

CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR



Escuela de ingeniería

Redes de Computadoras
Diseño de Interfaces

Proyecto: hayCarrito

Presenta:

Rubén Guerra Marín - 15661
Daniel E. Moctezuma - 11851
Alexis Ortega Cisneros - 18493
Luis Pulido Díaz - 19365

Tijuana B. C., 06 de diciembre del 2011

Introducción

El presente trabajo es el resultado de los conocimientos adquiridos durante el semestre en la materia de redes de computadoras y en la que se vieron temas tales como las capas del modelo híbrido.

El adquirir los mencionados conocimientos y aplicarlos en la vida real, es el objetivo de toda materia de estudio, principalmente cuando se estudia ingeniería ya que se requiere que el ingeniero posea la capacidad de solucionar problemas.

En el presente reporte, se explican cada una de las etapas en el desarrollo para el proyecto “hayCarrito” que es en términos generales un carro controlado mediante microcontroladores, un dispositivo móvil y diversas tecnologías de tipo código abierto.

Cabe resaltar que los mencionados conocimientos sirvieron como base para la elaboración del proyecto que, como se explicará con más detalle en el reporte, tiene como principal objetivo el poder controlar un vehículo a través de las comunicación vía TCP/IP remoto.

Por otro lado, este tipo de proyectos son de gran ayuda para nosotros próximos ingenieros ya que lo que se obtiene al final es un gran aprendizaje y el gusto por saber de con tu dedicación e investigación puedes crear cosas útiles.

Para todo proyecto de fin práctico, existen momentos en los que surgen diversos problemas o inconvenientes en las que en base al ingenio y esfuerzo se puede llegar a buenas soluciones.

Cabe mencionar que para este proyecto, se organizó el grupo en dos equipos. El primer equipo que se encargó del mecánica y electrónica, el segundo equipo del área de la programación.

hayCarrito

Objetivo

Aplicar los conocimientos adquiridos a través del curso en la elaboración de un sistema que controle un vehículo a través de el protocolo TCP/IP.

Roles

- Rubén y Luis Pulido: parte mecánica y electrónica.
- Daniel y Alexis: parte del software.

Materiales

- Carrito de control remoto.
- Microcontrolador ATMEGA-128.
- Módulo Wifi.
- Protoboard.
- Transistores.
- Diodos.
- Resistencias.
- Cables.
- Soldadura.
- Teléfono con sistema operativo Android.

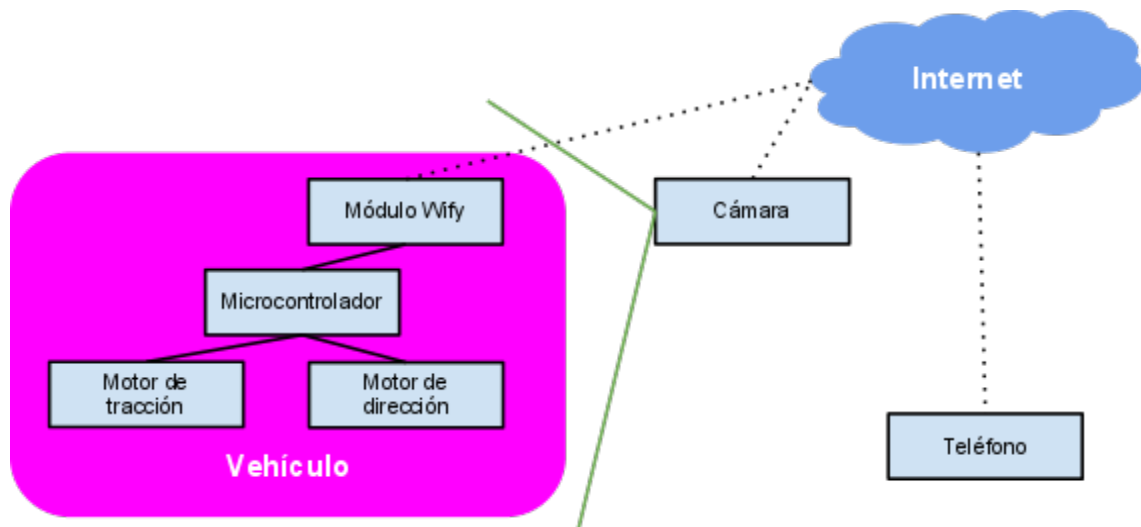
Especificaciones de hardware

El hardware utilizado es un microcontrolador ATMEGA-128 conectado a un módulo Wifi. El microcontrolador a la vez está conectado por medio de los puertos PWM a los motores del vehículo: un motor DC para la tracción de las llantas traseras, y otro motor DC para la dirección en las llantas delanteras.

Se utilizó también una cámara web, la cual está fija en un lugar apuntando al vehículo y conectada a Internet.

En el lado del cliente, el software que se utiliza para controlar el dispositivo se conecta tanto al módulo Wifi como a la cámara web.

Conectividad



Especificaciones de software

El software en el celular fue programado en el framework nativo de Android utilizando el lenguaje Java. Se utilizaron las funciones que controlan la lectura de los sensores de posición (acelerómetro y giroscopio), para controlar la dirección y la tracción del vehículo, así como los sockets necesarios para establecer la comunicación punto a punto vía TCP/IP.

El microcontrolador fue programado mediante un programador JTAG. Incluye las funciones para la comunicación con el módulo Wifi, el control de los motores por pulsos (PWM) y un socket para establecer la comunicación entre las partes.

Código

Android (java)

```
private boolean connect( String ip, int port )
{
    try
    {
        socket = new Socket( ip, port );
        in      = new BufferedReader( new InputStreamReader( socket.getInputStream() ) );
        out     = new BufferedWriter( new OutputStreamWriter( socket.getOutputStream() ) );

        StringBuilder sb = new StringBuilder();
        // Wait for a response and append it when an end of line is received.
        sb.append( in.readLine() );

        isConnected = true;
    }
    catch ( IOException e )
    {
        toastMessage( "Could not establish connection:\n" + e.getMessage() );
        isConnected = false;
    }

    return isConnected;
}
```

Microcontrolador (c)

```
#include "uart.h"
#include "pwm.h"
#include "acciones.h"
#include <avr/io.h>
#include <stdio.h>

int main(const int argc, const char **argv)
{
    DDRA |= (1 << 5); // Led bien
    DDRA |= (1 << 6); // Led error
    DDRA |= (1 << 7); // Led espera de socket
    PORTA |= (1 << 5);
    PORTA |= (1 << 6);
    PORTA |= (1 << 7);

    char serial[50];
    int id_socket;
    char estado[11];
    char pedir_socket[13];
    char accion;
    char velocidad[2];

    inicializar_uart();
    inicializar_pwm();
}
```

```

while (1)
{
    PORTA |= (1 << 6);
    escribir_uart("AT+"); // Ver que haya conexion
    leer_uart(serial);
    if (serial[1] == 'O' && serial[2] == 'K')
    {
        escribir_uart("AT+WJOIN"); // Conectarse a la red
        leer_uart(serial);
        if (serial[1] == 'O' && serial[2] == 'K')
        {
            escribir_uart("AT+SKCT=0,1,0,5000"); // Crear socket
            leer_uart(serial);
            if (serial[1] == 'O' && serial[2] == 'K')
            {
                id_socket = serial[4]+1; // Guardamos el numero del socket
                sprintf(estado, "AT+SKSTT=%c", id_socket);
                sprintf(pedir_socket, "AT+SKRCV=%c,4", id_socket);
                do
                {
                    escribir_uart(estado); // Checar el estado del socket.
                    leer_uart(serial);
                    if (serial[6] == '2') // Si ya están conectados al socket
                    {
                        PORTA |= (1 << 7);
                        PORTA &= ~(1 << 5);
                        escribir_uart(pedir_socket);
                        leer_uart(serial);
                        if (serial[1] == 'O' && serial[2] == 'K' && serial[4] != '0')
                        {
                            leer_socket(&accion, velocidad);
                            if (accion == 'F')
                                adelante(velocidad);
                            else if (accion == 'B')
                                atras(velocidad);
                            else if (accion == 'L')
                                izquierda(velocidad);
                            else if (accion == 'R')
                                derecha(velocidad);
                        }
                    }
                } while (serial[6] != '0');
            }
            else // Si aun no estan conectados al socket
            {
                PORTA |= (1 << 5);
                PORTA &= ~(1 << 7);
                OCR1A = 0;
                OCR3A = 0;
                OCR1B = 0;
                OCR3B = 0;
            }
        }
    }
    else
        PORTA &= ~(1 << 6);
}
else
    PORTA &= ~(1 << 6);
}
else
    PORTA &= ~(1 << 6);
}
else
    PORTA &= ~(1 << 6);
}

```

} }

Conclusión

Para finalizar, gracias a los conocimientos adquiridos a lo largo del curso, se obtuvieron las bases para construir este proyecto. Por otro lado, cabe señalar que se cumplió con el objetivo que se trazó y aunque el camino no fue fácil para lograrlo, se optó por tener perseverancia y ambición para llegar al resultado deseado.

Durante la realización del proyecto se presentaron algunos inconvenientes. Uno de ellos fue la compra de los aparatos, ya que tuvimos que ordenarlos y esperar a que los enviaran y esto ocasiono un atraso en el proyecto. Además otro de los obstáculos para la construcción del proyecto, más que nada en el aspecto económico; ya que como se sabe construir este tipo de proyectos a veces no es tan barato.

Cabe señalar que la elaboración de este proyecto, permitió un cierto trabajo colaborativo y cooperativo, ya que permitió desarrollar actitudes positivas hacia el trabajo interdisciplinario, cooperación, contribución y además que forma lazos entre compañeros.

Fue muy satisfactorio para nosotros como ingenieros en ciencias computacionales y en cibernética electrónica que se pudiera realizar un proyecto de este tipo; porque este proyecto está enfocado en gran parte al área de la electrónica. Es cierto que muchos de los integrantes nunca habían trabajado en esa área; sin embargo, a pesar de las dificultades se logró sacar adelante el proyecto y construir el proyecto.

Referencias

Android Developers. (s.f.). *Package Index*. Recuperado el 30 de noviembre del 2011, de <http://developer.android.com/reference/packages.html>

Android Developers. (s.f.). *User interface* Recuperado el 30 de noviembre del 2011, de <http://developer.android.com/guide/topics/ui/index.html>

AVR Freaks Forum. Recuperado el 25 de noviembre del 2011, de <http://avrfreaks.net>