

Motion control of a robotic arm from a computer using free control software and wireless communication with XBEE modules

Control de los movimientos de un brazo robótico desde un computador utilizando software libre de control y comunicación inalámbrica con módulos X-BEE

C. Vélez^{1*}, F. Orosco¹, M. A. Álvarez²

¹Facultad de Ingeniería en Electricidad y Computación. Escuela Superior Politécnica del Litoral (ESPOL).
Campus Gustavo Galindo, Km. 30.5 Vía Perimetral.

²Facultad de Ingeniería en Electricidad y Computación, Centro de Visión y Robótica. Escuela Superior Politécnica del Litoral (ESPOL). Campus Gustavo Galindo, Km 30.5 Vía Perimetral Guayaquil, Ecuador.

*Autor principal/Corresponding author, e-mail: cavelez@espol.edu.ec

Editado por/Edited by: Cesar Zambrano, Ph.D.

Recibido/Received: 27/01/2014. Aceptado/Accepted: 20/03/2014.

Publicado en línea/Published on Web: 19/12/2014. Impreso/Printed: 19/12/2014.

Abstract

This document demonstrates the control of a robotic arm wirelessly by using free software. For this, an operator (user) will be the one giving the orders and actions to be taken, using a computer through various input devices and XBEE modules that are part of our communication channel. The fact that the system includes wireless communication allows distancing the operator from the robot arm and thus can reduce risk when working in addition a large number of wires that may hinder or impede the development of any activity is eliminated. The system has several options for handling the mechanical arm which has developed a free software interface that would maximize all basic input devices of the computer. The software in question is the Processing program which allows the arm to be controlled by keyboard, mouse or even using motion detection through a webcam, and to employ USB output port to transmit data through the modules XBEE. Finally have to say that the results obtained in a complete system prototype robotic arm can serve as a basis for developing several applications including, for example, remote rovers.

Keywords. Processing, robotic arm, wireless communication

Resumen

El presente trabajo muestra el control de un brazo robótico de forma inalámbrica usando software libre. Para esto un operador (usuario) será quien dé las órdenes y acciones a realizar, haciendo uso de un ordenador a través de los diferentes dispositivos de entrada y los módulos XBEE que forman parte de nuestro canal de comunicación. El hecho que el sistema incluya comunicación inalámbrica permite distanciar al operador del brazo robótico y de esta manera se pueda reducir el riesgo al momento de trabajar, además que se elimina una gran cantidad de cables que pueden obstaculizar o dificultar el desarrollo de cualquier actividad.

El sistema tiene varias opciones de manipulación del brazo mecánico para lo cual se ha desarrollado una interfaz en software libre que permita aprovechar al máximo todos los dispositivos de entrada básicos del computador. El software en mención es el programa Processing que permite que el brazo sea controlado por el teclado, el mouse o incluso usando detección de movimiento a través de una cámara web, y emplear un puerto USB como salida para la transmisión de datos a través de los módulos XBEE.

Finalmente queda decir que los resultados del sistema completo obtenidos en un brazo robótico de prototipo pueden servir como base para el desarrollo de varias aplicaciones entre ellas, por ejemplo, telecontrol de robots exploradores.

Palabras Clave. Procesamiento, brazo robótico, comunicación inalámbrica.

Introducción

Los robots telemanipuladores fueron los primeros o progenitores de todos los robots. Desarrollados en un inicio

con transmisión mecánica, el avance de la electrónica y el conocimiento de la teoría de servocontrol permitieron reemplazar la transmisión mecánica por eléctrica y al mismo tiempo consiguiendo mejorar en aspectos

de precisión y control a los robots que empezaron a ser usados tanto en la industria nuclear como la industria submarina o la industria espacial [1].

Los robots manipuladores o brazos robóticos se los puede clasificar según su tipo o uso. Existen brazos robóticos de tipo industrial, tipo comercial y los de diseño propio. Aquellos brazos comprendidos dentro del uso industrial como PUMA o SCARA son elaborados en una arquitectura cerrada y la mayoría de veces programados en su propio entorno de programación, por lo que los costos son elevados. Los brazos que se encuentran dentro del uso comercial como LEGO MINDSTORMS o MECANO son de arquitectura abierta y son muy utilizados en investigación. Por último los pertenecientes al grupo de diseño propio suelen ser también de arquitectura cerrada, de ambiente de programación limitado pero de un reducido costo. Todo lo mencionado se puede resumir en la Tabla 1.

Cuando se trabaja con robots, en especial con brazos robóticos, resulta de mucha importancia y ayuda poder saber o conocer que movimientos debe realizar cada articulación del robot para que el mismo pueda llegar a ubicarse en una posición específica. Una herramienta que permite determinar los movimientos de las articulaciones es el modelo cinemático de un robot.

La cinemática estudia el movimiento del robot con respecto a un sistema de referencia. Existen dos problemas fundamentales a resolver en la cinemática del robot: el primero se conoce como el problema cinemático directo, consiste en determinar cuál es la posición y orientación del extremo final del robot con respecto a un sistema de coordenadas que se toma como referencia (los valores de las articulaciones y los parámetros geométricos de los elementos del robot son conocidos); el segundo, denominado problema cinemático inverso, resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas.

Cinemática Inversa

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot para que su extremo se posicione y oriente según una determinada localización espacial [1].

A pesar que el problema cinemático inverso tiene algunas desventajas frente al método cinemático directo, la mayor parte de los robots poseen cinemáticas relativamente simples que facilitan en cierta medida la resolución de un problema cinemático inverso. Por ejemplo, si

Tipo de robot	Tipo industrial	Tipo comercial	Diseño propio
Ejemplos	PUMA	MECANO	Material reciclado
Costo	Alto	Medio	Bajo
Arquitectura abierta	No	Si	No
Programación robusta	No	Si	No
Mantenimiento	Alto costo	Bajo costo	Bajo costo

Tabla 1: Clasificación de robots manipuladores [2].

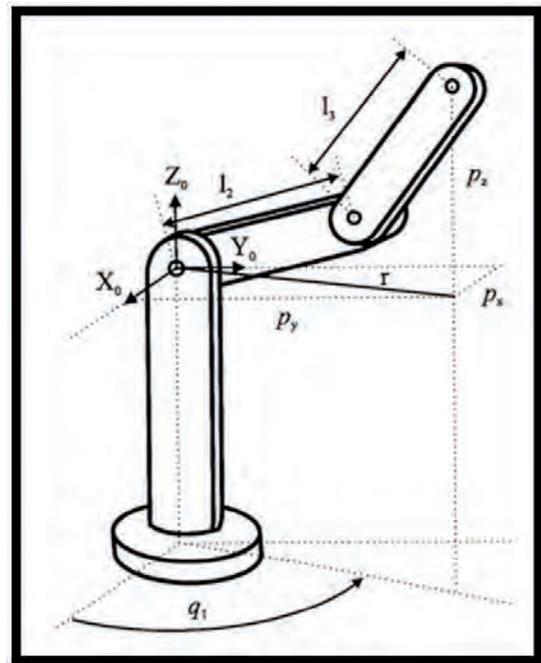


Figura 1: Robot Articular de 3 GDL [4].

se consideran solo los 3 primeros grados de libertad de muchos robots, estos tienen una estructura planar. Esta circunstancia facilita la resolución del problema.

Los métodos geométricos permiten obtener normalmente los valores de las primeras variables articulares que son las que consiguen posicionar el robot. Para ello utilizan relaciones trigonométricas y geométricas sobre los elementos del robot.

Resolución del Problema Cinemático Inverso por Métodos Geométricos

Este procedimiento es adecuado para robots de pocos grados de libertad o para el caso de que se consideran sólo los primeros grados de libertad, dedicados a posicionar el extremo, como el mostrado en la Figura 1.

El valor de q_1 (ángulo a girar la primera articulación del brazo) se obtiene:

$$q_1 = \arctg \left(\frac{p_y}{p_x} \right) \tag{1}$$

donde p_x y p_y son las coordenadas en el eje X y Y de la posición en la cual se quiere ubicar el extremo del brazo.

Considerando que los siguientes elementos se encuentran contenidos en un plano y utilizando el teorema del coseno:

$$p_x^2 + p_y^2 + p_z^2 = l_2^2 + l_3^2 + 2l_2l_3 \cos q_3 \tag{2}$$

$$\cos q_3 = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3} \tag{3}$$

donde p_z es la coordenada en el eje Z de la posición a ubicar el extremo, q_3 es el ángulo a girar la tercera articulación del brazo, l_2 y l_3 son las longitudes del segundo y tercer elemento del brazo, respectivamente.

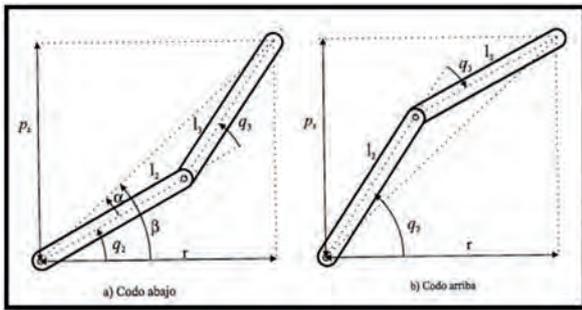


Figura 2: Elementos 2 y 3 del robot de la Figura 1 contenidos en un plano. a) Configuración codo abajo. b) Configuración codo arriba [4].

Esta expresión permite obtener q_3 en función del vector de posición del extremo p . No obstante, y por motivos de ventaja computacional, es más conveniente utilizar la expresión de la arcotangente en lugar del arcoseno.

Puesto que:

$$\sin q_3 = \pm \sqrt{1 - \cos^2 q_3} \tag{4}$$

se tendrá:

$$q_3 = \arctg \left(\frac{\pm \sqrt{1 - \cos^2 q_3}}{\cos q_3} \right) \tag{5}$$

El signo de la solución de la expresión anterior corresponde a la configuración de codo arriba como en la Figura 2a o codo abajo como en 2b.

El cálculo de q_2 (ángulo a girar la segunda articulación del brazo) se hace a partir de la diferencia entre β (ángulo medido entre el plano XY y la ubicación deseada del extremo del brazo) y α (ángulo medido entre la posición del segundo elemento del brazo y la ubicación deseada del extremo del brazo robótico):

$$q_2 = \beta - \alpha \tag{6}$$

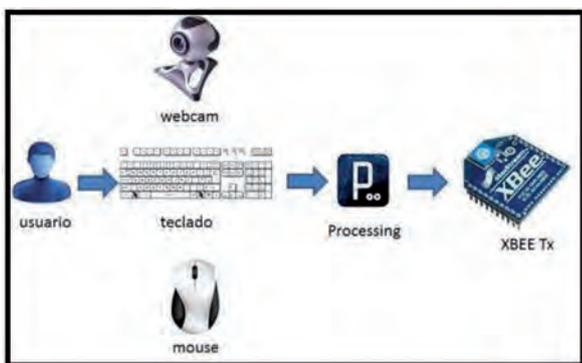


Figura 3: Proceso de generación de instrucciones y envío de la información.



Figura 4: Proceso de recepción de instrucciones y ejecución por parte del brazo robótico.

siendo:

$$\beta = \arctg \left(\frac{p_x}{\pm \sqrt{p_x^2 + p_y^2}} \right) \tag{7}$$

$$\alpha = \arctg \left(\frac{l_3 \sin q_3}{l_2 + l_3 \cos q_3} \right) \tag{8}$$

finalmente:

$$q_2 = \arctg \left(\frac{p_x}{\pm \sqrt{p_x^2 + p_y^2}} \right) - \arctg \left(\frac{l_3 \sin q_3}{l_2 + l_3 \cos q_3} \right) \tag{9}$$

Materiales

El presente trabajo se coloca dentro del grupo de diseño propio donde a pesar de tener una arquitectura cerrada, presenta la ventaja que su programación no es limitada. De hecho por contar con un software libre las interfaces con las cuales se puede manipular al brazo robótico son extensas debido a que las funciones se encuentran implementadas en librerías y al usar software libre es posible programar funciones.

La mayoría de brazos robóticos que existen son manipulados por una unidad de procesamiento robusta como suele ser un computador pero con el limitante que la conexión entre brazo y computador requiere de cables. El proyecto realiza la comunicación del brazo y el computador de manera inalámbrica permitiendo así alejar al operario de una estación de trabajo y ponerlo a mayor distancia de la operación del brazo.

El brazo robótico que se utiliza en el proyecto es un brazo de 4GL (4 Grados de Libertad) elaborado con madera y siguiendo el modelo o estructura en [6]. Dentro de la fabricación del brazo se utiliza servomotores de modelismo (HITEC HS-311), ya que ofrecen la suficiente potencia para los movimientos del brazo y simplifica la operación.

Para el control del brazo se emplea una interfaz desarrollada en el programa PROCESSING que es un software libre. Este programa basado en JAVA cuenta con librerías para comunicación serial y periféricos como el teclado, mouse y cámara web que son el principal uso en el proyecto. La computadora, portátil o de escritorio, debe contar con los periféricos requeridos para la manipulación del brazo.

La comunicación inalámbrica se la realiza por medio de los módulos XBEE que cuentan con el protocolo ZigBee embebido. Estos módulos resultan adecuados para el proyecto porque las instrucciones a enviar no son tramas demasiado largas y la velocidad máxima de 256kbps en la que trabajan son suficientes para la aplicación; además entre otras características destacan: el alcance que ofrecen (100 m en lugares con línea de vista), bajo consumo (menor a 50mA), potentes y precio accesible, convirtiendo de esta manera a los xbee en la

Tecla	Articulación afectada	Movimiento	Servomotor
Y	Hombro	Hacia Arriba	2
B	Hombro	Hacia Abajo	2
G	Pinza	Abrir	4
H	Pinza	Cerrar	4
F	Eje de Rotación	De izquierda a derecha	1
J	Eje de Rotación	De derecha a izquierda	1
T	Codo	Hacia Arriba	3
V	Codo	Hacia Abajo	3

Tabla 2: Funcionamiento brazo robótico.

primera opción a considerar para este tipo de aplicación. La comunicación es punto a punto entre la estación de control (computador) y la estación remota que controla el brazo robótico por medio de un microcontrolador como unidad de procesamiento de las órdenes recibidas.

Las Figuras 3 y 4 describen el esquema de comunicación implementado.

Método

La interfaz gráfica permite operar el brazo robótico con la opción de mouse/teclado o la opción de reconocimiento de movimientos frente a la cámara web. De acuerdo a la opción de movimiento, el funcionamiento del brazo responde a la Tabla 2.

Interfaz Gráfica del Proyecto

Processing es un lenguaje de programación de código abierto basado en JAVA y orientado a los usuarios que desean crear imágenes, animaciones, interfaces e interacciones con diferentes medios o periféricos [3].

¿Por qué usar Processing y no otros lenguajes de programación?

La respuesta viene dada por el gran número de librerías que tiene implementadas; si bien es cierto hay muchos

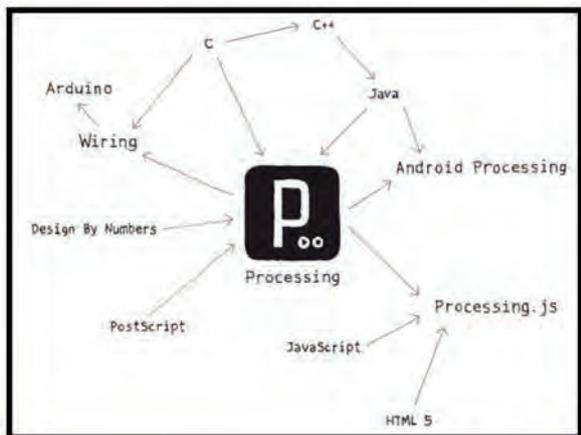


Figura 5: Comparación entre Processing y otros lenguajes de programación [5].

programas que nos proveen opciones similares, Processing tiene la ventaja de ser código abierto, lo que permite modificar las implementaciones ya existentes, además que provee toda capacidad para manipular cualquier dispositivo que se pueda conectar vía USB. Una de las cualidades más interesantes es la portabilidad que ofrece y la facilidad para mover los recursos, pues no se necesita instalar el software y provee compatibilidad entre los sistemas operativos. En la Figura 5 se muestra la comparación entre distintos lenguajes de programación.

Para la creación del proyecto se maneja varias funciones que permiten al usuario manipular de forma exitosa periféricos como mouse, teclado, cámara web y módulos X-BEE.

En la manipulación por mouse, se tiene la función **mousePressed**, la cual indica si en algún momento se ha presionado el mouse (click), retorna un valor booleano que de ser verdadero será true (1) caso contrario false (0). Adicionalmente se usa las funciones **mouseX** y **mouseY**, las cuales nos dan las coordenadas en X y Y de la posición actual del cursor y que son de gran importancia para saber sobre qué cuadrante se encuentra el operario y qué acción debe tomar el robot.

En la manipulación por teclado, se aprovecha la función **Key**, la cual almacena la tecla que el usuario presiona y retorna su código ASCII. Hay varias letras para poder controlar el movimiento del servo-motor como se detalla en la Tabla 2.

Con respecto a la cámara web se efectúa un barrido de imágenes para detectar el movimiento, de la misma forma el programa sondea el cuadrante en donde existe el movimiento y en función de eso el brazo robótico realiza una acción. Para cubrir con este propósito la función **ABlob ab = (ABlob)flob.getABlob(i)** adquiere, utilizando la detección de movimiento combinado con el barrido de imágenes, la posición en que ocurrió el movimiento y la función **image(flob.getSourceImage(), 0, 0, width, height)**, permite mostrar en pantalla la imagen capturada por la cámara web (movimientos del operario).

Las 3 acciones anteriores se ayudan de posiciones o cuadrantes para determinar la acción a tomar. Es importante mencionar que según sea el caso lo que se transmite son letras (en código ASCII) a través de un puerto USB al cual está conectado el Xbee. Para la transmisión de los comando de acción se usa 4 funciones importantes: **println(Serial.list())**, **String portName = Serial.list()** [5], **myPort = new Serial(this, portName,**

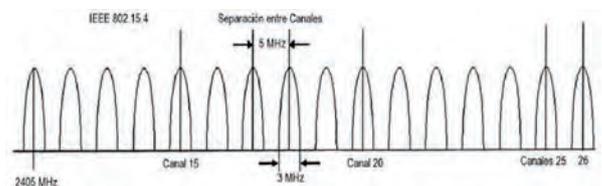


Figura 6: Canales disponibles para el protocolo IEEE 802.15.4 [?].

Frecuencia [GHz]					
Canal	Hexadecimal	Inferior	Central	Superior	Comando AT
11	0x0B	2,4025	2,4050	2,4075	ATCH0B
12	0x0C	2,4075	2,4100	2,4125	ATCH0C
13	0x0D	2,4125	2,4150	2,4175	ATCH0D
14	0x0E	2,4175	2,4200	2,4225	ATCH0E
15	0x0F	2,4225	2,4250	2,4275	ATCH0F
16	0x10	2,4275	2,4300	2,4325	ATCH10
17	0x11	2,4325	2,4350	2,4375	ATCH11
18	0x12	2,4375	2,4400	2,4425	ATCH12
19	0x13	2,4425	2,4450	2,4475	ATCH13
20	0x14	2,4475	2,4500	2,4525	ATCH14
21	0x15	2,4525	2,4550	2,4575	ATCH15
22	0x16	2,4575	2,4600	2,4625	ATCH16
23	0x17	2,4625	2,4650	2,4675	ATCH17
24	0x18	2,4675	2,4700	2,4725	ATCH18
25	0x19	2,4725	2,4750	2,4775	ATCH19
26	0x1A	2,4775	2,4800	2,4825	ATCH1A

Frecuencia Base 2,405 GHz

Tabla 3: Canales de frecuencia y su respectivo comando AT [5].

9600) y myPort.write('A') [4]. La primera permite enlistar los puertos que hay disponibles para establecer la comunicación, la segunda le asigna a una variable el puerto que se desea utilizar, la siguiente establece la comunicación indicándole la velocidad a la que desea transmitir y con la última función se especifica que se va a transmitir.

Comunicación Inalámbrica

Los módulos empleados son los Xbee que tienen embebido en sí el protocolo ZigBee 802.15.4, estándar que define un conjunto de protocolos para la configuración de redes inalámbricas de corta distancia y baja velocidad de datos. Opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz, y puede transferir datos hasta 250Kbps.

Si bien al igual que ZigBee hay otros estándares que se pueden usar en redes de corto alcance tales como el 802.11 y Bluetooth. ZigBee es un estándar ampliamente utilizado para implementar redes de sensores y actuadores que deban operar a batería.

Para el proyecto se emplea los módulos en su modo de operación transparente. En este modo la información que se recibe por el pin Data In se guarda en el buffer de entrada, se la integra a un paquete RF y se transmite. Para el proyecto se selecciona transmitir a una tasa de 9600 baudios que es la velocidad por defecto del

módulo Xbee. En la sección de Resultados se detalla la configuración del hardware.

En la Figura 6 y la Tabla 3 se muestran todos los posibles canales de comunicación del módulo y entre los cuales se debe escoger uno de ellos al configurar los módulos. El seleccionar un canal que esté libre de interferencia permite que la comunicación entre la estación de control y la estación remota sea confiable.

Brazo Robótico

En la fabricación del brazo robótico se usa servomotores de modelismo en cada uno de los 4 grados de libertad. Debido a que los servomotores funcionan con una señal PWM específica, es necesario contar con una señal para cada motor. El cerebro del brazo será un microcontrolador PIC16F887. Este microcontrolador cuenta con 2 fuentes de señal PWM, al no ser suficientes, se emplea las fuentes de señal y se genera la señal PWM que se ajuste a las necesidades del proyecto dentro del microcontrolador. En 4 pines del PIC se coloca la señal PWM para cada uno de los motores.



Figura 7: Electrónica empleada entre PIC y motores.

Número de Ejes	4 Incluyendo la pinza
Motores	Servomotores HS-311
Control de los servos	PWM
Altura en reposo	150 mm
Altura máxima	250 mm
Alcance medio	100 mm
Apertura de la pinza	25 mm
Peso sin baterías	300 gramos
Rango de giro por eje	180 °
Precisión en cada eje	5 °
Tensión de los servos	5V nominal
Consumo en reposo	8 mA cada servo.
Consumo en movimiento	200 mA cada servo

Tabla 4: Detalle técnico del brazo robótico.

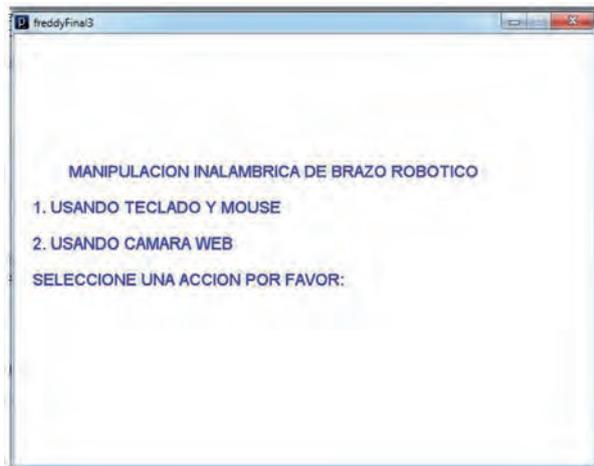


Figura 8: Menú principal de la aplicación.

La generación de las señales PWM se lo hace colocando en nivel alto el pin y luego de un tiempo pasar a nivel bajo que durará un tiempo específico. El tiempo que permanezca en nivel alto el pin dependerá de las instrucciones que reciba el microcontrolador por medio del Xbee. Según la instrucción receptada se aumentará o disminuirá el tiempo en alto de la señal. Esta misma operación se repite en los 4 servomotores, es decir, se requiere de 8 comandos distintos para operar los 4 motores.

Debido a la corriente que consume cada servomotor en funcionamiento, resulta imposible conectar directamente el servomotor a los pines del microcontrolador, motivo por el cual se colocó una etapa de potencia intermedia. Esta etapa son puentes H para cada motor. Se utilizó el L293D, ya que este integrado está conformado por 4 puentes H y es la cantidad que se requiere para los motores. La Figura 7 muestra un diagrama de bloques representando al microcontrolador, puentes H y servomotores con las señales respectivas para cada motor.

Resultados y Discusión

Usando las funciones especificadas en la parte de la explicación del programa Processing en la sección de Mé-

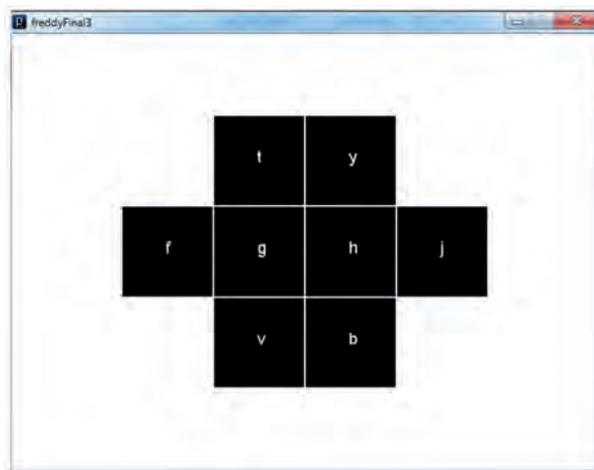


Figura 9: Interfaz cuando se ingresa a la opción de manipulación por teclado o mouse.

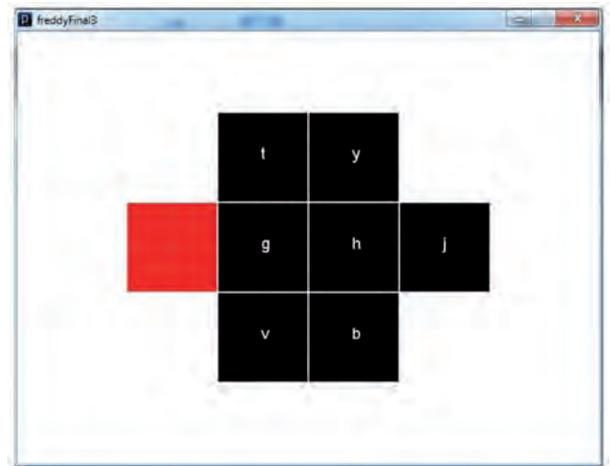


Figura 10: Cambio de color para indicar al usuario que letra se ha presionado.

todo junto con funciones que permiten modificar la estética de la interfaz se obtiene la aplicación de la Figura 8.

Si el usuario presiona la primera opción, el programa desplegará una nueva interfaz tal como se muestra en la Figura 9. Desde aquí el usuario puede presionar la tecla que se indica en la pantalla desde el teclado, o puede dirigir al mouse sobre la cuadrícula correspondiente y darle un click.

Cuando se ingrese alguna letra, ya sea por presionar con el mouse o porque se presionó en el teclado podremos observar cómo se realiza un cambio de color para que el usuario identifique que letra se va a enviar, como se ejemplifica en la Figura 10.

Sin embargo, si la opción que seleccionó el usuario en el menú principal fue la alternativa 2, se muestra una interfaz cuya imagen está determinada por la captura de la cámara en tiempo real, con la adición de cuadrículas para saber hacia dónde dirigir nuestra mano y el robot responda realizando los mismos movimientos, como en la Figura 11.

Si el usuario se encuentra dentro de alguna de las opciones del programa y desea regresar al menú principal

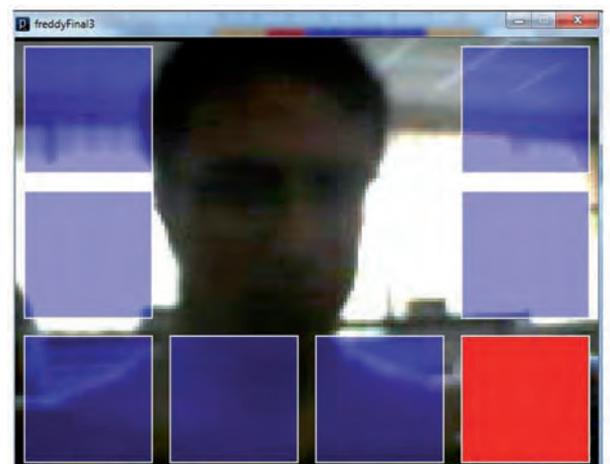


Figura 11: Interfaz para la manipulación por cámara web.

Velocidad de transmisión (baudios)	Tiempo de respuesta usando mouse o teclado (ms)	Tiempo de respuesta usando webcam (ms)
1200	3.8	5.5
2400	3.1	5.3
4800	2.3	3.9
9600	1.5	2.4
19200	<1	1.3

Tabla 5: Tiempos de respuesta al enviar instrucciones.

	Ángulo deseado	Ángulo obtenido	% Diferencia
Mouse o teclado	25°	20-22°	80 - 88 %
	45°	38-41°	84 - 91 %
Webcam	25°	18-20°	72 - 80 %
	45°	36-40°	80 - 88 %

Tabla 6: Comparación entre valores de ángulos esperados y obtenidos.

bastará con presionar alguna tecla distinta de las que se usa para enviar instrucciones al robot en el caso de la interfaz 1, y en el caso de la interfaz 2 puede ser cualquier tecla.

Como se definió en la Tabla 2, cada par de letras controla los movimientos de un motor, una letra provoca que el motor gire en sentido contrario a las manecillas del reloj mientras que otra letra hace que el motor gire en sentido horario.

Conforme se envían las letras “F”, “T”, “G” y “V”, puede observarse que la señal PWM correspondiente empieza a disminuir su ancho de pulso hasta llegar a un valor mínimo de 1ms de tiempo en alto (TA), expresado en (10), que permitirá al servomotor colocarse en su posición más extrema hacia la izquierda. En la Figura 12 se muestra la señal PWM especificada.

$$TA \cong 1ms + (2 * 0) us = 1ms + (2 * (0)) us = 1ms + 0us = 1ms \tag{10}$$

Si selecciona las letras “J”, “Y”, “H” y “B” se puede ver como el ancho de la señal PWM aumenta hasta llegar a un valor máximo de aproximadamente 2ms de TA,

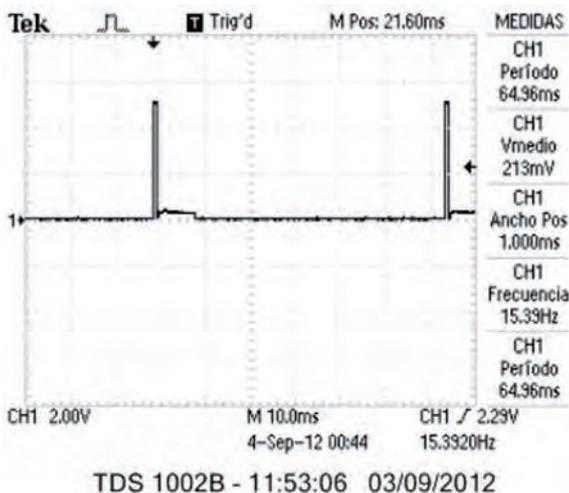


Figura 12: Gráfica de señal PWM para uno de los servomotores con ancho de pulso de 1ms.

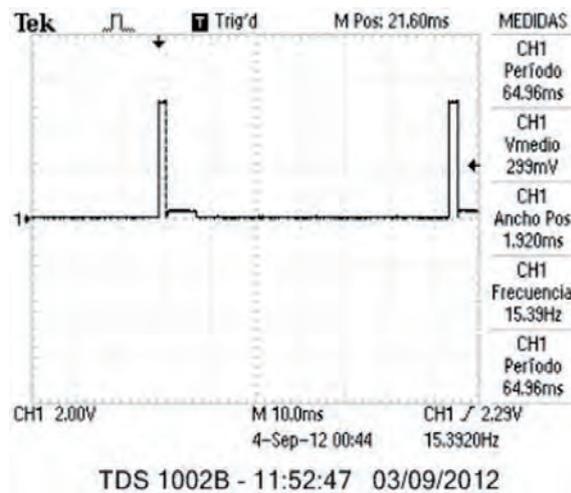


Figura 13: Gráfica de señal PWM para uno de los servomotores con ancho de pulso de 2ms.

expresado en (11), el servomotor tomará como referencia para ubicarse en su posición más extrema hacia la derecha. En la Figura 13 se muestra la señal PWM especificada.

$$TA \cong 1ms + (8 * (100)) us = 1ms + 800us = 1ms + 0,8ms = 1,8ms \tag{11}$$

Para poder seleccionar la tasa de baudios a transmitir se realizan pruebas a distintas velocidades permitiendo medir el tiempo de respuesta entre el envío de la instrucción y la reacción del lado de recepción ante la señal enviada. En la Tabla 5 se muestran los resultados de estas pruebas.

Al revisar los resultados de las pruebas se optó por transmitir a una tasa de 9600 baudios porque ofrecen tiempos pequeños de respuesta, aunque si se desea mejorar o disminuir estos tiempos se puede aumentar la velocidad siempre y cuando los elementos y las configuraciones así lo permitan. En este proyecto el microcontrolador sólo se podía configurar hasta un máximo de 19200 baudios.

Un factor que incidió en la precisión de los movimientos del brazo robótico cuando se empleó la opción de la cámara web fue el computador utilizado. Siempre el procesamiento de imágenes requiere de muchos recursos y el computador utilizado no poseía características óptimas para estas funciones, por ello se tuvo que disminuir la resolución de la pantalla teniendo como consecuencia que los movimientos no fueran tan precisos como podían ser cuando se ejecutaba instrucciones con el teclado o mouse. En la Tabla 6 se presentan las diferencias entre las posiciones a las cuales se deseaba ubicar alguno de los motores y la posición final conseguida.

Es claro que cuando se emplea la cámara web se obtuvo menos precisión en los movimientos como consecuencia del procesamiento de las imágenes, sin embargo se aprecia que el hecho de emplear las funciones del mouse o teclado tampoco permitieron llegar al 100 % de precisión.

Otro factor que influyó en ambas opciones, tanto en operación con webcam como con teclado, fue el diseño del robot. El prototipo desarrollado tenía un sistema de poleas para cada articulación o grado de libertad y al momento de los movimientos que realizaba el brazo las cuerdas obstaculizaban el correcto posicionamiento de los motores en muchas ocasiones viéndose reflejado en la precisión de los movimientos.

Conclusiones

Este trabajo puede ser la base del desarrollo de muchas aplicaciones futuras en cualquier ámbito de la ingeniería en donde se requiera la detección de movimiento, creación de interfaces inteligentes en donde el usuario necesite interactuar con dispositivos remotos.

Los módulos por defecto traen la configuración en el canal "OX0C" pero al encontrar fuentes de interferencia se debe modificar este parámetro hacia otros canales probando si se elimina la interferencia y por tanto quedar en el canal donde se recibe las instrucciones sin problemas y garantizar la confiabilidad en la comunicación.

El diseño del brazo robótico debe ser escogido con la precaución que permita libertad de movimientos además no se dañe ningún componente, en especial los motores, durante la operación. El prototipo empleado tuvo inconvenientes en su diseño y por este motivo se vio comprometida la exactitud, a pesar de aquello, tomando como base los resultados en la Tabla 6 se obtuvo que con teclado o mouse los movimientos del brazo correspondían a un 80-90 % de precisión y con la cámara web el porcentaje decaía a un 70-80 %, que sin embargo sigue siendo un buen resultado.

El presente proyecto ha sido diseñado para poder realizar comunicación inalámbrica desde el computador hacia el controlador con ayuda de módulos Xbee que implementan el protocolo ZigBee. La versión de módulos empleados en el proyecto fue la versión 1. Dicha versión sólo permite crear una conexión punto a punto. La siguiente versión de estos módulos, versión 2, permite conformar una red de sensores en la cual podemos realizar comunicación de tipo broadcast y multicast entre varios dispositivos, o en nuestro caso particular ejercer el control sobre distintos brazos robóticos al mismo tiempo.

Referencias

- [1] Barrientos, A.; Balaguer, C.; Aracil, R.; Peñín, L. 2007. "Fundamentos de Robótica". *McGraw-Hill*.
- [2] Martínez, G.; Jáquez, S.; Rivera, J.; Sandoval, R. 2008. "Diseño propio y Construcción de un Brazo Robótico de 5 GDL". *Revista de Ingeniería Eléctrica, Electrónica y Computación*. ISSN 1870-9532, 4(1).
- [3] Casey, R.; Fry, B. 2010. "Getting Started with Processing". *O'Reilly Media*.
- [4] Tutorial de Processing. www.processing.org/reference/libraries/.
- [5] Guía del usuario XBEE Serie 1. www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia-Usuario.pdf
- [6] Fabricación de brazo robótico. <https://sites.google.com/site/proyectosroboticos/fabricar-brazo-robot>.