



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEOPERACIÓN EN UN CARRO DE GOLF ELÉCTRICO

Fabián Arancibia Santti¹, Claudio Urrea Oñate²

Grupo de Automática. Departamento de Ingeniería Eléctrica. Universidad de Santiago de Chile.
Avda. Ecuador N° 3519. Estación Central. Santiago. Chile.

¹fabian.arancibia@usach.cl, ²claudio.urrea@usach.cl

RESUMEN

En este artículo se presentan algunos avances de un proyecto realizado por el Grupo *Smart Cars* del Departamento de Ingeniería Eléctrica de la Universidad de Santiago de Chile. El desarrollo del tema consiste en la implementación de un sistema de teleoperación para un carro de golf eléctrico. El sistema está constituido por una estación de operación y un controlador a bordo del carro. El primero está formado por un computador portátil, con una tarjeta de red inalámbrica y un *joystick* tipo volante, para simular el control de la dirección del vehículo. El segundo corresponde a un conjunto de dispositivos que permiten el control de los actuadores del carro y la comunicación con el operador. Para el control de los actuadores se utiliza una placa Arduino Mega y una interfaz electrónica, que interconectan un grupo de relés con las luces, intermitentes y bocina del carro. Para la comunicación inalámbrica, entre el carro y la estación, se utiliza un enrutador Wi-Fi, que está conectado a la placa Arduino a través de una interfaz de comunicación, llamada Ethernet Shield. Además, se han instalado un par de cámaras IP que permiten obtener realimentación visual del entorno del vehículo, cerrando el lazo de control junto al operador. Por otro lado, el computador portátil posee un *software* creado en Visual Basic 2005, diseñado para comunicarse a través de Ethernet, utilizando WinSock, y para manejar dispositivos periféricos como *joysticks*, a través de DirectX 9.

PALABRAS CLAVES: Teleoperación, vehículos eléctricos, Interfaz Gráfica de Usuario (GUI).

DESIGN AND IMPLEMENTATION OF A TELEOPERATION SYSTEM FOR AN ELECTRIC GOLF CAR

ABSTRACT

In this article we present some advancements in a project carried out by the Smart Cars Group of the Department of Electric Engineering of the Universidad de Santiago de Chile. The development consists in the implementation of a teleoperation system for an electric golf car. The system consists of an operation station and an on-board controller. The first one is composed by a portable computer with a wireless network card, and a "steering wheel" *joystick*, to simulate the directional control of the vehicle. The second part is composed by a set of devices that allow controlling the car's actuators and the communication with the operator. For the control of actuators we employed an Arduino Mega board and an electronic interface, aimed to interconnect a group of relays with the lights, the turn signals and the horn of the car. For wireless communications between the car and the station, we used a Wi-Fi router connected with the Arduino board through a communications interface, called Ethernet Shield. In addition, we installed a pair of IP cameras that allow visual feedback from the vehicle's surroundings, closing the control loop with the operator. In the other hand, the portable computer has software created in Visual Basic 2005, designed for communications through Ethernet, by using WinSock, and to manage peripheral devices like *joysticks*, through DirectX 9.

KEY WORDS: Teleoperation, electric vehicles, Graphical User Interface (GUI).



1. INTRODUCCIÓN

En general, los sistemas de teleoperación han basado su desarrollo en la seguridad del operador, ya sea porque no es posible manipular ciertos materiales debido a su peligrosidad o porque la presencia del operador es imposible debido a las características del medio. Por ejemplo, la manipulación directa de materiales radiactivos produce graves daños físicos y, por otro lado, aún existen lugares imposibles de alcanzar como las profundidades del océano o el vasto espacio exterior [1]. Sin embargo, tales razones no son suficientes para el uso de sistemas teleoperados, ya que algunos dispositivos podrían ser reemplazados por sistemas automatizados, que realizarían esas mismas tareas de manera más eficiente. La teleoperación se vuelve necesaria cuando las decisiones son complejas y sólo la inteligencia humana puede realizarlas. Uno de esos casos, que aún no ha sido resuelto por la automatización, es la conducción de automóviles. Por lo tanto, la orientación de este tema apunta a la necesidad de desarrollar sistemas de transporte más seguros, tanto para el operador como para los usuarios, por medio de sistemas automatizados de conducción. El desarrollo de la teleoperación en estos vehículos es el punto de partida para la creación de sistemas inteligentes que cumplan con las expectativas en esta área de trabajo.

El principal objetivo de este proyecto es implementar un sistema de teleoperación en un carro de golf eléctrico basado en un estudio técnico y económico realizado para tales fines. Por lo tanto, el desarrollo del tema está basado en el estudio de las actuales investigaciones relacionadas, las tecnologías necesarias para su construcción, los costos asociados a la implementación, el desarrollo de una interfaz gráfica de usuario y la integración de tal sistema en el carro. Lo anterior, tiene como propósito sentar las bases para la experimentación e implementación de sistemas de teleoperación que serán mejorados por las nuevas generaciones de alumnos de la carrera de ingeniería eléctrica de nuestra Universidad.

2. MODELO GENERAL DEL SISTEMA

A grandes rasgos, el sistema de teleoperación está constituido por tres bloques básicos, mostrados en la Fig. 1, que representan cada una de las partes involucradas en el proceso. El sistema posee una comunicación bilateral, con el fin de obtener realimentación informativa. Los bloques se han denominado como: el operador, el canal de comunicación y el objeto de control.

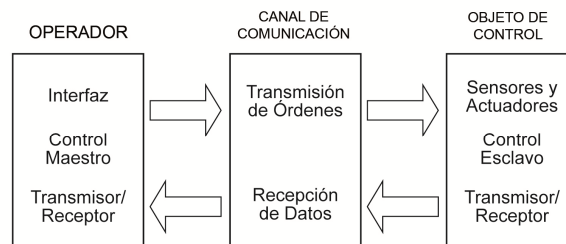


Figura 1: Diagrama de bloques del sistema

El operador

En este bloque, el operador accede al medio externo a través de una interfaz gráfica de usuario, donde se puede visualizar el entorno del vehículo, mediante un juego de cámaras a bordo del carro, y que por medio de un manipulador físico (como un *joystick*), es posible controlar los actuadores del vehículo. En esta etapa, es necesaria la utilización de un controlador donde puedan conectarse y coordinarse los elementos manipuladores, según las distintas acciones que realicen. Es importante que esta estación de control sea lo suficientemente robusta para que soporte la cantidad de información que reciba y que ésta pueda ser gestionada correctamente.



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



Finalmente, cada una de las señales de control debe ser enviada desde el controlador de operación hasta el controlador del carro, que se encargará de entregar las señales a los actuadores electromecánicos. Al originarse en esta etapa todas las órdenes, que serán ejecutadas en el bloque final, este controlador pasa a denominarse “controlador maestro”.

El canal de comunicación

El sistema de teleoperación necesita autonomía de movimiento, por lo tanto, el medio de comunicación debe ser inalámbrico. Es necesario saber cuáles son los rangos de operación óptimos y las limitaciones de cada una de las plataformas de comunicación ya existentes, que permitan el envío de señales de control y señales de video para la realimentación visual del operador.

Es preciso separar los canales en los cuales se envía o recibe información. Para la etapa de control no es necesario contar con un gran ancho de banda en comunicaciones, sino más bien, contar una vía rápida en donde la información viaje sin problemas y pueda ser ejecutada en tiempo real. Por otro lado, la señal de audio/video destinada a las cámaras requiere de un ancho de banda mayor.

El objeto de control

Éste corresponde al elemento que será controlado a distancia, que en este caso es el carro de golf eléctrico. Cada uno de los elementos que compone este bloque debe encontrarse a bordo del vehículo y, por consiguiente, requiere un sistema de alimentación independiente de la red doméstica. Luego, se debe utilizar baterías que alimenten a cada uno de los dispositivos a bordo, sin afectar la alimentación propia del vehículo.

El carro posee elementos mecánicos de actuación que fueron intervenidos con el fin de activarlos eléctricamente. Para ello, se han diseñado actuadores electromecánicos, los cuales están conectados a un controlador que recibe los comandos desde la estación de operación. Para el control del volante se ha diseñado e implementado una barra de dirección con un sistema de piñones y cadena, accionado mediante un motor de corriente continua. El giro del motor es controlado a través de un puente H y un controlador que recibe las señales enviadas por el operador. Este dispositivo, por estar sometido a las decisiones del controlador maestro, es llamado control esclavo, ya que no interfiere en las decisiones tomadas por el operador. Para el caso de los pedales de aceleración y frenado, éstos también han sido intervenidos con un juego de potenciómetros digitales que pueden ser manipulados desde el sistema controlador local.

A bordo del vehículo deben estar fijadas las cámaras que visualizarán el entorno y que enviarán las imágenes al operador, por medio del sistema de transmisión inalámbrica. Como se dijo anteriormente, éstas deben tener una batería autónoma que las alimente con el fin de que sigan funcionando a pesar de que el vehículo se quede sin energía eléctrica. Adicionalmente, para la monitorización del vehículo podrían leerse los estados de funcionamiento, como niveles de energía, fallas en los dispositivos, posición geográfica o proximidad de objetos que pueden entorpecer la marcha de éste, mediante una serie de sensores que estén conectados al controlador.

3. EVALUACIÓN TÉCNICA

El sistema de teleoperación requiere satisfacer las necesidades de los tres bloques que lo componen. Éstos pueden definirse en seis grupos de interés, de los cuales se pueden evaluar una serie de características adecuadas para la creación del sistema general.

Primero, un controlador maestro que posea gran capacidad de procesamiento y almacenamiento, para que en un futuro puedan quedar registrados los historiales de funcionamiento, errores, tiempo de duración de baterías, retardos de comunicación, etc. y un monitor que muestre la interfaz de usuario, entradas para conectar los periféricos de control y una conexión con algún dispositivo de comunicación.



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



Segundo, una interfaz gráfica de usuario que permita la interacción entre el operario y la máquina. En ella deben mostrarse las diferentes vistas de las cámaras, menús desplegables con los estados del vehículo y un panel de configuración para cada necesidad. Esta interfaz debe ser lo más sencilla posible, con el fin de que cualquier usuario pueda utilizarla sin mayor instrucción de operación.

Tercero, un manipulador que emule el funcionamiento del sistema original, para facilitar el uso al operador. Para ello, se deben buscar aparatos que simulen el uso de volantes, para la dirección, y de pedales, para la aceleración y el frenado.

Cuarto, un sistema de comunicación inalámbrico robusto que sea inmune al ruido y a la interferencia. Además, es necesario que el sistema posea un gran alcance para que el vehículo pueda recorrer largas distancias y que sea rápido para tener respuestas en tiempo real de las imágenes y del control.

Quinto, un controlador esclavo que esté conectado a los dispositivos de actuación electromecánicos. Como no necesita un gran procesamiento de información, un microcontrolador es ideal para este propósito, ya que posee suficientes entradas/salidas para la conexión de los dispositivos y tiene una velocidad de procesamiento adecuado a estos casos.

Finalmente, es necesario tener un conjunto de videocámaras que estén posicionadas en el carro, de manera tal que el operador pueda tener un campo visual amplio de su entorno. Principalmente, es necesaria una cámara frontal que muestre una vista panorámica de la pista y una trasera que ayude a tomar decisiones en el viraje.

Selección de tecnologías

La selección de cada una de las plataformas tecnológicas está basada en las necesidades del sistema anteriormente mencionadas. No se consideran, como prioridad, factores económicos sino más bien la base técnica para poder elegir la mejor opción que se ajusta a los requerimientos establecidos.

Para el control maestro se seleccionó un computador portátil, ya que puede usarse en cualquier lugar como estación de trabajo, posee los requerimientos técnicos para la instalación de la interfaz gráfica y las salidas adecuadas para la conexión de los dispositivos periféricos. Es suficiente emplear un *notebook* con más de un puerto USB y con una tarjeta de red para conexión Ethernet.

En cuanto al *software* necesario para la creación de la interfaz, se han comparado tres alternativas: Java, MatLab y Visual Basic. Éstas poseen una gran variedad de características que resultan ventajosas para el proyecto. Se han ocupado una serie de criterios con el fin de elegir el *software* más adecuado.

Es necesario que el programa sea independiente, es decir, que no requiera de éste para ser ejecutado. Sólo dos de los tres *softwares* revisados poseen ésta característica, Java y Visual Basic.

El programa debe ser portable. La aplicación creada puede ser ejecutada en cualquier plataforma, tales como Windows, Mac y Linux. Sólo Java cumple con esta exigencia.

El *software* debe ser accesible, es decir, debe ser fácil de utilizar o cuyo uso sea intuitivo. Todos éstos utilizan un entorno de desarrollo muy similar pero el lenguaje de programación varía en los tres. Visual Basic necesita una base de conocimientos menor que los otros *softwares*.

El *software* debe poseer un gran motor de procesamiento, para ejecutar algoritmos complejos. El único que cumple con los requerimientos es MatLab, pero el uso constante de la consola hace más lenta la aplicación.

Los programas deben ejecutarse en tiempo real. Las aplicaciones creadas en Visual Basic son relativamente rápidas en comparación a las de Java, que necesitan la ejecución de su máquina virtual.



Finalmente, se ha elegido Visual Basic por su facilidad de uso y porque facilita el acceso a las API's de Windows [2], que permiten la manipulación de dispositivos periféricos de manera directa. Además, no requiere de mayores conocimientos para la utilización del mismo. Por otro lado, como no es necesaria la utilización de esta interface en múltiples plataformas, más que en Windows, es la mejor opción en este caso.

Para el tercer tema, se necesita un manipulador que sea compatible con el sistema operativo del computador y que pueda conectarse al mismo por medio de uno de sus puertos USB. Debe poseer características similares al volante del vehículo, como también de los otros elementos de actuación. Por lo tanto, se necesita un *joystick* con forma de volante que pueda girar al menos 2 vueltas y media (900°), que posea pedales para aceleración y frenado, y algunos botones para el control de luces, bocina y marcha.

Para el sistema de comunicación se han contrastado tres alternativas en la tabla 1. WiMax posee un mayor alcance y velocidad de comunicación [3], pero debido a que su costo es mayor, se ha considerado WiFi como la opción más adecuada. WiFi es mucho más asequible, en cuanto a costo y a la oferta de dispositivos de comunicación. Sin embargo, la posibilidad de lograr sólo un centenar de metros en su alcance limita el radio de acción del sistema en general [4]. Una ventaja de WiFi es que los sistemas, en general, están basados en Ethernet, que es muy familiar por su masivo uso en la creación de redes de área local e internet y, por lo tanto, en el mercado comercial hay una gran variedad de dispositivos compatibles que emplean estas tecnologías.

Tabla 1: Comparación de tecnologías inalámbricas.

Tecnología / Características	Wi-Fi	WiMax	ZigBee
Alcance	10 – 100 m	50 Km	13 – 154 m
Velocidad	11 – 54 Mbit/s	124 Mbit/s	28 – 200 Kbit/s
Frecuencia	2,4 – 5 GHz	2,4 – 3,5 – 5,8 GHz	0,86 – 0,91 – 2,4 GHz
Ancho de Banda	1,2 Mbit/s	0,64 - 1,024 Mbit/s	0,250 Mbit/s
Potencia	100 mW	100 mW	1 mW
Aplicaciones	Internet (Computadores portátiles, PDA)	Servicio de banda ancha de largo alcance	Domótica, automatización industrial, reconocimiento remoto
Ventajas	Velocidad	Velocidad y alcance	Bajo consumo de energía
Desventajas	Bajo alcance	Alto Precio	Bajo alcance y velocidad

Para la elección del controlador esclavo se estudiaron tres posibles microcontroladores que serían de utilidad en el proyecto: AVR, PIC y MSP430. De acuerdo a las características de cada uno de ellos, se ha establecido una serie de criterios para su selección.

Según la velocidad de procesamiento, los microcontroladores AVR y PIC alcanzan grandes valores según sea el modelo. Gracias a que éstos utilizan la arquitectura Harvard, su procesamiento es más eficiente. En cambio, los MSP430 son muy limitados en características y su arquitectura Von Neumann los hace más lentos, respecto a sus competidores [5].

La cantidad de Puertos E/S de AVR y PIC es muy variada, dependiendo del modelo del *chip*, ya que hay de diferentes gamas y cada una de ellas ha sido diseñada para diversas aplicaciones. Aunque posee una gran cantidad de puertos, el MSP430 es utilizado para aplicaciones más reducidas.

De acuerdo a la variedad de puertos, AVR, PIC y MSP430 poseen conversores análogos/digitales, comparadores, salidas PWM, E/S análogas y digitales, puertos seriales y temporizadores. Sin embargo,



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



PIC no tiene sus puertos diferenciados como AVR, y es posible ocupar la mayoría de sus puertos tanto como E/S digitales o como E/S análogas.

La mayoría de las plataformas de programación de los microcontroladores PIC, requieren de un programador externo, lo que hace más engorrosa la grabación de los programas. En cambio, las plataformas desarrolladas para AVR y MSP430, sólo requieren de un *software* programador y de la placa que ya contiene dicho programador integrado.

Las tres plataformas poseen un sinnúmero de información disponible, en las propias páginas del fabricante o en foros externos, como también tutoriales y guías simples para su utilización. Pero es en esta área donde AVR, con su plataforma Arduino, ha ganado poco a poco una gran cantidad de adeptos y ha creado una gran red de información, que está fácilmente disponible tanto en inglés como en español.

Finalmente, y de acuerdo al último punto, se ha elegido el microcontrolador AVR, ya que la plataforma Arduino es muy fácil de utilizar y está presente en una gran cantidad de proyectos ya desarrollados. Además, respecto a las cámaras necesarias para la realimentación visual del sistema, es necesario utilizar un par de cámaras que sean fácilmente conectables a la red local del sistema. Como se ha seleccionado Wi-Fi para la comunicación de datos, se usarán enrutadores para la conexión. Por esta razón, es más adecuada la utilización de cámaras IP, ya que poseen su propio servidor y no necesitan de ningún otro elemento adicional, más que conectarse a los puertos Ethernet y ejecutar su aplicación desde un navegador; en contraste con las cámaras Web, que requieren de un computador en el carro y las cámaras análogas que necesitan de adaptadores para conectarse al enrutador [6].

4. EVALUACIÓN ECONÓMICA

Según la evaluación técnica y el modelo formulado inicialmente, es posible crear un esquema más detallado del sistema de teleoperación, tomando en cuenta cada elemento asociado. Consecuentemente, es posible realizar una lista de dispositivos y materiales adecuados para la implementación del mismo. Los elementos necesarios para implementar el proyecto son: un computador portátil que tendrá el *software* de teleoperación y donde se conectará el manipulador, un *joystick* tipo volante y una pedalera que servirán de manipuladores, un enrutador inalámbrico que repetirá la señal del servidor, un enrutador inalámbrico que actuará como servidor del sistema, un par de cámaras IP que serán conectadas directamente al enrutador, una placa de desarrollo que servirá como controlador esclavo, una interfaz de comunicación entre el enrutador y la placa de desarrollo, cables para la conexión hacia los enrutadores y para la conexión entre los dispositivos electrónicos, y baterías para alimentar a cada dispositivo a bordo del carro.

Como se ha mencionado anteriormente, es necesario utilizar dos enrutadores inalámbricos. Uno de ellos se usará como servidor y el otro como repetidor. Para el primero no hay problemas, ya que todos los enrutadores funcionan como servidor, pero el segundo debe poseer las características de un repetidor de señal. En el mercado existen los denominados Access Point, útiles para acceder a una red inalámbrica, pero con la desventaja que poseen sólo un puerto Ethernet y se necesitan conectar al menos: dos cámaras IP y el microcontrolador con interfaz Ethernet. Por otra parte, la potencia de transmisión y la ganancia de recepción son parámetros útiles para la selección del equipo.

Para el enrutador servidor se ha elegido el TL-WR1043ND que posee 3 antenas y la mejor cobertura de los equipos revisados. Para el segundo caso, el dispositivo más adecuado es el enrutador WRT54GL, ya que posee la mejor recepción/transmisión y además es programable, lo que permite transformarlo en un repetidor al instalar una modificación del *software* en su memoria. El precio de ambos está comprendido entre los USD\$ 80 y USD\$ 100, aproximadamente.

En relación a los *joysticks*, el criterio que impera en este sentido es el económico. Los mejores poseen un amplio rango de giro (900°), como el Driving Force GT, que es lo más cercano al manipulador real: dos pedales como mínimo y realimentación de fuerza. Estos dispositivos bordean los USD\$ 200, aproximadamente. Dado que este proyecto es sólo la base de la investigación total, se ha elegido el Twin Wheel F1, equivale a un quinto del precio anterior y a un quinto de su rango de giro (180°).



En cuanto a los microcontroladores, en el mercado existe una gran diversidad de *chips*, un gran número de entradas y salidas, y una gran variedad de puertos e interfaces de programación. Si se trata de elegir componentes según sus características técnicas, la elección es un poco más compleja de realizar, ya que todas las opciones del mercado ofrecen sus propias ventajas, aún cuando muchas de ellas comparten la misma arquitectura física con el Arduino. El criterio para la selección de la placa de desarrollo, en este caso, tiene más relación con la cantidad de información disponible sobre la placa misma, el soporte técnico y la cantidad de proyectos creados para éste, disponibles en la web. Por lo tanto, se ha elegido la placa Arduino Mega 2560 que posee la mayor cantidad de información relacionada con proyectos desarrollados y por su sencilla programación. Conjuntamente, se ha elegido la Shield Ethernet compatible con esta placa, para la interconexión con el enrutador.

Para las cámaras de vigilancia, el criterio utilizado fue meramente económico, ya que la cámara con mejores características es la más cara. Esta cámara posee un mejor lente, resolución, ángulo de visión, características adicionales y, por sobre todo, sólo requiere alimentación desde los puertos Ethernet. Pero el precio de una de esas cámaras equivale, prácticamente, a cuatro veces la más barata y se necesitan por lo menos dos de ellas. Por lo tanto, las cámaras elegidas para este caso son las TV-IP501P de TRENDNET con un valor de aproximadamente USD\$ 120.

El problema de la alimentación

El problema que se presenta, después de la elección de los dispositivos principales, es la forma de cómo éstos se alimentan. Por ende, es necesario realizar una aproximación de los consumos de cada elemento para obtener un consumo total que permita elegir una batería adecuada para este caso. Para ello, se ha calculado un consumo promedio, según los datos que ofrece cada proveedor. En la Tabla 2 se muestran dichos valores que son utilizados como referencia para la compra de la batería.

Tabla 2: Consumo del sistema abordado del carro.

Dispositivo	Voltaje (V)	Corriente (mA)	Potencia (W)
Cámara IP 1	5	300	1,50
Cámara IP 2	5	300	1,50
Enrutador TP-Link	12	1.500	18,00
Placa Arduino	9	240	2,16
Total Aproximado		2.340	23,16

Se calcula aproximadamente un consumo de 2,3 A y 23 W. Esta elección se realiza buscando las denominadas baterías de ciclo profundo, llamadas así porque están diseñadas para ser descargadas en un largo período de tiempo. Son baterías de plomo y ácido, tal como las baterías de arranque usadas en los automóviles, pero con la diferencia de que éstas contienen gel y no líquido. Esta condición permite que las baterías estén libres de mantención, ya que no requieren agua destilada para compensar el vapor disipado por calentamiento. Según lo calculado, se ha seleccionado una batería de 12 V - 12 AH, por dos razones; la primera tiene que ver con el volumen: mientras más grande la batería, más pesada es, ya que el plomo posee una alta densidad; la segunda tiene relación con la autonomía de funcionamiento que ofrece, según la cantidad de corriente que consume el sistema. Si todos los elementos que se conectan a la batería consumen un máximo de 3 A, entonces la batería entregará energía durante 4 horas. Esta batería fue conseguida por USD\$ 40 en una tienda de proyectos de energía solar y requiere de un regulador de voltajes para distribuir la energía a los diferentes dispositivos del sistema.

5. IMPLEMENTACIÓN

La interfaz de operación, sin duda, se lleva el mayor peso del proyecto. Consiste, en términos generales, en un *software* que gestiona tres tipos de periféricos: la tarjeta de red del computador, mediante el uso de



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



sockets; el *joystick* tipo volante, que se conecta al puerto USB; y un par de cámaras IP, que se manejan igualmente mediante conexiones de red.

Para el desarrollo de ésta interfaz se ha utilizado Visual Basic .NET en su versión 2005. La elección de este *software* se hizo considerando principalmente la capacidad de manejo de toda clase de periféricos a través de DirectX 9 y la versatilidad en el control de elementos de red.

Diseño gráfico de la interfaz

Para diseñar una interfaz es importante tener en cuenta tres características fundamentales: que sea visualmente agradable a la vista, que su uso sea intuitivo y que sea robusta en su operación. En base a estos tres puntos, la interfaz de operación se ha diseñado para que el operador de la estación comprenda rápidamente cómo se usa. La interfaz desarrollada posee una ventana principal, donde se alojan todas las funciones del sistema, que consiste en un espacio donde se encuentran dos paneles de control y un menú que permite la interacción con los demás elementos. En la barra de menú existen cuatro opciones que permiten realizar diferentes operaciones: El submenú “Principal”, que permite cerrar la aplicación; el submenú “Ventana”, que permite acceder a las vistas de las cámaras IP; el submenú “Configuración”, que permite acceder a las ventanas de configuración de la red, el *joystick* y las cámaras, y el submenú “?” que entrega la información de ayuda para el usuario.

El panel de control del *joystick* permite visualizar las características del dispositivo conectado, tales como el nombre del controlador y sus capacidades, los estados de los botones y de los ejes X e Y. Por otro lado, el panel de estados está encargado de la información de red. Indica si un cliente se ha conectado o no al sistema. En esta primera etapa sólo se muestra el estado de la conexión, pero la función principal de este panel es mostrar la situación de los actuadores montados en el carro.

En la ventana de configuración de red deben ingresarse los datos del puerto y la dirección IP del servidor (el computador) para comenzar la escucha de la red. La configuración del *joystick* está diseñada para que el operador pueda asignar las acciones del vehículo a uno de los botones o ejes del volante. La configuración de las cámaras sólo necesita el ingreso de las direcciones IP de las cámaras asignadas por el enrutador.

Programación de la Interfaz de Usuario

En esta aplicación se han creado dos nuevas “clases” que permiten manipular los *sockets* de conexión y el controlador del *joystick*. Además, se ha agregado un módulo, con el fin de definir variables globales que sean modificables desde cualquiera de los formularios del proyecto.

La clase DIJoystick se ha creado para interactuar con la Interfaz de Programación de Aplicaciones (API, *Application Programming Interface*), que es un conjunto de funciones y procedimientos ofrecidos por Windows para ser utilizado por otro *software* como una capa de abstracción. DirectX es una colección de APIs desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente en la programación de videojuegos. Esta colección consta de varias soluciones para el manejo de distintos periféricos. Algunas de ellas son: Direct3D, que se utiliza para procesar gráficos tridimensionales; DirectGraphics, que es usada para dibujar imágenes bidimensionales; DirectInput, que se utiliza para el manejo de periféricos (teclados, *mouse* y *joysticks*); DirectSound, que se ocupa de la reproducción y la grabación de sonidos, entre otros. Esta clase consta de una rutina llamada Activar_Joystick(), que es la encargada de la configuración inicial del *joystick* a través de la verificación de la existencia del dispositivo y de otra rutina llamada Sondear_Joystick(), que lee constantemente el estado del manipulador.

La clase Servidor es la encargada del manejo de *sockets* en Windows. Un *socket* es un concepto abstracto por el cual dos programas situados en computadoras distintas (o en la misma) pueden intercambiar cualquier flujo de datos. Este término es usado para nombrar la API encargada del manejo de los protocolos de Internet TCP/IP. Los *sockets* constituyen el mecanismo de transmisión de paquetes



informáticos provenientes de la tarjeta de red a los procesos o hilos apropiados. Están definidos por un par de direcciones IP (local y remota), un protocolo de transporte y, un par de números de puerto local y remoto. Esta clase contiene tres eventos, que determinan cuándo: un nuevo cliente ingresa a la red, hay datos disponibles en el buffer y el cliente abandona la conexión. También posee dos propiedades, una que captura la dirección IP y otra que captura el puerto del servidor, ingresados por el operador. Finalmente, un grupo de funciones que permiten escuchar las peticiones de conexión de cada cliente, cerrar la conexión, obtener los datos enviados por el cliente y enviar datos al cliente.

El programa principal contiene la lógica de control del sistema e interactúa tanto con la red como con el *joystick*. Al iniciar la interfaz es necesario conectarse a la red antes de realizar cualquier acción, ingresando los datos del servidor (IP y puerto de escucha). Al iniciarse la comunicación es posible acceder al panel de control del *joystick*. Para utilizar el manipulador es necesario activar su funcionamiento mediante un botón ubicado del panel; esto permite realizar un sondeo constante de los estados del *joystick*, gracias a la integración de un objeto temporizador en la aplicación. Por cada pulso del reloj se leen los estados de los botones y de los ejes del dispositivo y, entrega, además, los datos del fabricante y sus características. Una vez activado, es posible acceder a las ventanas donde se muestran las imágenes de las cámaras IP.

La configuración del *joystick* permite asignar, a cada acción relacionada con el control del vehículo, un botón que el operador elija. El programa envía dos tipos de datos: continuo, que corresponde a los valores de los ejes X e Y (en un rango de tres dígitos, entre 000 y 900) y discretos, correspondientes a los estados de los botones (180 para activo y 0 para inactivo). En el primer caso, los datos son codificados para poder distinguir entre los datos del eje X y del eje Y. Por lo tanto, al dato numérico enviado por el *joystick* se le agrega un carácter en el encabezado y uno al final. El primero de éstos representará la letra correspondiente al eje (X o Y) y el segundo es el carácter S, que determina el final del dato. En el segundo caso se envía un carácter cuando es presionado el botón. Éstos pueden ser: “A” para activar la intermitente derecha, “B” para la intermitente izquierda, “C” para las luces frontales, “D” para la bocina, “E” para la marcha directa y “F” para la reversa.

Programación del controlador esclavo

El controlador esclavo es el encargado de manipular los actuadores electromecánicos del vehículo y el programa ha sido diseñado para la plataforma Arduino en lenguaje C++. Esta aplicación está encargada de manejar dos clases de dispositivos periféricos. El primero, corresponde a la comunicación con la Ethernet Shield y el segundo, a los actuadores electromecánicos, tales como relés y servomotores.

La comunicación con la Ethernet Shield se realiza mediante el uso de la librería “Ethernet”. Esta librería permite configurar todas las variables involucradas en la comunicación. En este caso, se debe configurar la dirección IP del equipo servidor, para poder conectarse a él, y la dirección MAC del Ethernet Shield, para que el enrutador asigne una dirección a la placa según ese valor. El programa inicia la conexión con el servidor, según los datos anteriormente mencionados, como un cliente. Al corroborar la conexión, el programa entra en un bucle que permite leer permanentemente los datos enviados por el servidor. La aplicación lee los datos y realiza ocho acciones correspondientes a los actuadores del carro. Los primeros dos casos corresponden a la palanca que activa las intermitentes del carro. Se reciben dos caracteres desde la Ethernet Shield: “A” para activar la luz intermitente derecha y “B” para la izquierda. Aquí, existe una interdependencia entre los dos estados, ya que forman parte de un mismo actuador. Por lo tanto, se han declarado tres casos posibles para cada situación:

En el primer caso, cuando el carácter recibido es “A”, si la intermitente derecha e izquierda están apagadas, entonces la luz intermite derecha debe encenderse. Ahora bien, si la intermitente derecha está encendida y la izquierda está apagada, entonces la intermitente derecha debe apagarse. Finalmente, si la intermitente derecha está apagada y la izquierda está encendida, entonces se debe encender la intermitente derecha y apagar la izquierda. La cuarta combinación lógica no es aplicable, ya que no pueden estar encendidas ambas intermitentes de forma simultánea.



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



Para el segundo caso, cuando el carácter recibido es “B”, si la intermitente izquierda y derecha están apagadas, entonces la luz intermite izquierda debe encenderse. Por otro lado, si la intermitente izquierda está encendida y la derecha está apagada, entonces la intermitente izquierda debe apagarse. Finalmente, si la intermitente izquierda está apagada y la derecha está encendida, entonces se debe encender la intermitente izquierda y apagar la derecha. Con esta lógica es posible asegurar el funcionamiento de una sola intermitente a la vez, tal y como sucede en el actuador original.

El tercer caso corresponde al encendido de las luces frontales y traseras del carro. El actuador es un botón con enclavamiento, es decir, al presionar o al jalarlo mantiene su posición. Por lo tanto, cada vez que se presiona el botón correspondiente, se activa uno de los dos estados alternadamente. Para su actuación debe recibir el carácter “C” y su lógica es bastante sencilla: si al presionar el botón “C” las luces están apagadas, entonces éstas deben encenderse y si al presionar “C” las luces están encendidas, entonces éstas deben apagarse.

El cuarto caso corresponde a la bocina. Si se recibe el carácter “D”, cuando se presiona el botón respectivo, se activa el pin conectado al relé de la bocina y al soltarlo éste se desactiva.

Para el quinto y sexto caso, se aplica la misma lógica de las intermitencias, ya que el *switch* de marcha posee tres estados.

Para el caso de los servomotores la lógica es muy distinta. Debido a que los datos que se envían desde la estación de operación son sucesivos, fue necesario implementar un decodificador que permite diferenciar los datos del eje X e Y. Como ya se ha mencionado, los datos se han codificado de la siguiente manera: encabezado (X ó Y) + valor (000-180) + final de código (S).

La lógica del decodificador es bien sencilla. Primero, lee el identificador para ver si corresponde al eje X o al Y. Luego, realiza una nueva lectura y rellena un arreglo de caracteres de tamaño 4 (tres dígitos + carácter nulo) con cada lectura hasta que encuentra el carácter de parada.

Los valores rescatados de la decodificación vuelven a transformarse en un entero, por medio de la función *atoi()*. De esta manera, es posible usarlos para controlar la posición del servomotor.

Conexión y alimentación del sistema

El conexionado general del sistema puede ser representando mediante un esquema que muestra cada uno de sus componentes. Para la alimentación se creó una caja de regulación que entrega tres niveles de tensión provenientes de una única batería de 12 V. Con un regulador LM7805CV es posible entregar los 5 V necesarios para alimentar las cámaras IP y los relés. Para alimentar la placa Arduino se utiliza un regulador LM7809CV que entrega los 9 V, necesarios para su buen desempeño. Para alimentar el enrutador se necesitan 12 V, que pueden obtenerse directamente de la batería, pero como medida de resguardo se ha colocado un fusible que protege el dispositivo de cortocircuitos. Todos ellos están soldados en una placa reticulada y son activados por un *switch* de palanca.

En la Fig. 2 se muestra el diagrama de bloques, que ubica cada dispositivo en su posición original. Ésta corresponde a una vista superior del vehículo, es decir, los elementos dispuestos a la izquierda del diagrama están ubicados en la parte frontal del carro y los de la derecha están en la parte trasera. Las líneas de color rojo representan los conductores conectados al borne positivo de cada aparato, tanto de alimentación como de salida de información. Los de color negro representan los conductores conectados al borne negativo, exceptuando a los de mayor grosor, que representan a los cables UTP, que conectan los puertos Ethernet de cada dispositivo. También, es necesaria una interfaz de conexión de relés ya que no es posible activarlos desde las salidas del Arduino. Los dispositivos que usan electrónica TTL manejan corrientes de hasta 40 mA y, por lo tanto, no es posible activar los relés, ya que éstos necesitan 60 mA, aproximadamente. Es por esta razón que se diseñó un circuito bastante sencillo que permite activar cada relé y proteger las entradas del Arduino de las corrientes inducidas por las bobinas. Este circuito consta principalmente de un transistor NPN, que tiene conectado una resistencia entre la salida del Arduino y la



base del mismo. Su colector está conectado a los 5 V de la caja reguladora y está dispuesto en paralelo con el relé y en antiparalelo con un diodo. Por otro lado, el emisor está conectado a tierra para permitir el flujo de corriente que activa el relé. Esto sólo funciona cuando el transistor es usado en corte y saturación, ya que trabaja como un *switch* controlado por corriente.

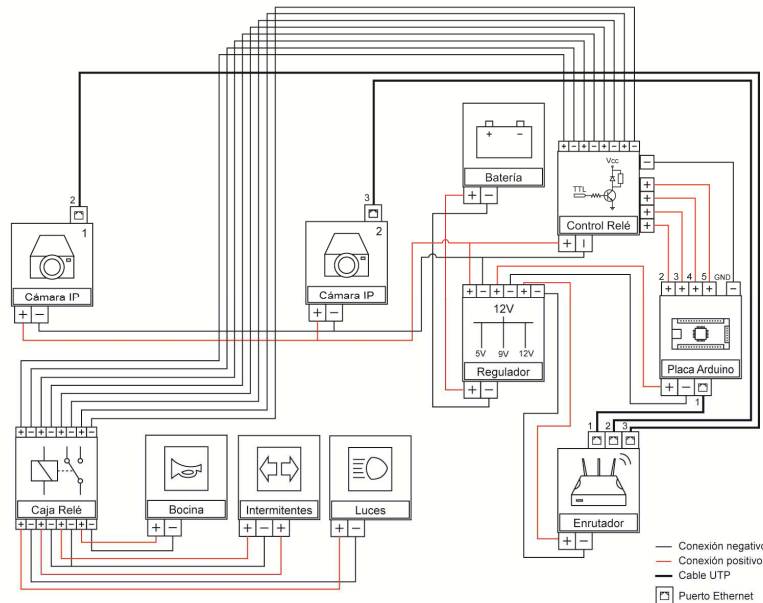


Figura 2: Conexionado del sistema.

CONCLUSIONES

El estudio realizado ha recopilado suficientes datos que permiten determinar qué tipos de plataformas son las más adecuadas en la creación de un sistema de teleoperación y el nivel de integración que representan. De esta manera, es posible concluir la primera de las aseveraciones, *i.e.* un aparato puede tener características técnicas muy atractivas, pero si el precio, la asequibilidad y la facilidad de manejar son un problema, no importa cuántas ventajas comparativas posea. Por otro lado, al momento de crear el presupuesto se deben tener en cuenta los problemas futuros. Teóricamente, los dispositivos adquiridos en las tiendas están probados y no presentan ninguna falla, sin embargo, es muy probable que alguno de ellos posea un desperfecto de fábrica, lo que agrega retrasos en la implementación ya que deben ser reparados o cambiados. Por esta razón, siempre es necesario tener los documentos que acrediten la compra de cada uno de ellos, para evitar problemas innecesarios. Una mala gestión de los recursos aumenta los tiempos de trabajo y puede significar un aumento adicional en los costos del proyecto.

El desarrollo del programa implica un gran trabajo, ya que es necesario conocer muy bien el funcionamiento y el lenguaje de programación que se utiliza. Es importante el orden de cómo se escribe el código. No da lo mismo llamar una variable en el principio del formulario que al final, ya que ésta puede definir qué elementos se activan primero y podría ser determinante a la hora de ejecutar el código. Uno de los principales problemas que tiene el código generado en la placa Arduino es el orden en que se declaran las variables y cómo se ejecuta el programa. Las acciones que necesitan ser ejecutadas constantemente deben ser escritas dentro de un bucle infinito. El problema radica en que la conexión Ethernet debe realizarse fuera de esta función, de lo contrario estaría conectándose constantemente e impediría el normal funcionamiento de la aplicación. Al definirse fuera del bucle sólo es posible realizar esa conexión una sola vez y si por algún motivo el *software* de operación es cerrado, el controlador debe ser reiniciado para poder conectarse nuevamente con el sistema. Por otro lado, en la aplicación creada en Visual Basic siempre es importante definir correctamente cada uno de los elementos. Para asignar una acción a uno de estos objetos (por ejemplo, un botón), es necesario agregarla dentro del código generado



del 26 al 30 de noviembre de 2012
**16 CONVENCIÓN CIENTÍFICA
DE INGENIERÍA Y ARQUITECTURA**
PALACIO DE CONVENCIONES DE LA HABANA



por el programa al seleccionar el elemento, ya que si se escribe directamente en el formulario, es posible que no sea reconocido por el compilador y ni siquiera lo advierta como un error.

A pesar de todos los inconvenientes en la programación de la interfaz, ésta funciona de manera estable y es suficientemente amigable para que cualquier persona pueda utilizarla, sin necesidad de tener conocimientos previos. El programa está preparado para manejar los actuadores del volante, los pedales, las luces y la bocina. Las cámaras de red se pueden ver perfectamente y el retardo de la conexión inalámbrica no va más allá de un segundo. La instalación de los dispositivos, la creación de placas electrónicas y la fuente que alimenta al sistema, fueron soluciones aplicadas cuando el sistema básico estaba armado. Muchas de las fallas del sistema, en cuanto a *hardware* y *software* quedaron de manifiesto a la hora de implementar el sistema en su totalidad y fueron resueltas con soluciones simples, dando un valor agregado al proyecto. Por lo tanto, es posible decir que factible es realizar un sistema de operación remota con las tecnologías disponibles en el mercado, con pocos recursos y sin la necesidad de diseñarlo desde cero.

RECONOCIMIENTOS

Los autores de este artículo agradecen el apoyo brindado por:

- El Departamento de Investigaciones Científicas y Tecnológicas de la Universidad de Santiago de Chile, por medio del Proyecto DICYT 060713UO.
- El Grupo de Automática, con su constante inspiración e impulso a la investigación científica.
<http://www.die.usach.cl/>

REFERENCIAS

1. NUÑO, E. y BASAÑEZ, L. *Teleoperación: técnicas, aplicaciones, entorno sensorial y teleoperación inteligente*. [en línea]. [ref. de 4 de Abril 2011]. Disponible en Web: <<http://upcommons.upc.edu/e-prints/bitstream/2117/570/1/IOC-DT-P-2004-05.pdf>>.
2. Varios. *Visual Basic*. [en línea]. Wikipedia. Disponible en Web: <http://es.wikipedia.org/wiki/Visual_Basic>.
3. GONZALEZ, A. *Redes: protocolos y estándares*. [En línea]. MailxMail. [ref. de 13 de Enero 2009]. Disponible en Web: <<http://www.mailxmail.com/curso-redes-estandares-3/caracteristicas-wimax>>.
4. JEFF. *Introducción a Wi-Fi*. [En línea]. Kioskea. [ref. de 16 de Octubre 2008]. Disponible en Web: <<http://es.kioskea.net/contents/wifi/wifiintro.php3>>.
5. RUGGER. *Comparativa de microcontroladores actuales*. [En línea]. Departamento de Ingeniería en electricidad, UPV. Disponible en Web: <<http://server-die.alc.upv.es/asignaturas/LSED/2002-03/Micros/downloads/trabajo.pdf>>.
6. Equipo Domodesk SL. *A fondo: Cámaras IP*. [En línea]. Domodesk. Disponible en Web: <<http://www.domodesk.com/content.aspx?co=82&t=146&c=43>>.

SOBRE LOS AUTORES

Fabían Arancibia Santti, Licenciado en Ciencias de la Ingeniería. Estudiante de Ingeniería Civil en Electricidad, Universidad de Santiago de Chile (USACH).

Claudio Urrea Oñate, Ph.D. en Ciencias de la Ingeniería, mención: Automática y Producción, INPG, Francia. Profesor/Investigador Jornada Completa del Departamento de Ingeniería Eléctrica de la USACH. Director del Programa de Modelación Industrial y Simulación Computacional de la Facultad de Ingeniería, USACH.