio de acelerómetros, y procesar instrucciones de control en aplicaciones nativas del dispositivo móvil.

Los sistemas operativos iOS de Apple y Android de Google presentes en teléfonos inteligentes y tabletas (Tablets), permiten la interacción del hardware presente en ellos, con sistemas embebidos y dispositivos programables como Arduino y FPGA, entre otros, por medio de comunicación Wi-Fi y Bluetooth.

En el sistema de telemando se describe las ventajas y desventajas que ofrecen los distintos protocolos en comparación a la respuesta obtenida por los microcontroladores. El tiempo de respuesta, distancia de transmisión, costo, tasa de transferencia de datos y la interferencia entre canales, son los aspectos a evaluar para elegir un eficaz y eficiente sistema de telecontrol.

1.1 Antecedentes

Wright, (2002) presenta las especificaciones del protocolo OSC (Open Sound Control), donde describe el protocolo como mecanismo de transferencia de datos, desarrollado para la comunicación entre computadoras, sintetizadores de audio y dispositivos multimedia.

Schmeder, (2008) desarrolla el protocolo uOSC, en su trabajo, describe la implementación del protocolo OSC, para un microprocesador embebido, comunicado mediante cable USB bajo el protocolo serial.

Curiel Razo, (2010) implementa una red de sensores inalámbricos de tipo Ad-hoc, utiliza el protocolo de comunicación ZigBee para la transferencia de datos. En su trabajo describe como logra implementar el monitoreo en tiempo real de la temperatura interior y exterior de un edificio.

Pérez Arreguín, (2011) desarrolla un robot móvil de tracción diferencial, donde implementa el protocolo ZigBee para comunicarse inalámbricamente al robot. Diseña una interfaz gráfica desarrollada en Visual Basic, lo que permite al usuario manipular el robot, por medio de los cursores del teclado de la computadora.

2. Metodología.

2.1 Protocolo OSC.

Open Sound Control (OSC) es un protocolo de comunicación entre computadoras, sintetizadores de sonido y dispositivos multimedia (Wright, 2002). Hoy en día el protocolo OSC se ha desarrollado como librerías, para muchos lenguajes de programación. OSC es un protocolo de alto nivel, designado en un modelo cliente/servidor. Los datos transferidos se agrupan en una unidad llamada paquete. La unidad básica de datos OSC se llama mensaje y tiene la siguiente estructura:

- **address pattern (dirección patrón).** Es una cadena OSC que comienza con el carácter “/”.
- **Type tag string (tipo de cadena).** Es una cadena OSC que comienza con el carácter “;” seguido de una secuencia de caracteres que corresponden al argumento del mensaje OSC. Los tipos de cadena más utilizados son: “i” que representa a un entero de 32 bit, “f” que representa a un valor flotante de 32 bits, y “s” que representa a una cadena de texto (Wright, 2002).

2.2 iOS - XCode.

iOS es el sistema operativo para dispositivos móviles de Apple. La arquitectura de este sistema operativo es similar a la de Mac OSX. iOS funciona como un intermediario entre el hardware presente en el dispositivo móvil y las aplicaciones que se observan en pantalla. La implementación de la tecnología iOS se distingue por niveles, donde el nivel más bajo contiene los servicios fundamentales para la ejecución de las aplicaciones. En el nivel más alto se implementan los servicios que interactúan con el usuario. Estos niveles consisten en un grupo de frameworks, los cuales proveen los objetos y clases necesarios para la construcción de aplicaciones nativas del sistema operativo.

2.3 Android SDK y Android Processing.

Android es el sistema operativo para dispositivos móviles de la compañía Google, está basado en el kernel de Linux.

Para la creación de aplicaciones Android se utiliza el SDK, que se ejecuta bajo la herramienta de desarrollo Eclipse, así como en la plataforma Processing. Processing es un entorno de desarrollo y al mismo tiempo es el nombre del lenguaje de programación, y está desarrollado bajo java, con lo cual es compatible para la creación de aplicaciones Android (© Google, 2012).

3. Desarrollo

El sistema consta de dos unidades principales, la interfaz de control contenida en el dispositivo móvil y la unidad del vehículo robótico, conectados por los protocolos de comunicación WiFi y zigBee. Funciona bajo el modelo cliente-servidor, donde las interfaces de control toman el papel de cliente, ya que son las que envían los datos a través de la red, y la unidad del vehículo robótico se desarrolla como servidor para procesar los datos adquiridos. Para el óptimo manejo y control del robot móvil se diseñaron diferentes interfaces, uno para cada dispositivo móvil, construido en una aplicación nativa de su sistema operativo. Teniendo así, tres distintas aplicaciones divididas de la siguiente manera: aplicación exclusiva para iPod/iPhone, aplicación para iPad y aplicación para Android.

A continuación se describe el proceso del desarrollo de cada aplicación, como también el diagrama de comunicación entre el dispositivo móvil y el robot.

3.1 Desarrollo de la aplicación para iPod.

La aplicación "OSC Control Car iPod" se desarrolla para poder ser utilizado en los siguientes dispositivos: iPod Touch 3G, iPod Touch 4G, iPhone 3GS, iPhone 4, iPhone 4S con sistema operativo iOS 5.0 o posterior. El procedimiento para desarrollar esta aplicación, se describe de la siguiente manera:

1. El primer paso en la construcción de la aplicación, consta en crear un proyecto nuevo basado en la plantilla "Single View Application", ya que proporciona las clases y frameworks indispensables para ejecutar la aplicación en el dispositivo, así mismo, es la base para construir los objetos y clases posteriores que se requerirán dentro de la aplicación.
2. Posteriormente se importan los frameworks VVOSC2 y Accelerometer.
3. A continuación se obtiene la dirección IP y el número de puerto UDP al cual se transferirán los datos.
4. Se crea un objeto OSC. En este paso se manda a llamar al manejador del protocolo OSC, el cual carga en el objeto creado la dirección IP y el número de puerto UDP. Así mismo, se crea un objeto acelerómetro, al cual se le asigna la propiedad de leer el valor del sensor en un intervalo de 0.1 segundos.

5. Se obtienen los datos procedentes de los acelerómetros. En este paso se crea un mensaje de tipo OSC con la siguiente estructura:
OSCMessage: accxyz
Address: /accelerometer
Values: float acceleration.x
float acceleration.y
float accelerarion.z
donde *accelaration* corresponde a los valores de los acelerómetros en los ejes x,y,z.
6. Se inicializa la conexión Wi-Fi mediante el manejador del protocolo OSC.
7. Se envía el mensaje OSC que contiene el valor de los acelerómetros. En este paso se cicla el proceso de envío de datos, renovando los valores obtenidos por los acelerómetros.
8. El último paso es cerrar la conexión Wi-Fi cuando ya no es requerida.

La figura 1, muestra el diagrama del proceso de desarrollo de la aplicación.

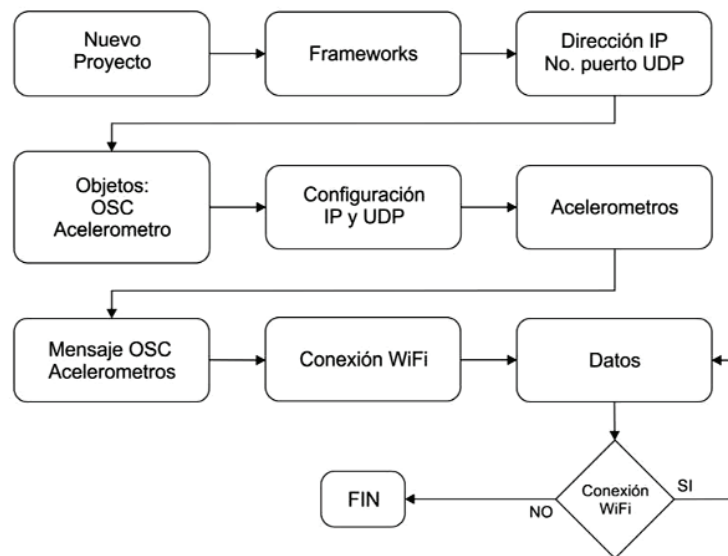


Figura 1. Diagrama de la aplicación "OSC Control Car iPod".

3.2 Desarrollo de la aplicación para iPad.

La aplicación "OSC Control Car iPad" se desarrolla para poder ser utilizado en los siguientes dispositivos: iPad, iPad2, iPad3 con sistema operativo iOS 5.0 o posterior. En el procedimiento para desarrollar la aplicación del dispositivo iPad, se utiliza la metodología que se detalla en la figura 2, más la implementación de recibir video por streaming. A continuación se describe el proceso de desarrollo para la aplicación "OSC Control Car iPad".

Siguiendo la metodología de la figura 1, se utilizan los primeros 4 pasos para el comienzo del desarrollo de la aplicación del dispositivo iPad. Posteriormente se implementan los siguientes pasos. Se obtiene la dirección URL para la recepción de video. A continuación se crea un objeto webpage, donde se asigna la dirección URL, este objeto cargará una página web dentro de una vista en la aplicación. Si no existe recepción de video, se intenta cargar nuevamente la dirección URL dentro del objeto webpage, por el contrario, si la recepción de video es exitosa, se continua con el paso 5 del procedimiento descrito, hasta concluir la metodología.

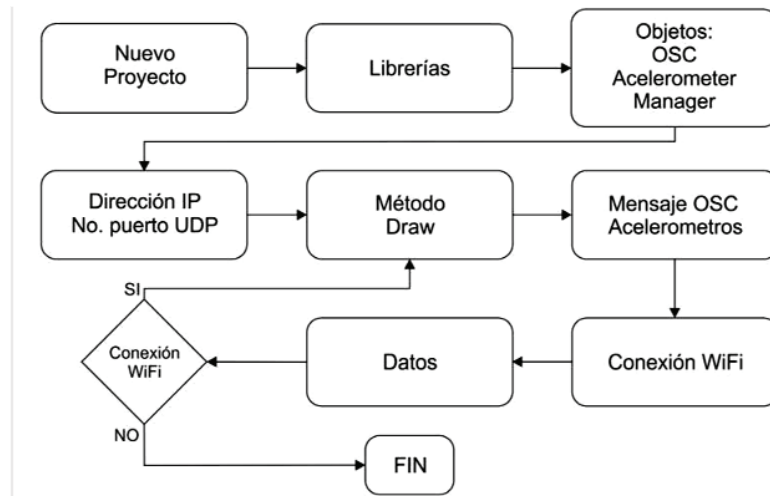


Figura 2. Diagrama del procedimiento de desarrollo para la aplicación “OSC Control Car” en el sistema operativo Android.

3.3 Desarrollo de la aplicación para Android.

La aplicación “OSC Control Car” se desarrolla para poder ser utilizado en teléfonos celulares y tabletas con sistema operativo Android 2.2 o posterior. El procedimiento para desarrollar esta aplicación se describe a continuación.

1. Primeramente se genera un nuevo proyecto Android dentro del SDK, el cual contiene los recursos necesarios para poder exportar la aplicación al dispositivo móvil. Se procede a importar las librerías OSC53 y la clase accelerometerManager.java
2. Se crea un objeto OSC5, al cual se le pasarán los parámetros del protocolo OSC. Así mismo se crea un objeto acelerómetro para obtener los datos del sensor.
3. A continuación se obtiene la dirección IP y el número de puerto UDP, al cual se transferirán los datos.
4. Posteriormente se manda a llamar el método Draw. En este paso se renuevan los datos obtenidos del sensor acelerómetro. Se crea un mensaje de tipo OSC con la siguiente estructura:
OSCMessage: accxyz
Address: /accelerometer
Values: float acceleration.x
float acceleration.y
float accelerarion.z
 donde *acceleration.x.y.z* corresponden a los valores de los acelerómetros en los ejes x,y,z.
5. Se inicializa la conexión Wi-Fi mediante el objeto OSC5.
6. Se envía el mensaje OSC que contiene el valor de los acelerómetros. En este paso se comprueba si existe conexión por Wi-Fi, de ser positivo se devuelve al método Draw, de lo contrario se finaliza el proceso.

3.4 Programa servidor en processing para recibir datos OSC.

El programa servidor que se describe a continuación y se desarrolla en el entorno de programación Processing. Una de las ventajas de este SDK es su versatilidad de ser implementado en distintos sistemas operativos como Linux, Mac OS, Windows y Android. Por tal motivo, se decidió desarrollar el programa en Processing, con lo cual se obtiene un sistema multiplataforma. La figura 3, describe el diagrama del proceso de desarrollo para la creación del programa servidor.

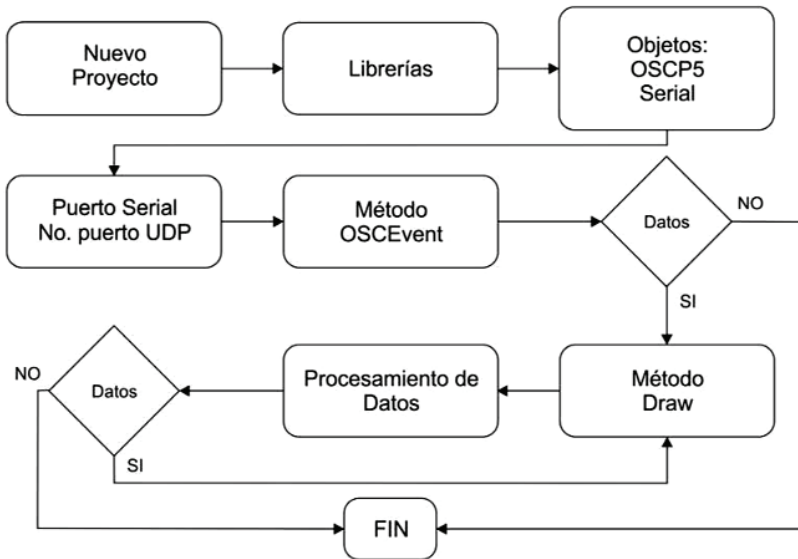


Figura 3. Diagrama de la aplicación servidor.

A continuación se describen los pasos para desarrollar el programa servidor. Se detalla el método de obtención de datos procedentes del dispositivo móvil. Así mismo, se detalla el proceso de envío de datos a través del puerto serial de la computadora hacia el robot móvil, por medio del protocolo de comunicación zigBee.

1. El primer paso es crear un nuevo sketch e importar las librerías necesarias para la ejecución de la aplicación (Serial, OSCP54). La librería processing.serial permite la lectura y escritura de datos con elementos externos de la computadora. Por otra parte la librería OSCP5, implementa los métodos necesarios para el manejo del protocolo OSC sobre Wi-Fi.
2. Se crea un objeto Serial. En este paso se identifica el puerto serial COM correspondiente al módulo xBee. Así mismo, se crea un objeto OSCP5, el cual se le asigna la propiedad de escuchar mensajes entrantes por el número de puerto UDP a establecer.
3. Posteriormente se manda a llamar al método OSCEvent, donde se reciben datos de tipo OSCMessage, procedentes del dispositivo móvil.
4. Se manda a llamar al método Draw. En este paso se renuevan los datos obtenidos con anterioridad. A continuación se procesan los datos y se envían a través del puerto serial correspondiente al módulo Xbee.
5. El último paso consiste en identificar si existen datos sin enviar, de ser positivo se retorna al método draw, de lo contrario se finaliza la conexión al puerto serie.

3.5 Robot móvil en FPGA

El Lenguaje de Descripción de Hardware de muy alta Velocidad (VHDL), es un lenguaje ampliamente empleado en numerosos sistemas digitales, por lo cual, es posible implementar un sistema digital embebido para el control de un carro móvil tele-operado a distancia de manera inalámbrica. Los dispositivos programados con este lenguaje son: GAL (Generic Array Logic), CPLD (Complex Programmable Logic Device) y FPGA (Field Programmable Gate Array), por mencionar los más importantes. La figura 4 muestra el diagrama de bloques principal del sistema para el control del carro anteriormente mencionado.

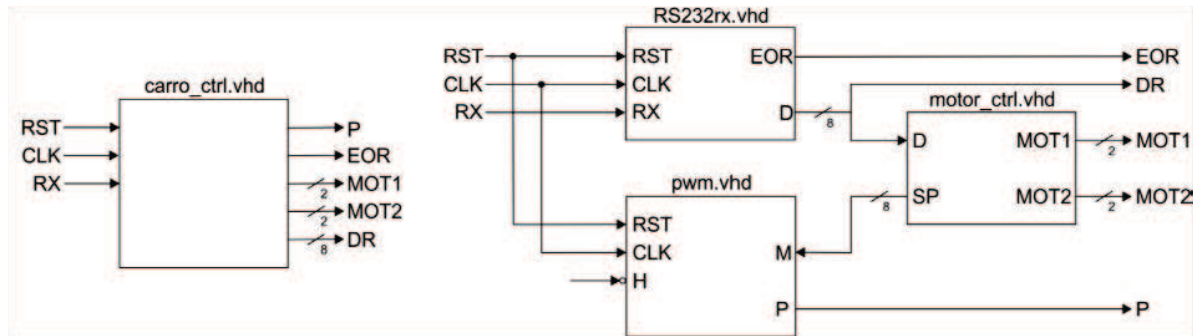


Figura 4. Diagrama General y módulos principales para el control del carro.

Las señales de entrada son un RST maestro y una señal de reloj principal, la señal RX proviene del receptor inalámbrico X-Bee, este recibe los datos para el funcionamiento de los motores del carro. La señal P es la salida de un PWM (Pulse Width Module) para controlar la velocidad de los motores cuando el carro se dirija hacia adelante, hacia atrás, vuelta a la izquierda o vuelta a la derecha. Las señales MOT1 y MOT2 controlan el sentido de giro y encendido/apagado de los motores. Por último la señal DR puede ser empleada para ver el número recibido por RX y saber que instrucción se está ejecutando. La entidad principal consta de 3 módulos principales para llevar a cabo dicho sistema de telemando, los cuáles son:

- RS232rx.vhd
- pwm.vhd
- motor_ctrl.vhd

4. Proceso de comunicación dispositivo móvil – robot móvil.

El proceso de comunicación entre el dispositivo móvil y el robot, se presenta con relativas diferencias dependiendo el dispositivo móvil a elegir. A continuación se describe el proceso para comunicar a cada uno de los sistemas.

La figura 5, muestra el detalle de comunicación entre el dispositivo móvil iPad y el robot móvil, con un sistema de retroalimentación por video streaming.

El proceso se inicia cuando el usuario ejecuta la aplicación “OSC Control Car iPad” en su iPad, en la vista principal se ingresa la dirección IP y el número de puerto UDP de la computadora servidor. En una segunda vista de la aplicación, se transmiten los datos obtenidos por los acelerómetros, a través del protocolo OSC sobre la tecnología Wi-Fi.

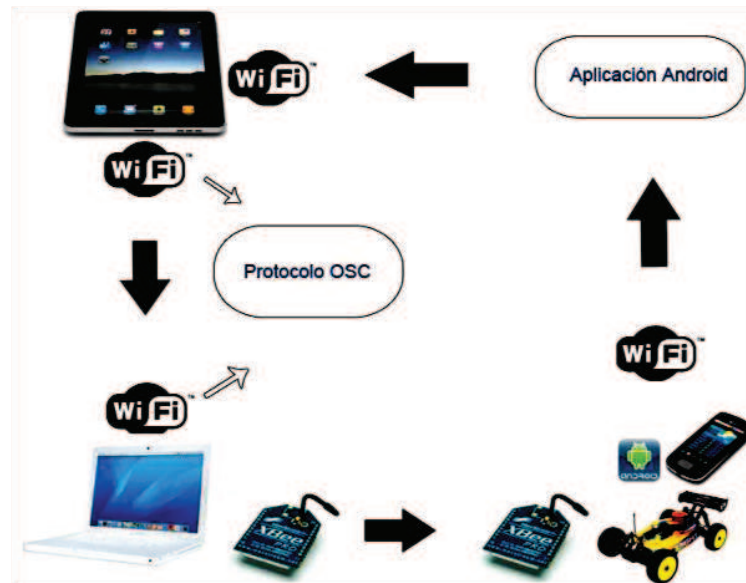


Figura 5. Diagrama del modelo de comunicación iPad – Robot móvil con retroalimentación por medio de video streaming.

La computadora ejecuta el programa servidor, el cual procesa los datos obtenidos, dentro de este programa, se implementan instrucciones de control para la manipulación del robot móvil, así mismo, se crean nuevos datos y estos son enviados a través del puerto serial correspondiente al módulo Xbee. Esta información se recibe en la tarjeta Arduino o FPGA, que contienen un programa para responder de acuerdo a los datos recibidos. El robot móvil está equipado con un teléfono celular, el cual ejecuta la aplicación IPWebCam, que es la encargada de transmitir video por streaming a través del protocolo Wi-Fi al iPad, cerrando así el sistema de retroalimentación, dando al usuario la posibilidad de tomar nuevas decisiones en el control del robot.

5. Pruebas y Resultados

Para probar el sistema de telemando se dividió en dos partes. En el primer conjunto de pruebas, se instaló la aplicación "OSC Control Car iPod" en tres dispositivos diferentes, se hizo lo mismo para las otras dos aplicaciones. Comprobando así, el funcionamiento adecuado de las aplicaciones, al enviar los datos obtenidos por el sensor del acelerómetro. En el mismo grupo de pruebas se realizó el modelado y simulación de movimiento de un dispositivo móvil, como se puede observar en la figura 6, se dibuja una caja que se mueve en base a la información obtenida de los acelerómetros. El segundo conjunto de pruebas, se basa en analizar el tiempo de respuesta que ofrece la FPGA, con respecto al procesamiento de las instrucciones de control para manipular el robot móvil. Se observó que la tarjeta FPGA refleja resultados deseables, dando menores tiempos de retraso en el procesamiento de los datos recibidos.



Figura 6. Simulación de valores obtenidos por acelerómetros.

Así mismo, se realizaron pruebas de la distancia efectiva que ofrece el conjunto de protocolos de comunicación utilizados, y el tiempo de respuesta al enviar y recibir datos. Debido a que las pruebas se realizaron con la infraestructura de comunicación Wi-Fi que ofrece la facultad, el ancho de banda disminuye con el aumento de computadoras conectadas a la red, dando como resultado retardos en la comunicación entre dispositivos. Posteriormente se realizó una conexión de tipo Ad-hoc entre el dispositivo móvil y la computadora, lo que ofrece un ancho de banda de 100 Mbps de transferencia máxima de datos, y una distancia efectiva de 25 metros por medio del protocolo Wi-Fi. La comunicación entre módulos xBee ofrece un rango de operación de 30 metros atravesando paredes, y 100 metros en lugares abiertos.

6. Conclusiones y trabajo a futuro

En este artículo, se abordó un sistema de telemando para controlar un robot móvil. Se presenta el uso de distintas tecnologías y protocolos de comunicación como lo son Wi-Fi, ZigBee, Xbee, OSC (Open Sound Control), para el manejo y transferencia de datos por medio de comunicación inalámbrica. Se detalla la implementación de estos protocolos, en el desarrollo de aplicaciones de diversos dispositivos móviles (iPod, iPhone, iPad, teléfonos celulares con SO Android, tablet's), así como en programas, que se pueden ejecutar en diversos sistemas operativos como Windows, Linux, Mac OS, Android, iOS. Por tal motivo se consiguió desarrollar un sistema multiplataforma que se adecue a las necesidades del usuario. Se implementó el uso del protocolo OSC, como protocolo de comunicación sobre Wi-Fi, entre dispositivos móviles, computadoras y sistemas embebidos.

Se desarrolló un robot móvil de tracción diferencial, el cual ejerce el rol de sistema esclavo, de un modelo maestro/esclavo. Para la implementación de la parte de control del robot, se utilizó una FPGA. Como se observa en las pruebas realizadas, los resultados son eficientes en el sistema de telemando descrito, pero se mantiene un puente entre el dispositivo móvil, y el robot a manipular, lo cual implica un retardo en el tiempo de respuesta.

Como trabajo a futuro se implementará un sistema de telemando sobre el protocolo de comunicación inalámbrico Bluetooth, mejorando la tasa de transferencia de datos, el tiempo de respuesta entre dispositivos, desaparecer el puente existente entre el dispositivo móvil y el robot.

La principal aportación de este trabajo es el desarrollo de una aplicación para dispositivos móviles, la cual utiliza un modelo de comunicación inalámbrica para comunicarse al robot. Se utilizó el protocolo de comunicaciones OSC (Open Sound Control) sobre el protocolo de red Wi-Fi, el cual permite una tasa de transferencia de datos alta y una mayor distancia entre los dispositivos móviles y el robot. Se presentan tres alternativas diferentes: un modelo de comunicación utilizando un iPod/iPhone y el robot móvil, una segunda alternativa utilizando un iPad que es retroalimentado mediante video por streaming, y por último un sistema de comunicación utilizando un teléfono celular con sistema operativo Android.

7. Referencias

- [1] H. Labiod, H. Afifi y C. De Santis; Wi-Fi™, Bluetooth™, ZigBee™ and Wimax™; Springer; ISBN: 978-1-4020-5397-9; Dordrecht, The Netherlands 2007.
- [2] R. Troncoso; Electrónica Digital y Lógica Programable; Universidad de Guanajuato, 2a Ed.; ISBN:968-8644498; Guanajuato Mexico 2007.
- [3] M. Aceves, R. Arreguin; Fundamentos de Sistemas Embebidos, Mediante Lenguajes Descriptivos de Hardware; Asociación Mexicana de Mecatrónica A.C.; ISBN 978-607-95347-4-5; Santiago de Querétaro, Querétaro 2012.
- [4] Android Developer; Sensor Type; http://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER; última consulta 15/01/2012;