

## CÓMPUTO MÓVIL PARA EL CONTROL Y SUPERVISIÓN DE UNA TARJETA ARDUINO UTILIZANDO UNA COMUNICACIÓN WI-FI

Arturo Juárez Ríos, Isis Tatiana Galván García, Emilio Onésimo Alvarado Pérez, Hind Taud, Juan Carlos Herrera  
Lozada

arturo.juarez@hotmail.com, isis.sist@gmail.com, emilio.alvarado@live.com.mx, htaud@ipn.mx, jlozada@ipn.  
mx

Centro de Innovación y Desarrollo Tecnológico en Cómputo Instituto Politécnico Nacional

### Resumen

*Este trabajo muestra cómo controlar un móvil con ruedas de manera remota utilizando una red Wi-Fi mediante el uso de un sistema Arduino, teniendo como unidad primaria de control una aplicación que se ejecuta en un sistema de cómputo móvil con sistema operativo Android. Se desarrolla un sistema de comunicación de red inalámbrica Wi-Fi Cliente-Servidor, el proceso de conexión entre dos sistemas diferentes, por una parte tenemos un móvil con ruedas con tracción diferencial al que se le conecta un sistema embebido con base en un microcontrolador, en este caso una tarjeta Arduino, programada con una aplicación para el control de los motores, sondeo de sensores de distancia y como Servidor de conexión de red Wi-Fi, y por otra parte un sistema de cómputo móvil con sistema operativo Android como lo son las Tablets o los Smartphones incluyendo una aplicación desarrollada para funcionar como Cliente en la comunicación de red inalámbrica, enviar la información de control de los motores y recibir la información correspondiente al monitoreo de los sensores remotos.*

## I. Introducción

Uno de los principales tópicos en el uso de las máquinas o robots es el poder monitorearlos y controlarlos de manera remota desde cualquier punto del planeta para que puedan desarrollar el trabajo o la tarea para la cual fueron diseñados o creados. El uso del control remoto se ha vuelto un ejercicio cotidiano en muchas de las actividades que realizamos comúnmente a nivel hogar o a nivel industrial, ya sea por comodidad para controlar una televisión, abrir la puerta de la cochera, controlar la iluminación, ya sea por seguridad como enviar robots para explorar áreas donde pueda estar en riesgo la vida o salud de un ser humano o sea inaccesible a él, por ahorro de tiempo y/o esfuerzo, etc. Para muchos sistemas robóticos que tienen una comunicación con su unidad de control primaria conectada mediante cableado, sería importante el sustituirlo por un sistema inalámbrico, por lo que el presente trabajo busca una alternativa que pueda ser implementada en este tipo de sistemas o en el desarrollo de sistemas nuevos.

Los medios de comunicación inalámbricas pueden ser por microondas, radiofrecuencias, Bluetooth, etc., cada uno de ellos tiene sus ventajas y desventajas dependiendo del uso que se les pueda dar a los sistemas que lo contienen. En este trabajo se busca la comunicación Wi-Fi ya que un equipo conectado por este medio es factible de conectarse y tener su acceso con un sistema de cómputo que se encuentre en cualquier parte del mundo donde haya internet.

Los sockets son utilizados para intercambiar información entre dos programas a través de internet o de alguna red local ya sea alámbrica o inalámbrica. Mediante esta interfaz se definen las reglas que los programas deben seguir para usar los servicios del nivel de transporte en una red TCP/IP [1].

## II. Desarrollo

### 2.1. Descripción General del Sistema

En la figura 1 se muestran de manera general los elementos del sistema.

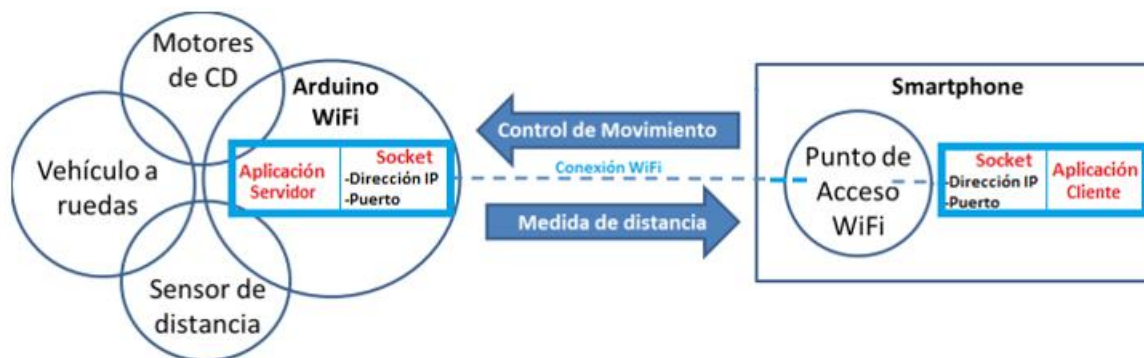


Fig. 1 Diagrama del Sistema.

El sistema consta de un vehículo móvil que tiene en su parte trasera 2 motorreductores de corriente continua (CD), trabajando de manera aislada, conectados a las llantas traseras del vehículo. Como etapa de potencia que alimenta a los dos motores se utiliza el circuito integrado L293D [2]. Se adapta también en el vehículo un sensor de distancia infrarrojo [3]. El sensor está diseñado de acuerdo a las especificaciones técnicas para medir distancias entre 10 y 80 cm. Sobre el vehículo se coloca un escudo Arduino Wi-Fi shield teniendo de base una tarjeta Arduino Uno. En el escudo Arduino se utilizan cuatro de sus entradas digitales para el control de movimiento de los motores a través del circuito de potencia y una de sus entradas analógicas para conectar la salida del sensor infrarrojo de distancia. El escudo Arduino se programa para conectarse a una red Wi-Fi a través de un punto de acceso, se almacena en ella una aplicación que trabaja como Servidor, como alimentador de información para los motores y distribuidor de la medida otorgada por el sensor infrarrojo. Se aprovecha el uso del Smartphone con la funcionalidad de punto de acceso, además de configurarlo como Cliente a través de una aplicación desarrollada sobre el sistema operativo Android, para lograr la conectividad y comunicación Wi-Fi entre ellos.

El sistema de comunicación se basa en una estructura Cliente/Servidor. El Servidor centraliza el servicio y lo ofrece a través de una dirección conocida (dirección IP y número de puerto). El Cliente controla la interacción del Servidor con el usuario. La figura 2 muestra el funcionamiento de esta estructura.



Fig. 2. Estructura Cliente/Servidor.

El Cliente y el Servidor se encuentran conectados por Wi-Fi a través de un módem o punto de acceso, la aplicación del Servidor espera todo el tiempo hasta que un Cliente se conecte.

## 2.2. Aplicación Server Arduino

Se le asigna un nombre para el SSID (Service Set Identifier) y una contraseña necesarios para conectarse a una red SSID con seguridad WPA2 PSK predefinida dentro de los parámetros de configuración del punto de acceso (Access Point) en el Smartphone utilizado, como se muestra en la figura 3.



Fig. 3. Configuración del Punto de Acceso en el Smartphone.

Se eligen y configuran los pines digitales 3,5,6 y 9 en el escudo Arduino como salidas de señales para controlar los motores de CD. Se elige el puerto 2000 para configurar el socket utilizado en la comunicación (*WiFiServer server(2000);*). El método *printWiFiStatus* nos muestra algunos parámetros de la conexión realizada del escudo Arduino al punto de acceso del Smartphone, entre ellos la dirección IP que se le

asigna a la tarjeta. La figura 4 muestra los datos obtenidos después de conectarse a la red, visualizados por el monitor serial del software de desarrollo de Arduino.

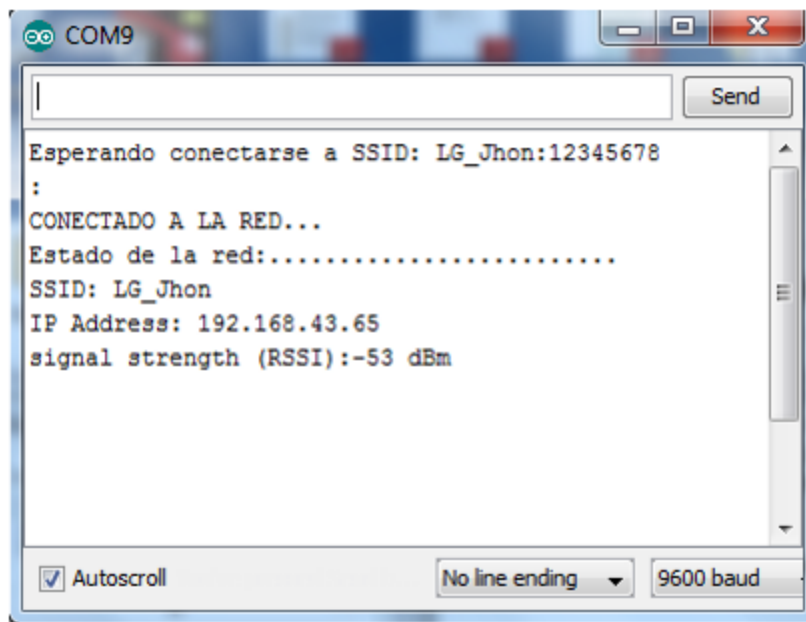


Fig. 4. Parámetros obtenidos visualizados en el monitor serial.

Esta dirección IP junto con el número de puerto predefinido (2000) son utilizados en la aplicación Android. Después de la conexión, la aplicación Arduino queda en espera a que se conecte un Cliente (*client.connected()=true*) y que empiece a enviar datos (*client.available()=true*). Estos datos serán leídos carácter por carácter (*client.read()*), si el carácter leído es un '2','8','4','6' o un '5' (avanzar, retroceder, girar a la izquierda, girar a la derecha o parar respectivamente), se le proporcionará a las 4 salidas los valores necesarios para controlar los motores. Se utiliza el comando *analogWrite(pin, valor)* para dar una señal de salida modulada por ancho de pulsos (PWM) con valores de 0 a 255 con lo cual podemos manejar la velocidad de los motores [4]. Cuando la aplicación server detecta un salto de línea ('\n'), significa que ya terminó el primer dato de entrada y ahora enviará al Cliente el valor leído del sensor infrarrojo (*client.println(medida);*). Si el Cliente sigue conectado, el Servidor continúa leyendo caracteres y escribiendo la medida del sensor. El código de la aplicación Arduino puede descargarse mediante la liga:

<https://www.dropbox.com//NbvjumERzif0rlcST1Wo8e?>

### 2.3. Aplicación Android

Esta es una aplicación Cliente. Se define el área donde se muestra la interface gráfica para controlar al vehículo motorizado y el dato de la distancia medida por el sensor infrarrojo, para esto se utiliza el método *onDraw*, definiendo un objeto *canvas* de la clase *Canvas* que será el "lienzo" donde se colocarán todos los gráficos [5]. Se define el método *onTouchEvent* que monitoreará cuando se toque la pantalla del Smartphone, obteniendo la posición (x,y) con la cual se puede saber cuál fué el "botón" presionado: 2 adelante, 4 izquierda, 5 detener, 6 derecha u 8 atrás. De acuerdo al botón presionado se muestra una imagen de un vehículo que indica el sentido de movimiento seleccionado, como se muestra en la figura 5. En un recuadro se muestra la distancia del sensor infrarrojo a un obstáculo colocado enfrente del vehículo.

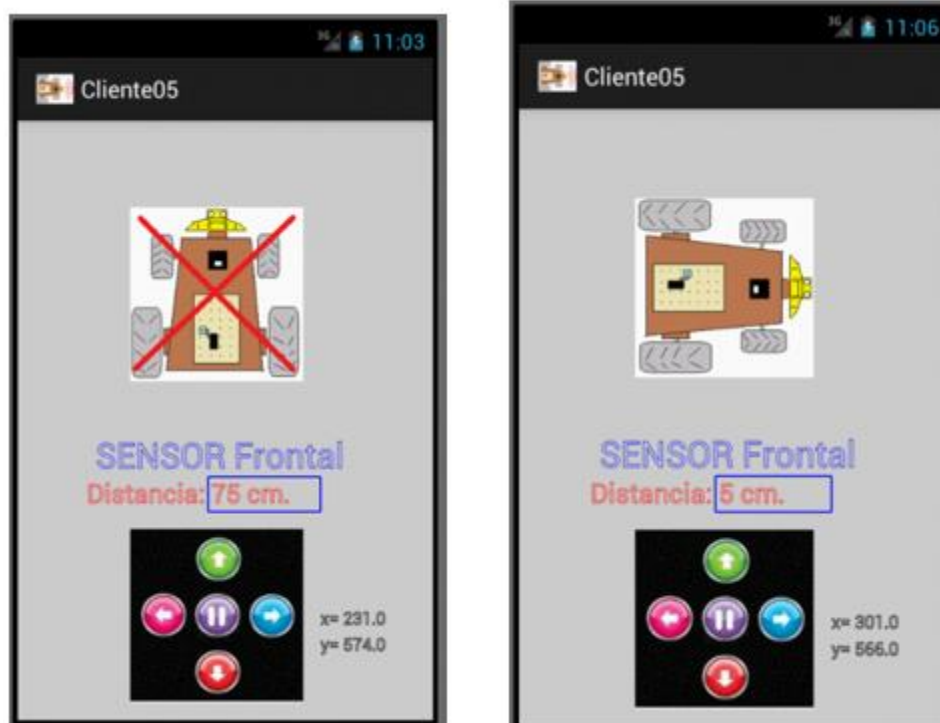


Fig. 5. Interface gráfica en el Smartphone, mostrando el vehículo detenido y girando a la derecha.

El número de botón presionado es enviado como parámetro de entrada para el método *ejecutaCliente*, quien devuelve en la variable "texto", precisamente el valor de la medición del sensor leído desde el Servidor. El método *ejecutaCliente* solicita y abre la conexión con la aplicación del Servidor utilizando una instancia *sk* (*Socket(ip,puerto)*);. Se crea una instancia para salida o envío de datos a Arduino y otra instancia para entrada o recepción de datos desde Arduino:

```
PrintWriter salida=new PrintWriter(new OutputStreamWriter(sk.getOutputStream()),true);  
BufferedReader entrada = new BufferedReader(new InputStreamReader(sk.getInputStream()));
```

Para enviar datos al arduino: *salida.println(datos)*;  
Para recibir información desde la conexión Arduino: *medida=entrada.readLine()*;

Después del envío y recepción de datos, la conexión al escudo Arduino es cerrada para que otro botón sea pulsado y así se repita el proceso. El código de la aplicación Android puede descargarse mediante la liga:

<https://www.dropbox.com/11oVnPwOuoO18LWxyX5pHfe?>

### III. Resultados

Se probó utilizando un Smartphone con sistema operativo Android 4.1.2, utilizándolo como punto de acceso y Cliente. La conexión se realizó exitosamente, las 5 diferentes órdenes de movimiento enviadas con el touch screen fueron recibidas y ejecutadas en el sistema Arduino. La medida de la distancia enviada por el sistema Arduino fue recibida y visualizada en el Smartphone. Esta medida requiere ser ajustada en la aplicación Android o Arduino para resultar más precisa. Es necesario pulsar el botón de paro para poder detener el movimiento del vehículo. La figura 6 muestra la foto del sistema.



**Fig. 6.** Sistema completo funcionando.

Se instaló la aplicación también en una Tablet de 2 procesadores y sistema operativo Android 4.0.4 y en otro Smartphone con sistema operativo Android 2.3.6, estos dos últimos conectados al punto de acceso del primer Smartphone realizando una red de 4 dispositivos. Desde los dos Smartphones y desde la Tablet se pudo realizar la conexión y la comunicación con el sistema Arduino de manera concurrente.

#### **IV. Conclusiones**

Se logró la comunicación inalámbrica Wi-Fi entre el escudo Arduino y un dispositivo de cómputo móvil, desarrollando e instalando una aplicación Servidor en el escudo Wi-Fi Arduino y otra aplicación Cliente en un dispositivo con sistema operativo Android, controlando el movimiento de un vehículo y monitoreando la medida de un sensor de distancia. Este proyecto servirá de base para otros proyectos que requieran monitorear o controlar un sistema remoto de manera inalámbrica a través del acceso a una red local o internet.

#### **Referencias**

1. Jesús Tomás Gironés, El gran libro de Android. Alfaomega Grupo Editor, páginas 349-359, México, Marzo 2013.
2. L293, L293D CUADRUPLE HALF-H DRIVERS, Texas Instrument, Reference Code SLRS008C September 1986- Revised November 2004.
3. GP2Y0A21YK Optoelectronic Device, Sharp Corporation, Reference Code SMA05008, 2005.

4. <http://www.arduino.cc/es/>

Consultada el 10 de Marzo del 2014.

5. José Enrique Amaro Soriano, Android. Programación de dispositivos móviles a través de ejemplos.

Alfaomega Grupo Editor, páginas 70-120, España, Febrero 2012.