

CONTROL Y SONDEO REMOTO DE UN MÓVIL A RUEDAS DESDE UN EXPLORADOR WEB UTILIZANDO UN ESCUDO WIFI ARDUINO.

Juárez, Arturo; Orozco, Roberto

Arturo.juarez@ hotmail.com; raov15@hotmail.com

Centro de Innovación y Desarrollo Tecnológico en Cómputo

Instituto Politécnico Nacional

Abstract—El presente trabajo muestra el uso del escudo wifi de Arduino para la comunicación inalámbrica entre un vehículo motorizado a ruedas y un equipo de cómputo, mediante un explorador de internet. Se utiliza el protocolo http para poder acceder como cliente a la tarjeta arduino y poder leer las mediciones de un sensor de distancia, así como controlar el movimiento del vehículo motorizado. Como equipo de cómputo se utiliza un smartphone que puede tener las 2 características, de cliente y de punto de acceso para crear una red inalámbrica local. En el sistema cliente-servidor, el escudo wifi Arduino es programado como servidor y en él se almacena la página web en código html, que será visualizada por el cliente que se conecte. Este cliente que se conecta, no requiere de software adicional, más que la capacidad para conectarse a la red wifi y tener un explorador de red.

I. INTRODUCCIÓN

El ser humano es capaz de recorrer grandes distancias por mar, tierra y espacio, puede cargar y desplazar objetos muy pesados, además de poder controlar objetos o situaciones, ya no sólo de manera física, sino utilizando diferentes máquinas o sistemas que permiten una interacción a distancia, inclusive fuera del espacio terrestre, en otros mundos o galaxias.

El control sobre las máquinas ha sido un tema primordial en el proceso de transformación del medio y al crear o desarrollar nuevos sistemas que ayuden o hagan más fáciles las tareas humanas cotidianas o no.

Las máquinas creadas son enviadas para obtener todo tipo de información, a lugares que pueden ser peligrosas, lejanas o inaccesibles al ser humano, por lo que se buscan sistemas de comunicación y de control remoto, para que desde cualquier lugar se pueda conocer, lo que se desea conocer del lugar alejado o inaccesible, sin estar personalmente ahí.

El trabajo aquí desarrollado tienen que ver con el uso de sensores, su manipulación para obtener información del ambiente que rodea al dispositivo que los porta, se buscó una comunicación inalámbrica y el uso de dispositivos de cómputo como unidades primarias de control, tales como el uso de la telefonía móvil.

II. DESARROLLO.

2.1. Descripción General del Sistema

En la figura 1 se muestra un diagrama del sistema desarrollado.

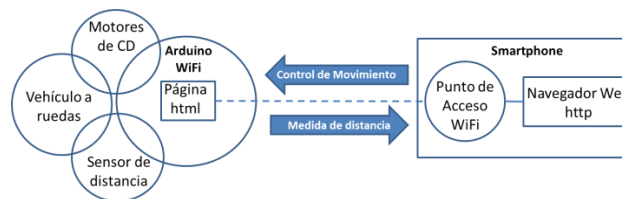


Fig. 1 Diagrama del Sistema

El sistema consta de un vehículo móvil que tiene en su parte trasera 2 motorreductores de corriente continua (CD), trabajando de manera aislada, cada uno de ellos conectado a una rueda (llanta). Como etapa de potencia que alimenta a los dos motores se utilizó el circuito integrado L293D [1]. Se adaptó también en el vehículo un sensor de distancia infrarojo [2]. Sobre el vehículo se colocó un sistema de Arduino wifi shield teniendo de base una tarjeta Arduino Uno. En el sistema Arduino se utilizan cuatro de sus salidas analógicas para el control de movimiento de los motores a través del circuito de potencia y una de sus entradas digitales como entrada para el medidor de distancia infrarojo. El sistema Arduino se programó para conectarse a una red wifi con información conocida, funciona como servidor web, se almacenó una página web en código html, que sirve como interface entre un cliente conectado y la misma tarjeta Arduino, tanto para controlar el movimiento de los motores como para mostrar las mediciones dadas por el sensor de distancia infrarojo. Se aprovechó el uso del smartphone con la funcionalidad de punto de acceso (Access point) para lograr la conectividad wifi y con la funcionalidad de cliente, para acceder al sistema Arduino desde el explorador web. La figura 2 muestra los elementos utilizados.

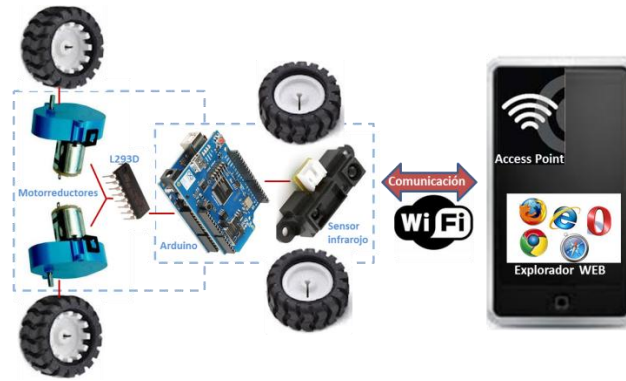


Fig. 2. Elementos Generales del Sistema

2.2. Preparación del sistema Arduino.

El escudo wifi de Arduino provee conectividad wireless en el estándar IEEE 802.11 b/g a través del módulo HDG104. La pila TCP/IP y las API para las bibliotecas wifi son manejadas en el escudo por el microcontrolador AT32UC3A1256. Ambos módulos tienen un firmware que debe ser actualizado. Inicialmente utilicé el software (versión 1.0.5) para la tarjeta Arduino que se descarga directamente de su página [3], lo instalé en mi laptop con Windows 2007, pero el sistema Arduino tiene problemas de comunicación. Después de varias pruebas, la solución fue instalar el IDE de Arduino Nightly y actualizar el firmware de la tarjeta, con esta versión. El firmware para el módulo HDG104 es nombrado "wifi_dnld.elf" y para el módulo AT32UC3 es "wifiHD.elf". El software IDE se puede bajar de la página de Arduino [4]. Para la instalación sólo se descomprime el archivo descargado y se copia la carpeta "arduino-nightly" en algún lugar del disco duro, yo por ejemplo lo copie dentro de "mis documentos" dentro de una carpeta llamada "software", el archivo ejecutable del IDE y los archivos de actualización de firmware se encuentran dentro. Para realizar la tarea de actualización se requiere instalar adicionalmente el software Flip 3.4.7, que puede ser bajado de la página de Atmel [5]. Ya instalado, mediante un cable usb, se conecta a la computadora el escudo wifi arduino (canectada a su vez a la tarjeta Arduino Uno). El jumper j3 del escudo wifi debe ser conectado físicamente para ponerla en modo de programación y posteriormente se abre una ventana de MSDOS (símbolo del Sistema), colocándose en la ruta donde se instaló el software Flip, en mi caso:

```
C:\Program Files\Atmel\FliP 3.4.7\bin>
```

Desde ahí se teclea el comando de actualización del firmware, colocando algunos parámetros de control y la ruta completa donde se encuentren los archivos *.elf. En mi caso para descargar el firmware al escudo wifi en el módulo HDG104 fué:

```
batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f memory flash  
blankcheck loadbuffer C:\Users\Q0RGD068\Documents\software\arduino-
```

```
nightly\hardware\arduino\avr\firmwares\wifishield\wifi_dnld\Release\wifi_dnld.elf  
program verify start reset 0
```

y posteriormente para descargar el firmware al escudo wifi en AT32UC3A1256:

```
batchisp.exe -device AT32UC3A1256 -hardware usb -operation erase f memory flash  
blankcheck loadbuffer C:\Users\Q0RGD068\Documents\software\arduino-  
nightly\hardware\arduino\avr\firmwares\wifishield\wifiHD\Release\wifi_HD.elf  
program verify start reset 0
```

Posterior a la actualización, se libera el jumper j3 para sacar a la tarjeta del modo de programación. Mediante el software IDE se carga la aplicación.

2.3. Aplicación Arduino.

Desde el IDE Arduino nightly se crea la aplicación que será instalada en el escudo wifi Arduino. La figura 3 muestra la ventana del IDE. En ella se muestra parte del código, así como el puerto (COM7) que se utilizará para cargar el software a través de una conexión por cable usb.

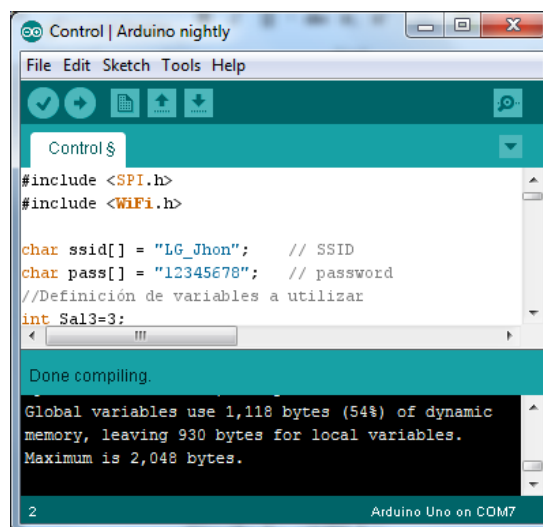


Fig. 3. Ventana de software de desarrollo IDE

El código de la aplicación a instalarse es el siguiente:

```
// Bibliotecas para la comunicación inalámbrica  
#include <SPI.h>  
#include <WiFi.h>  
//Nombre y password de la red a conectarse  
char ssid[] = "LG_Jhon"; // SSID  
char pass[] = "12345678"; // password
```

```
//Definición de variables a utilizar
int Sal3=3;
int Sal5=5;
int Sal6=6;
int Sal9=9;
int dist;
int status = WL_IDLE_STATUS;
WiFiServer server(80);
//Ciclo repetitivo:
void setup()
{
  //Pines digitales 3,5,6 y 9 del escudo wifi
  pinMode(Sal3, OUTPUT);
  pinMode(Sal5, OUTPUT);
  pinMode(Sal6, OUTPUT);
  pinMode(Sal9, OUTPUT);
  //Inicializa y espera a que el puerto abra:
  Serial.begin(9600);
  // Checa la presencia del escudo wifi:
  if (WiFi.status() == WL_NO_SHIELD)
  {
    Serial.println("WiFi shield No esta presente");
    // No continua
    while(true);
  }
  // Intenta conectarse a la red WiFi:
  while ( status != WL_CONNECTED)
  {
    Serial.print("Intentando conectarse a SSID: ");
    Serial.print(ssid);
    Serial.print(":");
    Serial.println(pass);
    status = WiFi.begin(ssid, pass);
    delay(1000);
  }
  server.begin();
  Serial.println(":");
  Serial.println("CONECTADO A LA RED...");
  // Está conectado e imprime el status:
  printWifiStatus();
}

void loop()
{
  // Escucha para la conexión de clientes
  WiFiClient client = server.available();
  int count=0;
  if (client)
  {
    Serial.println("nuevo cliente");
    // Un requerimiento http finaliza con una línea en blanco
    boolean currentLineIsBlank = true;
    while (client.connected())
    {
      if (client.available())
      {
        char c;
        int b;
        while((b = client.read()) != -1);

```

```
c = b;
// Si se ha llegado al final de la línea (recibe un
// carácter de nueva línea "\n") y la línea está en
// blanco, la petición http ha terminado,
// así que se puede enviar una respuesta
if (c == '\n' && currentLineIsBlank)
{
    count = 0;
    //Codigo html para página web
    client.println("<html><head><title>Control
wifi</title></head><body>");
    client.println("<center><h1> AUTO</h1>");
    //Configura pin 5 como entrada analógica
    int sensorReading = analogRead(5);
    client.println(" ");
    client.print("La Medida es:");
    dist=sensorReading;
    client.print(dist);
    client.print(" cm. ");
    client.println(" ");
    client.println("<br />");
    client.println("<p><form method='get'
action='\"><select name='I'>");
    client.println("<option value='0'>
DETENER.</option>");
    client.println("<option value='1'>ADELANTE
</option>");
    client.println("<option value='2'>ATRÁS
</option>");
    client.println("<option value='3'>IZQUIERDA
</option>");
    client.println("<option value='4'>DERECHA
</option>");
    client.println("</select><input type='submit'
value='C A M B I A' /></form></p>");
    client.println("</center></body>");
    client.println("</html>");
    client.println("");
    client.flush();
    break;
}
if (c == '\n')
{
    // Se está iniciando una nueva línea
    currentLineIsBlank = true;
    count = 0;
}
else if (c != '\r')
{
    // Ha llegado un caracter a la linea actual
    currentLineIsBlank = false;
    count++;
    if ((c == '=') && (count == 8))
    {
        while((b = client.read()) == -1);
        c = b;
        if (c == '0')
        {
            digitalWrite(Sal3, LOW);
        }
    }
}
```



```
Serial.print(rssi);  
Serial.println(" dBm");  
}
```

2.4. Etapa de potencia para los motorreductores.

El circuito integrado L293D que se ocupó, incluye cuatro circuitos para manejar cargas de potencia media [6], en especial pequeños motores, con la capacidad de controlar una corriente hasta 600 mA en cada circuito y un voltaje entre 4.5V y 36V. Con un circuito integrado se forman 2 puentes H completos, con los que se manejan los 2 motores. En este caso el manejo es bidireccional, con frenado rápido y con posibilidad de implementar el control de velocidad desde la tarjeta Arduino a través del manejo de las salidas con pwm (modulación por ancho de pulsos). La figura 4 muestra la conexión de los motores.

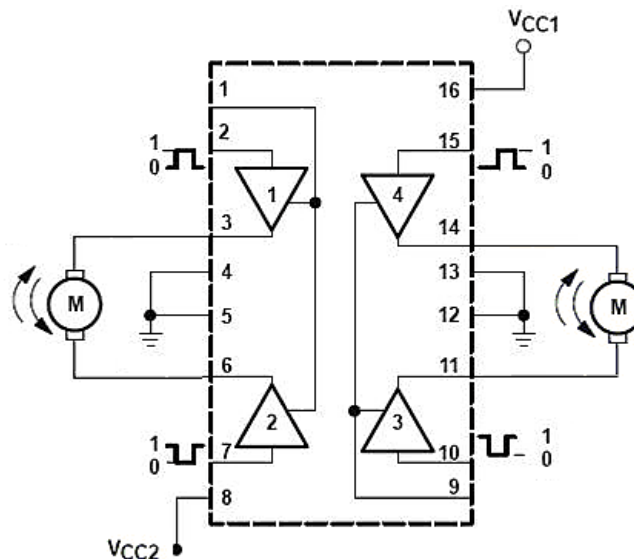


Fig. 4. Conexión de los motores en el circuito L293D.

Los pines 2 y 7 del L293D son conectadas a los pines de salidas digitales del Arduino 3 y 5, así como los pines 10 y 15 del L293D a los pines de salidas digitales 6 y 9.

2.5. Configuración y modo de acceso en el Smartphone.

Para este proyecto estoy utilizando un smartphone similar a los que actualmente existen con la opción de configurarse como punto de acceso (access point), el cual configuré y activé como sigue:

SSID de red: LG_Jhon

Seguridad WPA2 PSK

Contraseña: 12345678

Dentro del programa instalado en el Arduino podemos observar el código correspondiente a "WifiStatus()", con el uso del IDE en la opción de "Monitor serial", nos muestra los datos obtenidos de la tarjeta, entre otros la dirección ip que toma el escudo wifi Arduino al conectarse a la red inalámbrica, en mi caso tomó la dirección 192.168.43.65.

Mediante un led integrado en la tarjeta, el escudo wifi Arduino nos muestra el momento en que se conecta a la red. Ya conectado a la red, desde cualquier explorador web se accesa al arduino mediante la dirección ip asignada:

http://192.168.43.65/

III. RESULTADOS.

Desde el mismo smartphone, o desde cualquier computadora conectada a la red se puede acceder como cliente al control del vehículo a ruedas. En la figura 5 se muestra la ventana de acceso que se muestra en una computadora utilizando el explorador web.



Fig. 5. Ventana de acceso mediante Internet Explorer..

A través de esta ventana se muestra la medida proporcionada del sensor infrarojo de distancia. En realidad es un valor proporcional a la distancia que mide, por el momento no está ajustada dentro del programa, se debe introducir una fórmula que convierta al valor real.

Mediante un menú se puede elegir cualquiera de las 5 órdenes para el movimiento del vehículo a ruedas: "ADELANTE", "ATRÁS", "IZQUIERDA" "DERECHA" y "DETENER" como valor de default, y presionando el botón <Cambiar>. Se buscaron varias opciones

dentro del código html para crear la página y que fuera más amigable y de acceso más directo, pero el sistema está limitado a ciertos comandos, por lo que otras opciones no funcionaron. La respuesta del sistema está limitado a la actualización de la página web, por lo que después de elegir una orden de movimiento del vehículo, hay que esperar entre 2 y 4 segundos, dependiendo del sistema de cómputo, para poder volver a utilizar la página web.

IV. TRABAJO A FUTURO.

En un futuro se pretende ajustar la aplicación para controlar las velocidades en el manejo de los motores, ajustar la medida proporcionada por el sensor infrarojo de distancias. Definir una guía para el conjunto de comandos html que puede aceptar el sistema, para poder desarrollar una página más amigable y reducir los tiempos de respuesta. Definir claramente las limitaciones y alcances que puede tener el sistema Arduino para este tipo de aplicaciones. Desarrollar una aplicación en android para el control y monitoreo de este sistema.

V. CONCLUSIONES.

Se llegó al objetivo de realizar una comunicación bidireccional para envío y recepción de datos a través de un enlace wifi con el uso de un sistema Arduino. Del lado del cliente no es necesaria la instalación de software adicional, basta con la conexión a la red y el uso de algún explorador de red. El mismo smartphone funcionó como punto de acceso para crear la red inalámbrica y como cliente para el control y monitoreo del vehículo a ruedas. 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 de internet, para sistemas que no requieran una respuesta en tiempo real y puedan tolerar algunos segundos de retardo.

VI. REFERENCIAS.

- [1] L293, L293D CUADRUPLE HALF-H DRIVERS, Texas Instrument, Reference Code SLRS008C September 1986- Revised November 2004.
- [2] GP2Y0A21YK Optoelectronic Device, Sharp Corporation, Reference Code SMA05008, 2005.
- [3] <http://arduino.cc/es/Main/Software>
Consultada el 10 de Agosto del 2013.
- [4] <http://www.arduino.cc/en/Main/Software#toc4> Consultada el 12 de noviembre del 2013.
- [5] <http://www.atmel.com/tools/FLIP.aspx>
Consultada el 12 de noviembre del 2013.

[6] http://robots-argentina.com.ar/MotorCC_L293D.htm

Consultada el 6 de Octubre del 2013.

Autores:

Arturo Juárez Ríos, Ingeniero en Comunicaciones y Electrónica egresado de la ESIME Culhuacan en la Ciudad de México.

Roberto Orozco Velázquez, Ingeniero en Comunicaciones y Electrónica egresado de la ESIME Zacatenco en la Ciudad de México.

Ambos estudiando actualmente la maestría en Tecnología de Cómputo en el CIDETEC del IPN.