

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



**Manipulación de un carro de juguete vía Internet
Basado en el Paradigma de Computación Ubicua**

Reporte Técnico de Investigación presentado por:

Raúl García Martínez 71804

Miguel Carmona Rosales 75795

Requisito para la obtención del título de

INGENIERO EN SISTEMAS COMPUTACIONALES

Profesor Responsable: *M.C. Saúl González Campos*

Noviembre de 2009

Manipulación de un carro de juguete vía internet

Basado en el paradigma de Computación Ubicua

Los abajo firmantes, miembros del comité evaluador autorizamos la impresión del proyecto de titulación de:

Raúl García Martínez 71804

Miguel Carmona Rosales 75795

M.C. Saúl González Campos

Asesor

Dr. Carlos Alberto Ochoa Zezzatti

Profesor de la materia

DECLARACION DE ORIGINALIDAD

Nosotros Raúl García Martínez y Miguel Carmona Rosales declaramos que el material contenido en esta publicación fue generado con la revisión de los documentos que se mencionan en la sección de referencias, y que el programa de computo (Software) desarrollado es original y no ha sido copiado de ninguna otra fuente, ni ha sido usado para obtener otro título o reconocimiento en otra Institución de Educación Superior.

Raúl García Martínez

71804

Miguel Carmona Rosales

75795

AGRADECIMIENTOS

Doy gracias a Dios por dejarme llegar hasta este punto tan importante en mi vida, por darme ánimos día a día y por darme el apoyo de todas las personas que de alguna forma intervinieron en toda mi carrera para que yo pudiera terminar.

A mi padre, gracias por todo lo que me dejaste, por la mejor herencia que me pudiste regalar, la escuela. Mil gracias por tus consejos, por dejarme todo lo necesario para llegar hasta aquí, donde sea que te encuentres se que me estas apoyando, este trabajo es para ti y para mi madre.

A ti madre, que cada día me desear lo mejor, no tengo palabras para decirte cuanto te agradezco tu apoyo, en las buenas y en las malas siempre has estado conmigo, sin ti no hubiera llegado a ningún lado. Por todos los días de mi carrera que te levantabas a hacerme de comer para llevar a la escuela, por todos los detalles que tienes conmigo mil gracias.

A todos y cada uno de mis hermanos(as), por todo el apoyo que me brindaron cada día, por cada peso que se sacaron de su bolsa para que yo pudiera ir a la escuela, por animarme a seguir estudiando, por aconsejarme igual que nuestro padre, por sentirse orgullosos de mi por estar en la universidad. También sin ustedes no hubiera llegado lejos, les agradezco cada detalle que tuvieron conmigo.

A mi compañero y amigo Miguel Carmona Rosales, gracias por realizar este trabajo en conjunto conmigo, por tu cooperación y tu tiempo para poder terminar nuestro proyecto. De mi parte haber trabajado en esto, creo que nos unió mas como amigos. Suerte y Felicidades Ingeniero.

A mi asesor M.C Saúl González Campos, por su apoyo, sus consejos, su conocimiento y sobre todo el tiempo que nos brindo, por confiar en nosotros y aceptar ser nuestro asesor, fue parte importante en esto, por todo muchas gracias.

Raúl García Martínez

AGRADECIMIENTOS

Gracias a Dios, por permitirme llegar hasta este momento y cumplir una más de mis metas, ayudándome durante el camino de la universidad.

A mi familia, gracias a cada uno de ellos que me han visto crecer día con día. A mis padres, por su gran apoyo que me brindan y confiar en mí en cada una de las metas que he podido lograr. A mis hermanos que me brindan ese cariño y me hacen reír cada momento, los quiero mucho.

Maestra Elizabeth, gracias por brindarnos amablemente su conocimiento, apoyándonos desde el inicio y por el entusiasmo que nos mostró siempre.

A mis amigos, por esas palabras de aliento y estar en los momentos más difíciles, gracias a todos y cada uno de ellos que ocupan una parte importante en mi vida.

Al Raúl, por ser uno de mis grandes amigos y por supuesto gran compañero, juntos siendo un gran equipo dentro y fuera en la realización de este proyecto.

Maestro Saúl González, un agradecimiento especial por brindarnos su apoyo y confianza desde el inicio, por su ayuda y paciencia guiándonos en el transcurso del proyecto, pero sobre todo ser una excelente maestro y persona.

A todas y cada una de las personas que de alguna manera influyeron en mí y en la realización de esta meta. Gracias.

Miguel Carmona Rosales

LISTA DE FIGURAS E ILUSTRACIONES

ILUSTRACIÓN 1. COMPONENTES DE <i>ARDUINO</i> ...	14
FIGURA 1. ENTORNO DE SOFTWARE DE <i>ARDUINO</i>	15
FIGURA 2. COMPONENTES DE <i>LANTRONIX</i>	17
FIGURA 3. DIAGRAMA DE SECUENCIA	18
ILUSTRACIÓN 2. MOTOR ELÉCTRICO TOMADO DEL CJ	20
FIGURA 4. PUENTE H	21
FIGURA 5. FLUJO DE CORRIENTE EN SENTIDO ANTI HORARIO.....	22
FIGURA 6. FLUJO DE CORRIENTE EN SENTIDO HORARIO.	23
FIGURA 7. CIRCUITO ELÉCTRICO PARA EL CONTROL DE UN MOTOR.	25
FIGURA 8. PINADO DEL L298N.....	25
FIGURA 9. DIRECCIONAR LA RUTA DE <i>CLASSPATH</i>	29
FIGURA 10. COMPILACIÓN DE ARCHIVOS CON EXTENSIÓN JAVA.....	30
FIGURA 11. PASOS PARA LA CONFIGURACIÓN DE LANTRONIX.....	31
FIGURA 12. CONVERSIÓN A ARCHIVO “.COB”	32
FIGURA 13. TRANSFERENCIA DE PROYECTO.COB A LANTRONIX.....	32
FIGURA 14. CIRCUITO FINAL PARA LA MANIPULACIÓN DEL CJ.....	33
FIGURA 15. DIAGRAMA DEL FUNCIONAMIENTO DEL SISTEMA	34
FIGURA 16. INTERFAZ	35

TABLAS

TABLA 1. REQUISITOS CLIENTE Y SERVIDOR	19
TABLA 2. TABLA COMPARATIVA ENTRE L293 Y L298.....	24
TABLA 3. PUERTOS DE ARDUINO UTILIZADOS.....	26
TABLA 4. ESPECIFICACIONES FÍSICAS DEL CJ.....	40

ÍNDICE

DECLARACIÓN DE ORIGINALIDAD.....	IV
AGRADECIMIENTOS.....	V
LISTA DE FIGURAS.....	VII
TABLAS.....	VII

CAPITULO 1

1.1 Introducción.....	1
1.2 Planteamiento del problema.....	1
1.3 Antecedentes.....	2
1.4 Objetivo.....	2
1.4.1 Objetivos específicos.....	2
1.5 Justificación.....	3
1.6 Marco metodológico.....	3

CAPITULO 2

2.1 Mando a distancia o control remoto.....	6
2.2 Señales Analógicas y Digitales.....	6
2.3 Tipos de control remoto.....	8
2.4 El protocolo de comunicación.....	10
2.5 Computación Ubicua.....	11
2.6 Arduino.....	12
2.6.1 Componentes de Arduino.....	13
2.6.2 Versiones de Arduino.....	15
2.6.3 Software Arduino.....	15
2.7 Lantronix WiMicro (Serial to Wifi).....	16
2.7.1 Especificaciones.....	17

CAPITULO 3

3.1 Requisitos para el desarrollo.....	19
3.2 Desarrollo.....	19
3.2.1 Puente H.....	21
3.2.2 Integrado L298.....	23
3.2.3 Placa USB Duemilanove.....	26
3.2.4 Programa para controlar el L298N.....	27
3.2.5 Estructura Interna.....	27
3.2.5.1 Servidor Web.....	27
3.2.6 Circuito final.....	33
3.2.6.1 Alimentación del Circuito final.....	33
3.2.7 Interfaz.....	35
3.3 Experimentos.....	36
3.3.1 Circuito controlador de los motores.....	36
3.3.2 Servidor Web.....	36
3.3.3 Código.....	37
3.4 Guía de uso.....	38

CAPITULO 4

4.1 Resultados y conclusiones.....	38
4.2 Trabajo relacionado.....	39
4.3 Trabajo a futuro.....	40
 Anexo.....	 42
Apéndice A.....	47
Bibliografía	56

CAPITULO 1

1.1 Introducción

Actualmente se observan muchos avances tecnológicos en los cuales el medio que ha dado grandes cambios es el *Internet*, el cual ha causado un gran impacto en el trabajo, el ocio y el conocimiento a nivel mundial.

También se conoce la tecnología de radiocontrol usada en la actualidad, se sabe que tiene alcance limitado, de aquí surgió este proyecto que trata de usar otro medio diferente al radiocontrol para controlar un vehículo desde cualquier lugar con *Internet*, al realizar este proyecto se pretende modificar un carro de juguete de este tipo aplicando el paradigma de *computación ubicua*.

El usuario podrá manipular el carro desde cualquier lugar, siempre y cuando el carro y el usuario estén conectados a *Internet*. No se requiere que exista una computadora en el lugar donde se localiza el carro, solamente se requiere una conexión inalámbrica a *Internet*. El carro recibirá las instrucciones a través de una aplicación *Web* desarrollada en *PHP* (*PHP Hypertext Pre-processor*) la cual se comunicará por internet inalámbrica a un microcontrolador ATmega168 programado en lenguaje “C”, acoplado al carro de juguete.

Este proyecto es la base para ser complementado posteriormente y utilizarlo para diferentes tipos de aplicaciones para controlar diversidad de objetos.

1.2 Planteamiento del problema.

La comunicación de radio control no permite controlar a grandes distancias algún tipo de objeto remotamente. La idea de manipular el carro a través de Internet surge al observar la distancia limitada que proporciona la comunicación de radiocontrol, y se decide mejorarla por medio de una conexión inalámbrica a Internet utilizando artículos convencionales tal como es un carro de juguete.

1.3 Antecedentes

Como ya es conocido, existen dispositivos que se puede controlar remotamente gracias a las tecnologías como son: *Wi-Fi*, *bluetooth*, *infrarrojo*, *radiofrecuencias*, etc.

Durante la investigación se encontraron algunos dispositivos con los que contamos en la actualidad, los cuales se mencionan a continuación:

- *Wi-Fi*: Al ser redes inalámbricas, la comodidad que ofrecen es muy superior a las redes cableadas porque cualquiera que tenga acceso a la red puede conectarse desde distintos puntos dentro de un rango suficientemente amplio de espacio.
- *Bluetooth*: Se utiliza principalmente en un gran número de productos tales como teléfonos, impresoras, módems y auriculares. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda. Su uso más común está integrado en teléfonos y PDA's.
- *Radiofrecuencia*: La radiofrecuencia son radiaciones electromagnéticas que oscilan simultáneamente en el campo eléctrico y magnético, podemos encontrarlos en controles de televisión o autos de control remoto.
- *Infrarrojo*: Dispositivo electrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión.

1.4 Objetivo

Manipular un carro de juguete en forma remota, utilizando una aplicación *Web* y una conexión inalámbrica a *Internet*.

1.4.1 Objetivos específicos

- Diseñar el circuito que va a utilizar el carro de juguete.
- Programar el micro-controlador.

- Desarrollar las aplicaciones Web tanto la del dispositivo marca *Lantronix* como la del *Servidor web*.

1.5 Justificación

Un juguete diseñado para ser controlado a distancias pequeñas (por radiofrecuencia) se convierte en un juguete controlable desde cualquier lugar sin importar la distancia, siempre y cuando exista un acceso inalámbrico a Internet. Lo anterior hace que este proyecto sirva como base para futuros proyectos de investigación, con distinta aplicación pero que utilicen la misma metodología.

1.6 Marco metodológico

La computación ubicua es la integración de la informática en el entorno de la persona, de forma que los ordenadores no se perciban como objetos diferenciados; la tendencia de este paradigma es la fusión del computador con los objetos cotidianos típicos para que se vuelva literalmente invisible [1].

La idea del cómputo ubicuo propone un ambiente físico dotado de dispositivos de diferentes escalas, con capacidades computacionales y de comunicaciones, los cuales se integran de forma natural a nuestras actividades diarias [Weiser, 1991, 1993]. Es decir, los dispositivos computacionales se convierten en una herramienta de trabajo que solo requiere de nuestra atención periférica. La interacción humano-computadora sucede en forma natural e implícita, lo que permite que los usuarios no requieran estar conscientes de la existencia de los dispositivos, lo cual es posible mediante el acceso consciente del contexto a información y servicios disponibles en el ambiente [Abowd y Mynatt, 2000]. Por contexto se entiende que es cualquier información relevante para mejorar la interacción entre un usuario y una aplicación, por ejemplo localización, identidad y actividades de los usuarios, [Dey, 2001]. Así, la tecnología consciente del contexto es considerada el corazón de los sistemas de cómputo ubicuo ya que permite que el sistema reaccione ante los cambios de contexto de los usuarios [2].

La Computación Ubicua representa un gran desafío científico/técnico, un gran nicho de oportunidad y es un área atractiva para el sector empresarial. Su adopción, como una de las líneas estratégicas de desarrollo del país, se reflejará en un impacto social, un impacto científico, un impacto tecnológico, y finalmente en un impacto económico. Las áreas de investigación y desarrollo que se identifican como de alta prioridad dentro del cómputo ubicuo y sin ser exhaustivos, se presentan a continuación [3]:

- *Sensores*
- *Redes de próxima generación*
- *Sistemas Distribuidos*
- *Computación móvil*
- *Servicios de posicionamiento y localización*
- *Desarrollo de sistemas ubicuos (aplicaciones)*

La tecnología *Wireless* (redes sin cables) y los dispositivos móviles constituyen el fundamento tecnológico para construir este tipo de ambientes inteligentes [4].

Un lenguaje de programación diseñado para crear páginas Web dinámicas es *PHP* (*PHP Hypertext Pre-processor*),

- Este nos permite el envío de datos a través de internet.
- El código que debe ejecutarse se ejecuta siempre en el servidor.
- Lenguaje multiplataforma
- Conectividad con bases de datos
- La sintaxis de *PHP* es similar a la del lenguaje “C”, por esto cualquiera con experiencia en lenguajes del estilo “C” podrá entender rápidamente *PHP*. [5]

WampServer (Windows-apache-MySQL-PHP Server).- se encarga de interpretar el código dinámico de cada rutina de *PHP* y realiza las instrucciones que se lleven a cabo en cada aplicación [6].

LANtronix.- Es un dispositivo a nivel de tablilla que integra componentes de hardware que proveen comunicación vía internet, también cuenta con soluciones de software para facilitar la función de conexión [7].

Arduino.- Tablilla electrónica de código abierto, equipada con entradas digitales y análogas, es una plataforma de hardware de fuente abierta basada en una sencilla placa de entradas y salidas simple, las plataformas *Arduino* están basadas en los microcontroladores *Atmega168* o en el *ATmega8*, chips sencillos que permite el desarrollo de múltiples diseños basado en lenguaje “C” [8].

Java.- Es un lenguaje de programación, desarrollado por *Sun Microsystems*, permite crear programas de uso personal, para empresas, aplicaciones distribuidas a través de Internet, aplicaciones de bases de datos y muchos más. Es un lenguaje orientado a objetos fácil de usar, el cual usa sus famosos programas incrustados en las páginas Web llamados *applets*. Para que estos *applets* funcionen solo se necesita que el navegador de internet sea compatible con java en conjunto con la maquina virtual de java [9].

Maquina Virtual Java (JVM).- Es una de las piezas fundamentales de la plataforma java, la gran ventaja de la maquina virtual java es aportar portabilidad al lenguaje de manera que un programa *.class* escrito en *windows* puede ser interpretado en un entorno *linux*. Tan solo es necesario tener dicha maquina virtual para dichos entorno [10].

Java Developet Kit de Sun.- Incluye el API (Application Programming Interface) de Java, el JRE (JVM), compilador de *Java* y otras funcionalidades definidas por *Sun*. Si no está familiarizado programando en *Java*, el API de Java es un conjunto de clases que es utilizado para generar programas básicos en el lenguaje; utilizando una analogía, estas clases tienen la misma funcionalidad que las *funciones clases estándar* utilizadas en otros lenguajes.[9]

CAPITULO 2

2.1 Mando a distancia o control remoto

Un mando a distancia o control remoto es un dispositivo electrónico usado para realizar una operación remota (o telemando) sobre una máquina. Esta máquina responderá arrojando una señal de salida esperada en función de la señal de entrada que nosotros le apliquemos.

Nikola Tesla fue un físico, matemático, ingeniero eléctrico y célebre inventor que revolucionó la teoría eléctrica desarrollando las bases para la generación de corriente alterna. Su patente número 613809, titulada “*Método de un aparato para el mecanismo de control de vehículo o vehículos en movimiento*” resulta ser el primer ejemplo bien fundamentado de control remoto [11].

El mando a distancia nos permite participar en actividades peligrosas a través de máquinas, o en situaciones en que el hombre no tiene la fuerza, la energía o la característica necesaria para realizar alguna acción. También el control remoto puede hacer más precisas las acciones del hombre. Es ahí donde recae el proyecto que aquí se presenta.

2.2 Señales Analógicas y Digitales

La necesidad de contar y medir es innegable. Cuando deseamos medir la temperatura usamos un termómetro, pero en realidad no medimos directamente la temperatura sino el desplazamiento de una sustancia que es proporcional a dicha temperatura. Cuando medimos el tiempo en realidad medimos el movimiento de una aguja en un reloj proporcional a la duración del día. Esto es porque las cantidades analógicas son continuas mientras que las digitales son discretas.

“Un sistema digital es una combinación de dispositivos diseñado para manipular cantidades físicas o información que estén representadas en forma digital; es decir, que solo puedan tomar valores discretos” [11].

Si a la duración del día le damos valores numéricos, 0 horas para iniciar un nuevo día y 24 horas para finalizar entonces podremos medirlo en ese rango. Estos números son cantidades discretas porque no son continuas sino que cambian cada cierto rango.

“Un sistema analógico contiene dispositivos que manipulan cantidades físicas representadas en forma analógica. En un sistema de este tipo, las cantidades varían sobre un intervalo continuo de valores” [11].

Las señales analógicas como la luz, el tiempo, el voltaje, la corriente están presentes en nuestras vidas diarias. Si a éstas las manipulamos de forma discreta, apagado-encendido (0,1), bajo voltaje-alto voltaje ([0,0.8] volts, [2,5] volts) será más fácil manejarlas, entenderlas y manipularlas. Esto es sencillo de imaginar debido a que son números pero físicamente se hace con herramientas que llamamos convertidores de señal.

Un conversor (o convertidor) analógico-digital (CAD), (ADC) es un dispositivo electrónico capaz de convertir un voltaje determinado en un valor binario, en otras palabras, este se encarga de transformar señales análogas a digitales.

Así una señal analógica como es el sonido puede ser medido y digitalizado, y de esta forma podemos agregarle matices, eliminar ruido y amplificarlo. La señal digital resultante la convertimos a una cantidad analógica como puede ser movimiento, luz o calor y entonces obtener control de cualquier aparato.

Un mando a distancia de un sólo canal permite enviar una señal portadora, usada para accionar una determinada función. Para controles remoto multicanales, se necesitan procedimientos más sofisticados; uno de ellos consiste en modular la señal portadora con señales de diferente frecuencia. Después de la demodulación de la señal recibida, se aplican los filtros de frecuencia apropiados para separar las señales respectivas. Hoy en día, se suelen usar métodos digitales.

2.3 Tipos de Control Remoto

Por sistema cableado se entiende todo circuito eléctrico o electrónico que exige el montaje de distintos módulos unidos (cableados) entre sí, para realizar un determinado proceso o secuencia lógica, que por lo general servirá para controlar un sistema. El flujo eléctrico se transporta a través de material conductor rodeado por aislante. Esta señal eléctrica en la actualidad es la más empleada para comunicar información. Sin embargo, al incorporarse el cableado al sistema hace que este ocupe más espacio, aumente su peso, y la comunicación remota estará en función de la longitud de cable que aumentará su costo con cada m².

En los inicios de los años 80 se desarrollaron los semiconductores para emitir y recibir radiación infrarroja. El semiconductor es una sustancia que se comporta como conductor y como aislante y es capaz de emitir una frecuencia luminosa cercana al infrarrojo que aun que no sea visible para el hombre, es suficiente para transmitir datos hacia un aparato. Los enlaces infrarrojos son más prácticos que el cableado ya que no aumentan su peso y son más portables (control inalámbrico) pero se encuentran limitados a por el espacio y los obstáculos y como la longitud de onda de los rayos infrarrojos esta en el rango de 850 nm y 900 nm no pueden propagarse tanto como lo hacen las señales de radio.

La técnica de radiocontrol permite la manipulación de un objeto a distancia y de manera inalámbrica por medio de ondas de radio. Estas son electromagnéticas y de menor frecuencia, por lo tanto son de mayor longitud de onda y menor energía.

Los dispositivos que generan las microondas están basados en los mismos semiconductores. Hasta aquí se ha mejorado el mando a distancia en su practicidad, portabilidad, y longitud de control, pero al igual que las señales infrarrojas no puede haber obstáculos o la transmisión no se llevará a cabo.

Un avance en la tecnología importante para este tipo de transmisión es la tecnología satelital.

Al respecto el Dr. Dionisio M. Tun Molina Director Ejecutivo de Ingeniería y Operación Satelital de Satélites Mexicanos S.A. de C.V. publicó en enero 2002:

“Los sistemas de satélites no dependen de líneas y conexiones físicas montadas a lo largo de la superficie de la Tierra, sino de estaciones terrenas ubicadas en diferentes lugares, cuyo costo para su puesta en operación es mucho más bajo que construir una carretera. Una estación terrena que está bajo la cobertura de un satélite le envía una señal de microondas, denominada enlace ascendente. Cuando la recibe, el transponedor (aparato emisor-receptor) del satélite simplemente la retransmite a una frecuencia más baja para que la capture otra estación, esto es un enlace descendente” [12].

De esta manera se aprovecha la extensa propagación de las ondas de radio que son direccionadas a largas distancias siendo el satélite un intermediario que las re direccionan y así se reduce, a casi nula, la posibilidad de obstáculos alcanzando a comunicar grandes áreas.

La señal eléctrica, la señal infrarroja y las ondas de radio son en realidad tipos de energía. Cada una de ellas tiene propiedades y campos en los que se desenvuelven mejor que las otras, y también dependiendo de los parámetros que se deseen controlar en el dispositivo se selecciona la mejor. Entonces si se pudiera emplear dentro de un proceso el tipo de energía que mejor aprovechara las circunstancias, nos diera la

respuesta más exacta y/o mejor se manipulara, estaríamos optimizando a gran escala las funciones y respuesta del proceso.

Transductor: “Dispositivo que transforma el efecto de una causa física, como la presión, la temperatura, la dilatación, la humedad, etc., en otro tipo de señal, normalmente eléctrica [17].

Un ejemplo de un transductor es el micrófono. Este aparato recibe energía mecánica (ondas sonoras de la voz) y lo transforma (transduce) a energía eléctrica que es fácil de transportar y modificar (ejemplo: amplificar). Esta atraviesa el cableado llega a un amplificador (bocinas) y la señal de salida se vuelve a transformar en energía sonora pero esta vez amplificada.

Entonces el transductor nos permite transformar un tipo de señal en otra, estando esta última siempre en función de la primera. Y con esto un aparato mecánico puede ser accionado con una señal eléctrica, un dispositivo térmico puede generar una señal de respuesta eléctrica y un móvil basado en un circuito electromecánico puede ser controlado por señales de radio transmitidas desde un aparato eléctrico y digital como la computadora.

2.4 El protocolo de comunicación

La dirección IP es una serie finita de números delimitada en un espacio de 32 bits que funciona como un identificador de dispositivos. Este código de 4 cifras de 8 bits cada una separadas por un punto identifica lógica y jerárquicamente la interfaz dentro de una red que emplee el protocolo IP que corresponde al nivel de red del protocolo TCP/IP. Así como el usuario puede hacer uso de los dispositivos de entrada de cualquier computadora como el teclado u otros periféricos para comunicarse con ella, dos programas pueden entenderse entre si gracias a las aplicaciones Cliente-Servidor.

“Cuando arrancó la década del ’90 se estandarizaron los protocolos de conexión TCP/IP y fue entonces que Berners- Lee diseñó el primer sistema de comunicación entre un servidor y el cliente. Aquello fue el inicio de la World Wide Web (www). Poco después, en 1993, se unificó el lenguaje Web gracias al Hypertext Markup Language (HTML), que aún hoy se usa” [13].

Si las aplicaciones de las que estamos hablando permiten comunicar dos programas entre sí, también puede comunicar computadoras. Hay 2 tipos de conexiones para una red, las físicas que permiten a los ordenadores recibir y transmitir señales de forma directa, y las lógicas que intercambian información.

Internet usa el protocolo TCP/IP (Transmission Control Protocol / Internet Protocol), este recibe los paquetes de información y los redirige al usuario que los solicitó. UDP (Protocolo de datagramas de usuario) es el competidor directo en este medio, la diferencia más relevante entre uno y el otro, es que UDP se especializa más en transmisiones rápidas de información aceptando la pérdida de datos, mientras que TCP/IP asegura la transmisión completa empleando más tiempo.

Para el proyecto expuesto en este documento es importante la veracidad de los datos que se enviarán y serán recibidos por el móvil, pues de esto depende la respuesta esperada del sistema. Se selecciona pues el protocolo de red TCP/IP para la comunicación del sistema.

Anteriormente se trataron las distintas modalidades para la transmisión de señales remotas.

Si un programa puede acceder a otra computadora se puede denominar “Herramienta de transmisión remota”. Si esta terminal reacciona en función de una señal de entrada entonces estamos hablando de Mando a Distancia, pero esta vez por vía Internet.

2.5 Computación Ubicua

Internet maneja una comunicación *Wireless* que funciona a base de modulación de ondas electromagnéticas, radiaciones o medios ópticos. Varios clientes

comunicándose con un servidor conforman una red, un ejemplo de esto es el sistema de envío de datos *Wi-Fi* que transmite sus señales a través de ondas de radio.

Conforme avanza la tecnología se ha observado la miniaturización de los componentes hasta compararlos con pequeñas prendas de vestir. Esto deja al descubierto la necesidad de mantener dichos componentes “informados” con el movimiento del usuario y entonces crear un nuevo concepto “La tecnología móvil”.

Las redes móviles sin hilos (cables) utilizan servicios telefónicos y servicios públicos para recibir y transmitir señales. Para ello utilizan:

- Comunicación de paquetes vía radio.
- Redes celulares.
- Estaciones de satélite.

Con la tecnología móvil la convivencia hombre-máquina se hizo más constante hasta hacerse común el disponer siempre de una laptop y de un celular a la mano. Ahora la vida diaria del hombre se ve conjugada por nuevos elementos que figuran como parte de su ambiente.

“El cómputo ubicuo es la integración de las tecnologías de la información en la vida diaria, de tal forma que una computadora se perciba como elemento del ambiente”
[14].

2.6 Arduino

Tablilla electrónica, que cuenta con entradas análogas y digitales, su sencillez y bajo costo hacen de ella un elemento de aprendizaje dentro de la electrónica digital, implementando diversos proyectos.

La tablilla está compuesta por componentes según el uso que se le dé, pero también se puede adquirir sin componentes para darle un toque personal.

Entre los componentes de una tablilla predefinida están: microcontrolador, entradas y salidas, análogas y digitales, botón de *reset*, LEDS. Su fuente de alimentación puede ser de tres formas, si cuenta con USB tipo A y B de aquí puede tomar el voltaje necesario para su funcionamiento (5V), también cuenta con un *jack*, el cual se alimenta con un adaptador de corriente, y por último está compuesta por un puerto de entrada de voltaje.

2.6.1 Componentes de Arduino

En esta sección se describirá físicamente como se compone la tablilla, la cual tiene un tamaño aproximado de 7 cm de largo y 6 cm de ancho.

Se irán describiendo uno por uno sus elementos, los cuales se muestran en la *ilustración 1*.

Cuenta con 6 puertos análogos, 14 puertos de digitales que pueden ser utilizado tanto como de entrada o de salida numerados del 0 al 13, los primeros 2 (0 y 1) son utilizados como receptor (*Rx*) y transmisor (*Tx*) respectivamente, también cuenta con 3 puertos de tierra. Además está equipado con puertos para voltaje, 5v de salida, *Vin* para alimentar todo el circuito, y 3V3 para 3.3v. En versiones anteriores se contaba con un jumper, según la posición señala donde entrara la fuente de alimentación, si era por USB o alimentación externa, ya fuera batería o cargador. En esta versión (*Duemilanove*) ya se cuenta con detección automática de donde provenga la fuente de alimentación teniendo como prioridad USB, seguido de *jack* de alimentación y por último el puerto *Vin*.

Por otro lado, el puerto USB es la interfaz entre computadora-tablilla, por este medio se transfieren los programas realizados en lenguaje “C” hacia el microcontrolador ATmega168, además que toda la tablilla se puede alimentar también de dicha interfaz USB.

Por su lado el microcontrolador antes mencionado, tiene la capacidad de almacenar 16Kb en memoria flash el código del programa en lenguaje C, aunque 2Kb son usados para el sistema de *boot loader*. Este componente es de los más importantes, porque aquí es donde se ejecuta el programa cargado previamente. Al momento de cargar el programa al microcontrolador, uno de los *LED*'s enciende en el transcurso de la transferencia.

El botón *reset* o de reinicio, al presionarlo se inicializa el programa cargado en el microcontrolador.

Otro componente es un chip FTDI, su función se convertir la interfaz serial a USB esto para establecer la comunicación del microcontrolador que utiliza el puerto serial, con el puerto USB que se conecta a la computadora, esto facilita la transferencia de programa que anteriormente se haya realizado en el software de *Arduino*.

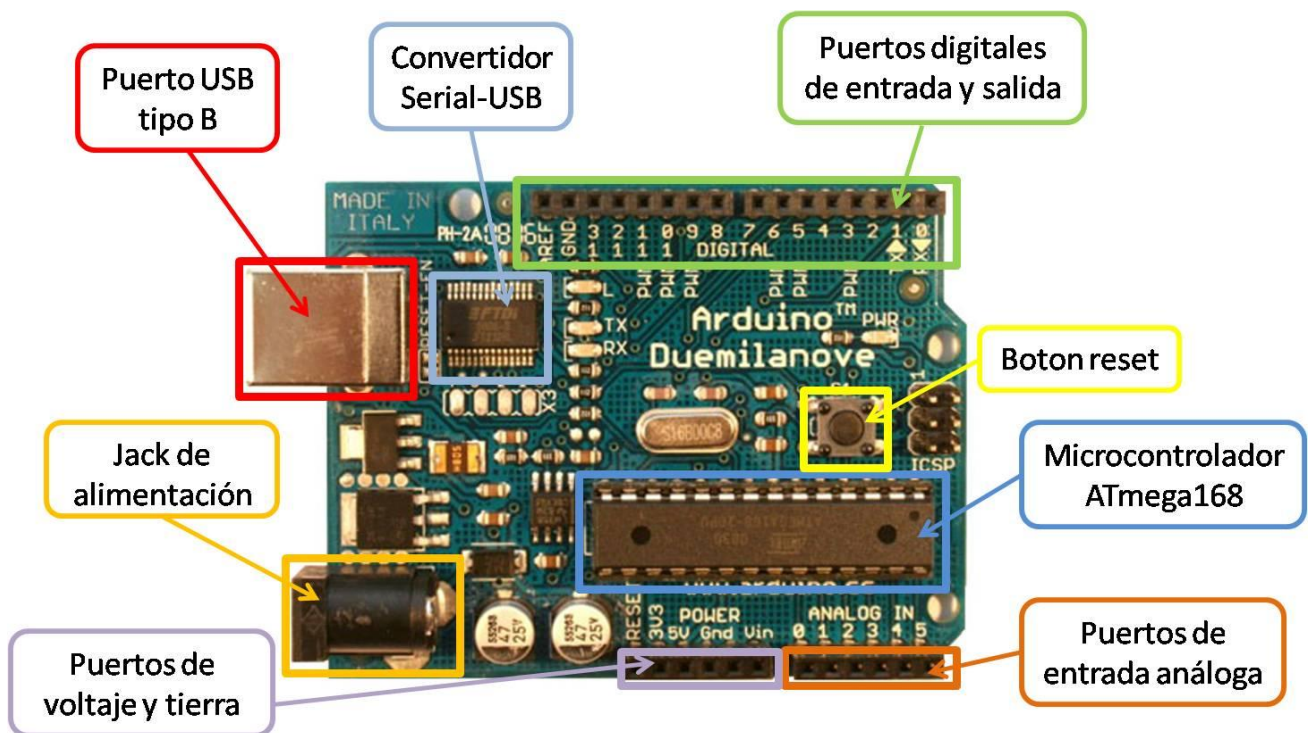


Ilustración 1. Componentes de Arduino

2.6.2 Versiones de Arduino

Los diferentes tipos de tablillas dependen de los elementos que lo integran, entre las cuales están:

- Duemilanove
- Diecimilla
- Placa de prototipos
- Placa serial
- Placa serial una capa
- Autónoma
- Nano
- LilyPad
- Mini
- Mini USB Adapter
- Pro y Pro mini
- Placa SMD
- Bluetooth

2.6.3 Software Arduino

Para codificar el programa y cargarlo al microcontrolador, este se desarrolló en el software de *Arduino*. Para este proyecto se utilizó la versión para *Windows 0016*. Una vez que el código se halla compilado correctamente se puede cargar al microcontrolador *ATmega168*.

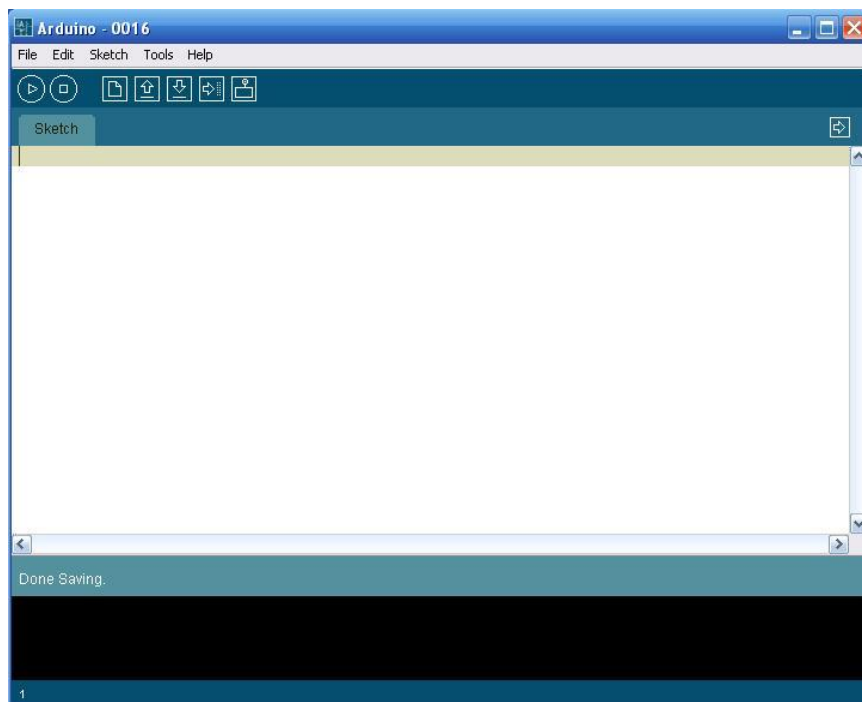


Figura 1. Entorno de Software de Arduino

Botones

-  **Run:** Compila el programa desarrollado.
-  **Stop:** Detiene el monitor Serial.
-  **New:** Abre una nueva área de trabajo.
-  **Open:** Se utiliza para abrir programas realizados.
-  **Save:** Guarda los programas generados.
-  **Upload to I/O Board:** Carga dicho programa al microcontrolador.
-  **Serial Monitor:** Muestra los datos que se envían y reciben por el puerto Serial.

2.7 Lantronix WiMicro (Serial to Wifi)

Servidor HTTP embebido, el cual nos permite alojar páginas web. Los lenguajes compatibles con esta versión esta tablilla son *Java*, *JavaScript*, *HTML*.

Cuenta con 6 módulos de memoria flash (WEB1-WEB6) con un tamaño de 64Kb; en cada uno se puede guardar una página web. Si el cliente realiza una petición al servidor, primero lo busca en WEB1 si no lo encuentra busca en WEB2 y así sucesivamente.

Este Servidor HTTP soporta métodos GET y HEAD.

Para agregar las páginas web se utiliza una herramienta llamada *web2cob.exe*, por medio de esta se envían las páginas a los módulos de memoria utilizando una sintaxis específica desde el *command prompt*; esta será vista más adelante en la sección de estructura interna.

Para acceder al puerto serial del dispositivo es necesario utilizar un *Java Applet*.

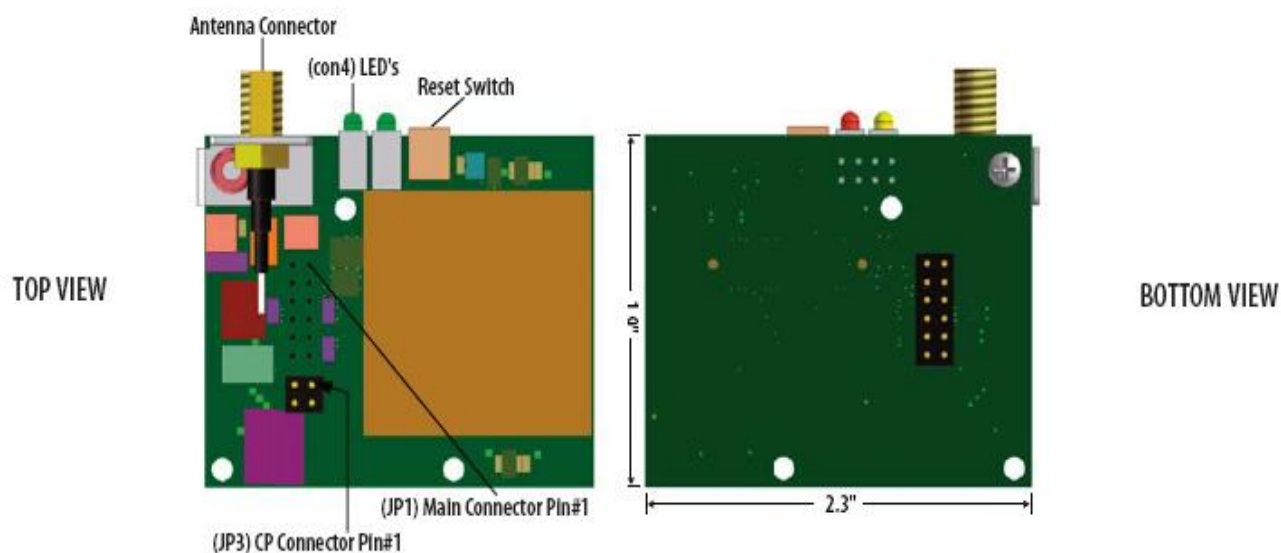


Figura 2. Componentes de Lantronix

2.7.1 Especificaciones

JP1 (2 x 6 Pins) TTL

Pin Function

1 +5V

2 GND

3 RXA (Input)

4 TXA (Output)

5 RTSA (Output)

6 DTRA (Output)

7 CTSA (Input)

8 DCDA (Input)

9 RVA/TX_EN

10 RESET #

11 RXB (Input)

12 TXB (Output)

(JP3) CP Connector

Pin Function

1 CP1

2 CP7

3 CP4

4 CP8

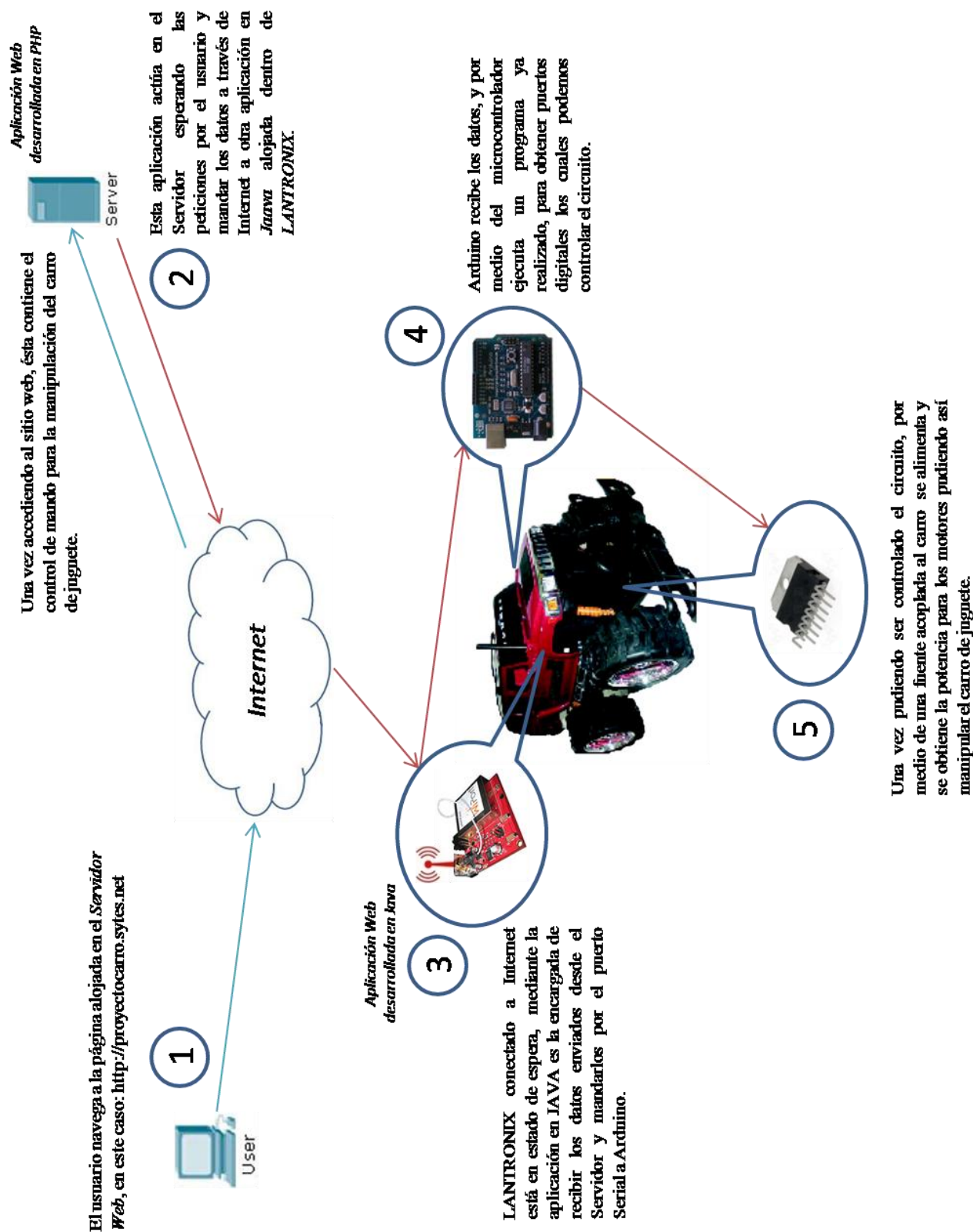


Figura 3. Diagrama de secuencia

CAPITULO 3

DESARROLLO DEL PROYECTO

3.1 Requisitos para el desarrollo.

Tabla 1. Requisitos Cliente y Servidor

	Servidor Web	Cliente
Plataforma	Windows 2000, XP, 2003 Server, Vista	Multiplataforma
Navegador de internet	Internet explorer y Mozilla Firefox	Multiplataforma
Programas previamente instalados	Wamp (Windows-Apache- Mysql-Php) Server	Navegador de internet Maquina Virtual Java

3.2 Desarrollo

Para comenzar el desarrollo del proyecto se parte de una decisión; ya que la idea esta basada en un carro de juguete (CJ), entonces se tiene la posibilidad de adaptar la tablilla al fin buscado, o desmontar el circuito para reemplazarlo por uno más apto.

En definitiva la segunda opción es la ideal y entonces aquí se parte con la primera cuestión siendo de las más importantes, el motor.

En el diseño del circuito base (CB) el motor es la pieza principal, pues este determina las propiedades y límites del mismo. A continuación se presentan las especificaciones del motor seleccionado:

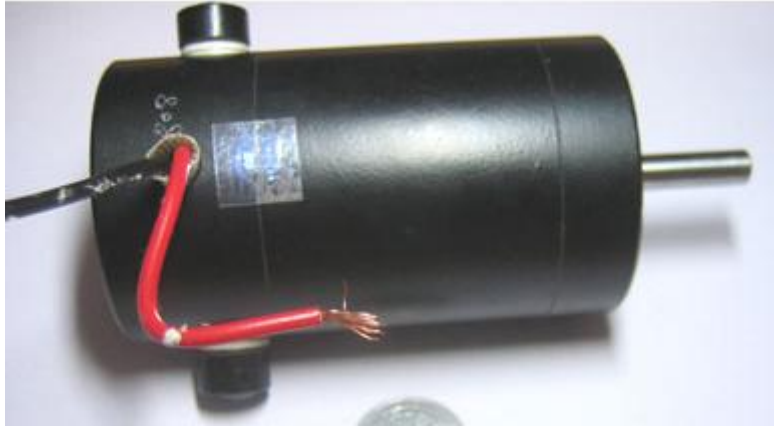


Ilustración 2. Motor eléctrico tomado del CJ

Aquí se observa la necesidad de un circuito que emplee y sea capaz de invertir un voltaje 6V y una corriente de 400mA.

3.2.1 Puente H.

El circuito basado en transistores llamado Puente H (Figura 4) es una configuración de las más utilizadas en el control de motores de CC, cuando es necesario invertir el sentido del giro del motor.

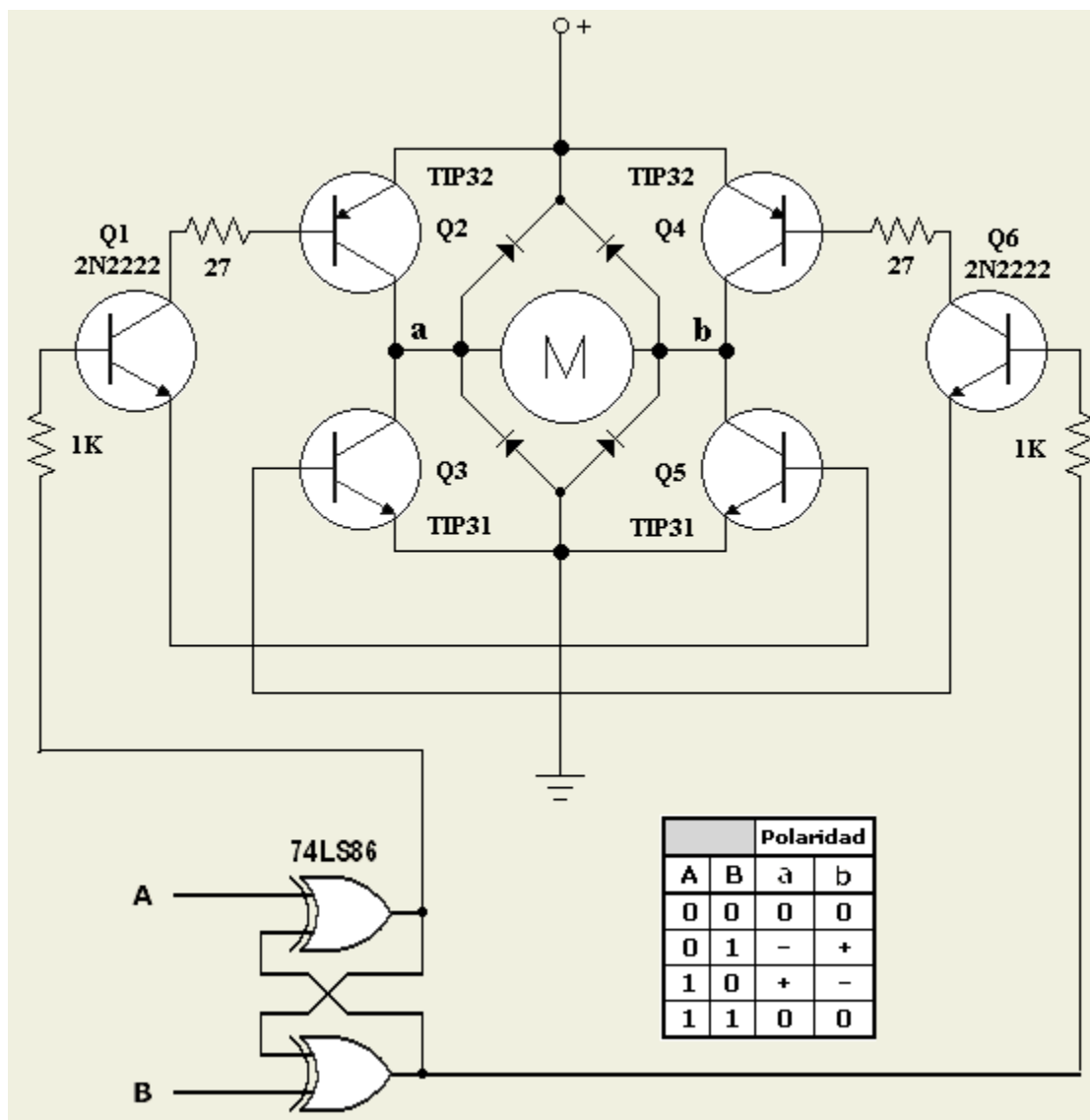


Figura 4. Puente H

Aplicando una señal positiva en la entrada marcada AVANCE se hace conducir al transistor Q1. La corriente de Q1 circula por las bases, de Q2 y Q5, haciendo que el terminal 'a' del motor reciba un positivo y el terminal 'b' el negativo (Figura 5).

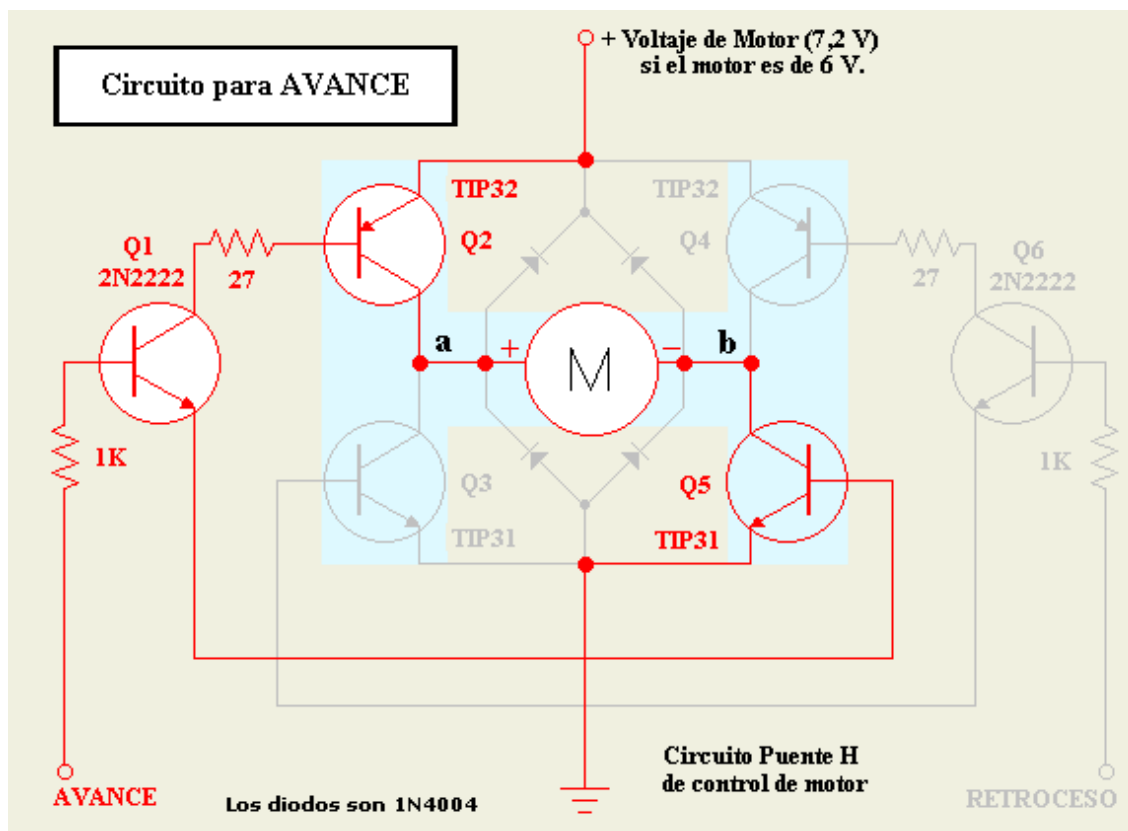


Figura 5. Flujo de corriente en sentido anti horario.

Si en cambio se aplica señal en la entrada RETROCESO, se hace conducir al transistor Q6, que cierra su corriente por las bases, de Q4 y Q3. En este caso se aplica el positivo a la terminal 'b' del motor, y el negativo (tierra) a la terminal 'a' del motor (Figura 6).

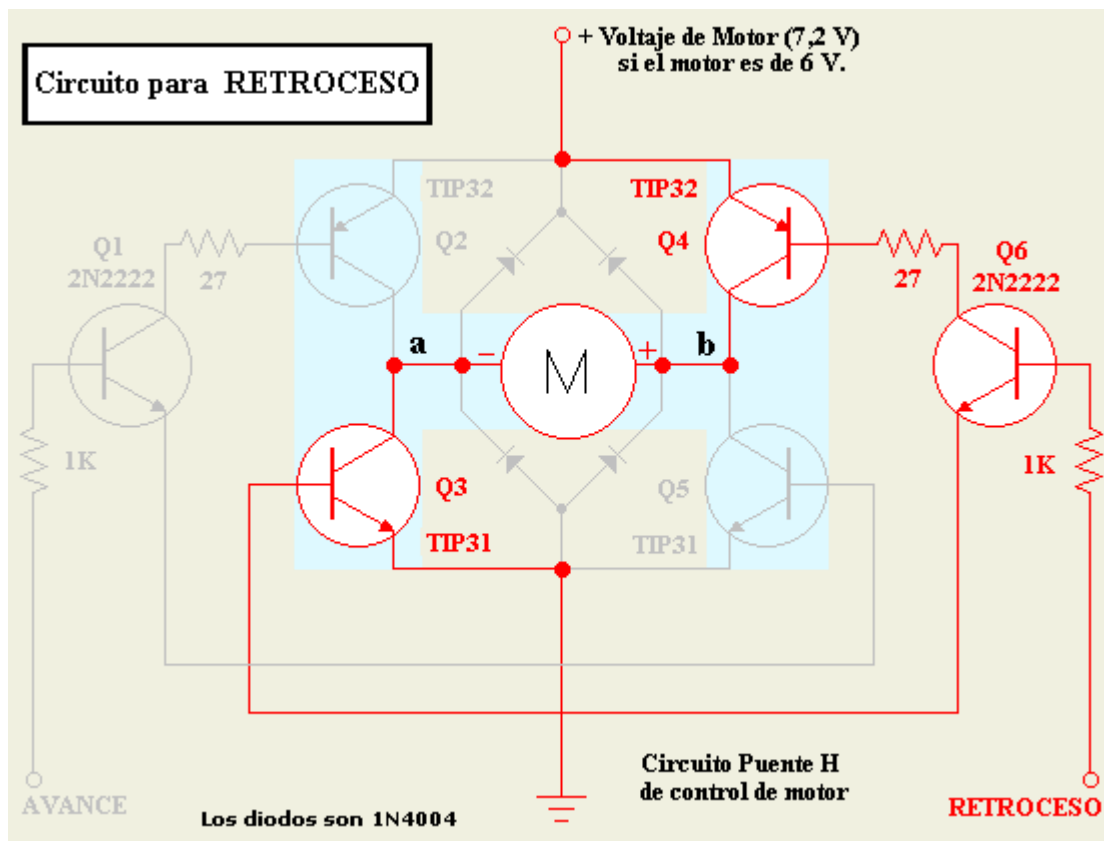


Figura 6. Flujo de corriente en sentido horario.

El elemento 2N2222 que aparece en el diagrama también conocido como PN2222 es un transistor bipolar NPN de baja potencia de uso general, este es capaz de manipular por si solo un motor de forma unidireccional. Entonces, con su doble aplicación en conjunción con los TIP's formando el Puente H se logra un sentido bidireccional del motor.

3.2.2 Integrado L298

El Puente H se desarrolló con pleno éxito. Ahora habría que extender este diseño para la aplicación de dos motores propios del CJ. El integrado L293 nos ahorra peso, espacio y circuitería para el CJ con un diseño más compacto ya que incluye en su interior dos Puentes H. La limitante de este integrado es la baja potencia generada por la corriente

inducida, por lo tanto se recurrió al integrado L298 que nos entrega más potencia y posee las mismas características.

Tabla 2. Tabla comparativa entre L293 y L298

L293

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Supply Voltage	36	V
V_{ss}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{inh}	Inhibit Voltage	7	V
I_{out}	Peak Output Current (non repetitive t = 5ms)	2	A
P_{tot}	Total Power Dissipation at $T_{ground-pins} = 80^{\circ}\text{C}$	5	W
T_{stg}, T_j	Storage and Junction Temperature	-40 to +150	$^{\circ}\text{C}$

L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Power Supply	50	V
V_{ss}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel) - Non Repetitive (t = 100 μ s) - Repetitive (80% on -20% off; t _{on} = 10ms) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^{\circ}\text{C}$)	25	W
T_{op}	Junction Operating Temperature	-25 to +130	$^{\circ}\text{C}$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^{\circ}\text{C}$

La diferencia es clara, el L298 resulta pues, ser el integrado indicado para formar el circuito del CJ. El diseño del circuito para un motor es el siguiente:

3.2.3 Placa USB Duemilanove

Este modelo es el que se usó para el control de los motores del CJ, junto con el integrado L298. *Arduino Duemilanove, es el siguiente al Arduino Diecimila, que a su vez previamente fue Arduino NG y antes Arduino USB.*

Los puertos que se utilizaron se mencionan en la siguiente tabla:

Tabla 3. Puertos de Arduino utilizados.

Puerto Digital 13 Puerto Digital 12	Estos puertos van de la mano, dependiendo de las salidas que nos ofrezcan se transmitirán a los pines 10 y 12 del integrado L298N. Las posibles salidas son 0-1,1-0,0-0,1-1 respectivamente.
Puerto Digital 11 Puerto Digital 10	Estos puertos van de la mano, dependiendo de las salidas que nos ofrezcan se transmitirán a los pines 5 y 7 del integrado L298N. Las posibles salidas son 0-1,1-0,0-0,1-1 respectivamente.
Puerto Digital 0 (Rx)	Aquí recibimos los datos desde <i>Lantronix</i> .
Vin	Alimentación de Arduino (5v).
Ground	Común de todo el circuito
5V Output	Activación para el L298N. Hacia los pines 6,9 y 11.

3.2.4 Programa para controlar el L298N.

Este se realizo en el Software Arduino con lenguaje “C”, para el control del circuito del CJ, y posteriormente fue cargado al microcontrolador *ATmega168*.

Por medio de una función se rastrea el valor del puerto *Rx* que recibe de *Lantronix*, en la tablilla *Arduino*. Según el dato obtenido se crean una serie de condiciones y con estas se habilitan los puertos digitales de salida mencionados anteriormente en la tabla 3, que van hacia el integrado L298N. El código del programa se muestra en el apéndice A.

3.2.5 Estructura Interna

Para implementar la pagina principal (interfaz), en este caso se utilizó el sistema operativo *Windows Server 2003*, con el cual podemos alojar páginas web utilizando *IIS* (Internet Information Services).

Debido a que *IIS* no permite ejecutar las rutinas en *PHP* se remplazó por *WAMP* Server.

Wamp provee los 4 elementos necesarios para un servidor web: un sistema operativo *windows*, un manejador de base de datos, un software para servidor Web (Apache) y un software de programación script web (PHP).

3.2.5.1 Servidor Web

Index.php es la pagina principal del servidor web, para acceder a él se creó un dominio con la dirección *http://proyectocarro.sytes.net*. Los principales *frames* están compuestos por:

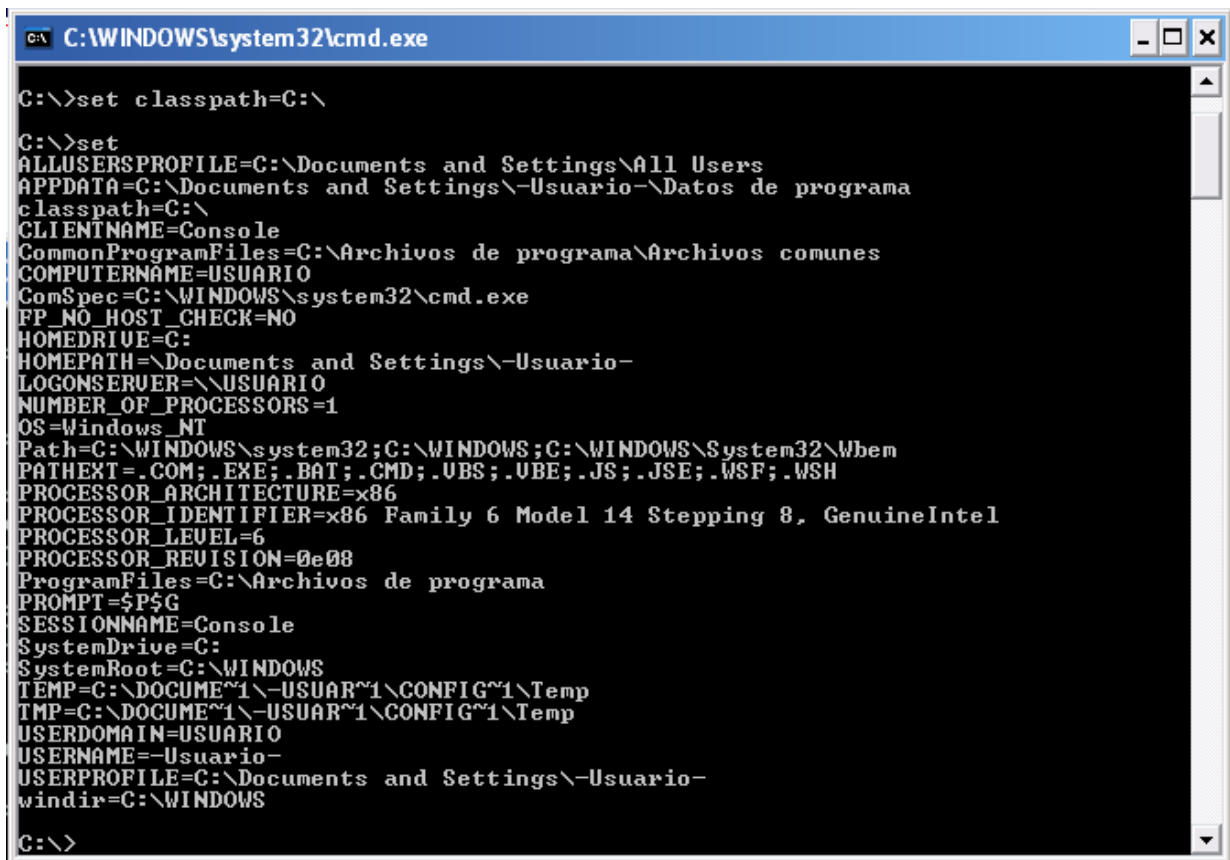
- a) *Frame control*: Es este *frame* se carga la página control.php que es la que envía la URL a la página Web de *Lantronix* (pagelantronix.html) para que esta la procese posteriormente. El envío se realiza por medio del método GET en cada “*submit*” desde dentro de un formulario.
- b) *Frame Lantronix*: Al realizar el “submit” desde mando.php, es en este *frame* donde se carga la pagina de Lantronix (pagelantronix.html) que es la que utiliza la URL y la procesa para después mandar el dato al puerto serial.

Es aquí también donde se puede visualizar el estado de la conexión TCP/IP entre el *Servidor Web* y *Lantronix*.

- c) *Frame cámara*: Este *frame* tiene como finalidad visualizar la dirección del CJ, por medio de una cámara IP AirLink101-AIC250W que se acopló al mismo.

Por parte de *Lantronix*, está compuesto por 2 clases implementadas en un *applet* de *Java*:

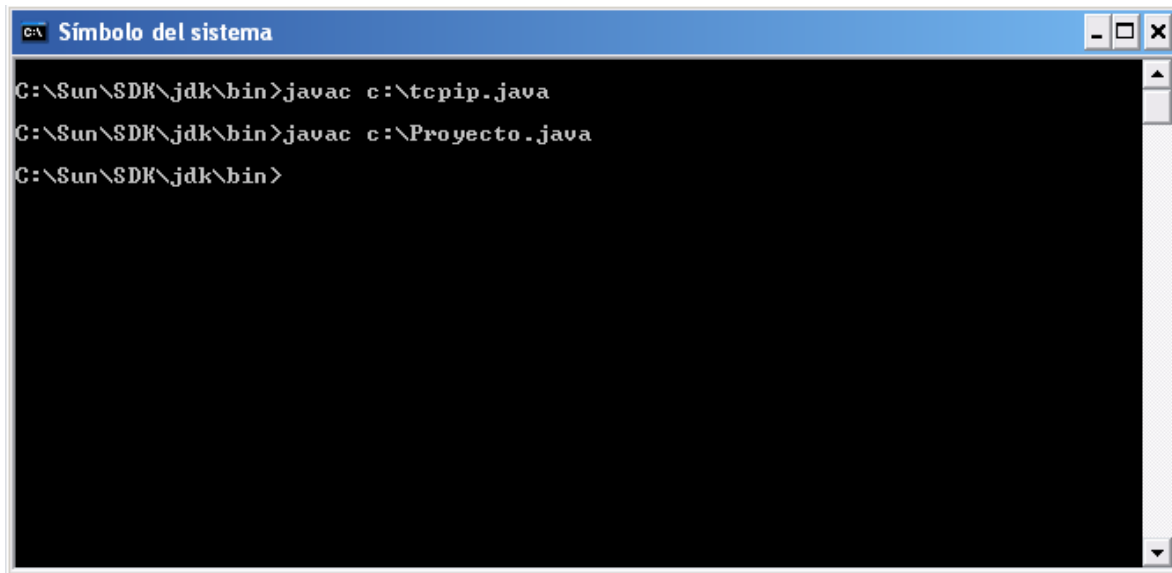
Primero hay que cambiar el *classpath* a raíz de C:\ con el comando “*set classpath*” desde el *command prompt*, ya que aquí es donde colocamos las clases que se van a utilizar como se muestra en la *figura 9*.

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt shows the command "C:\>set classpath=C:\\" being entered. Below this, a list of system environment variables is displayed, including ALLUSERSPROFILE, APPDATA, classpath, CLIENTNAME, CommonProgramFiles, COMPUTERNAME, ComSpec, FP_NO_HOST_CHECK, HOMEDRIVE, HOMEPATH, LOGONSERVER, NUMBER_OF_PROCESSORS, OS, Path, PATHEXT, PROCESSOR_ARCHITECTURE, PROCESSOR_IDENTIFIER, PROCESSOR_LEVEL, PROCESSOR_REVISION, ProgramFiles, PROMPT, SESSIONNAME, SystemDrive, SystemRoot, TEMP, TMP, USERDOMAIN, USERNAME, USERPROFILE, and windir. The prompt "C:\>" is visible at the bottom.

```
C:\>set classpath=C:\\  
  
C:\>set  
ALLUSERSPROFILE=C:\Documents and Settings\All Users  
APPDATA=C:\Documents and Settings\-Usuario-\Datos de programa  
classpath=C:\\  
CLIENTNAME=Console  
CommonProgramFiles=C:\Archivos de programa\Archivos comunes  
COMPUTERNAME=USUARIO  
ComSpec=C:\WINDOWS\system32\cmd.exe  
FP_NO_HOST_CHECK=NO  
HOMEDRIVE=C:  
HOMEPATH=\Documents and Settings\-Usuario-  
LOGONSERVER=\\USUARIO  
NUMBER_OF_PROCESSORS=1  
OS=Windows_NT  
Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem  
PATHEXT=.COM;.EXE;.BAT;.CMD;.UBS;.UBE;.JS;.JSE;.WSF;.WSH  
PROCESSOR_ARCHITECTURE=x86  
PROCESSOR_IDENTIFIER=x86 Family 6 Model 14 Stepping 8, GenuineIntel  
PROCESSOR_LEVEL=6  
PROCESSOR_REVISION=0e08  
ProgramFiles=C:\Archivos de programa  
PROMPT=$P$G  
SESSIONNAME=Console  
SystemDrive=C:  
SystemRoot=C:\WINDOWS  
TEMP=C:\DOCUME~1\-USUAR~1\CONFIG~1\Temp  
TMP=C:\DOCUME~1\-USUAR~1\CONFIG~1\Temp  
USERDOMAIN=USUARIO  
USERNAME=-Usuario-  
USERPROFILE=C:\Documents and Settings\-Usuario-  
windir=C:\WINDOWS  
  
C:\>
```

Figura 9. Direccionar la ruta de *classpath*

El código *tcpip.java*, y *proyecto.java* (apéndice A) se compila uno por uno en el compilador de java *Javac* contenido en el SDK. Se compila uno por uno como se muestra en la *figura 10*.



```
C:\Sun\SDK\jdk\bin>javac c:\tcpip.java
C:\Sun\SDK\jdk\bin>javac c:\Proyecto.java
C:\Sun\SDK\jdk\bin>
```

Figura 10. Compilación de archivos con extensión .java

Aquí, ya se cuenta con las clases que se enviarán a la memoria Flash WEB1 de *Lantronix*, solo falta llamar el *applet* de java desde un HTML (Apéndice A).

El siguiente paso es convertir todas las clases y el archivo *.HTML* a un archivo *.cob* para después enviarlo a *Lantronix*. En este caso se le nombró “*Proyecto.cob*”. Previamente se configuró Lantronix por el puerto COM7 para crear un puente *serie-red* utilizando un software libre llamado *PuTTY* la siguiente manera:

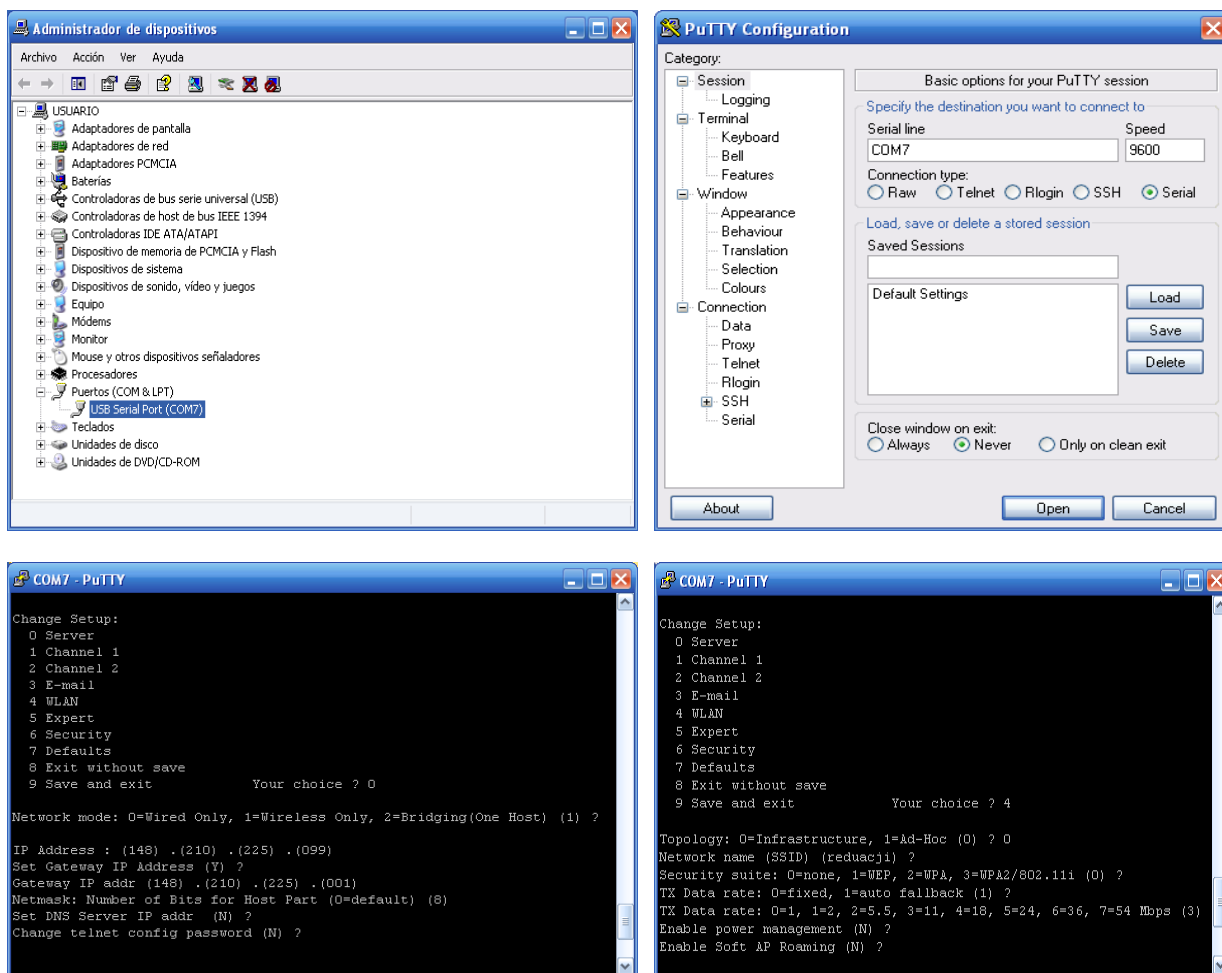
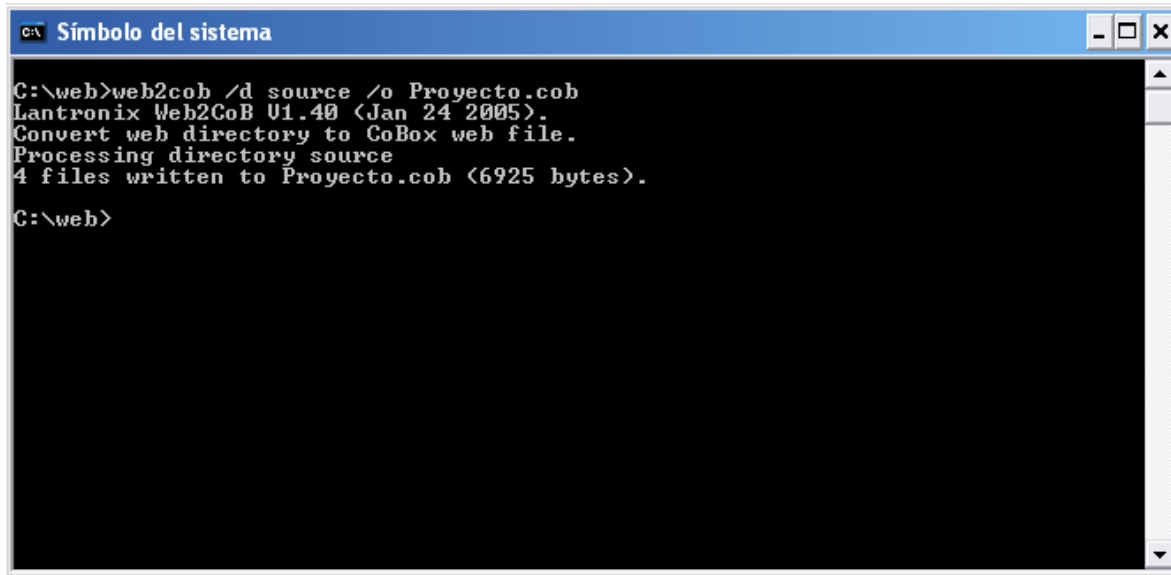


Figura 11. Pasos para la configuración de Lantronix.

Primero se identifica el número de puerto serial de *Lantronix* en este caso COM7, para después crear la conexión serial con la ayuda de *PuTTY* y entrar al menú de configuración.

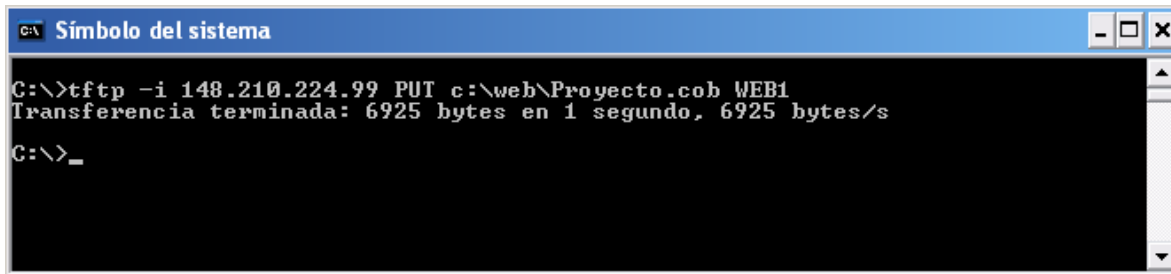
Posteriormente se configura la dirección IP que se le asignará a *Lantronix* y el nombre de la red donde se localiza (Figura 11).

La forma de cómo se convierten y envían el *Proyecto.cob* a *Lantronix* se muestra en las siguientes figuras:



```
C:\web>web2cob /d source /o Proyecto.cob
Lantronix Web2CoB V1.40 (Jan 24 2005).
Convert web directory to CoBox web file.
Processing directory source
4 files written to Proyecto.cob (6925 bytes).
C:\web>
```

Figura 12. Conversión a archivo “.cob”



```
C:\>tftp -i 148.210.224.99 PUT c:\web\Proyecto.cob WEB1
Transferencia terminada: 6925 bytes en 1 segundo, 6925 bytes/s
C:\>_
```

Figura 13. Transferencia de Proyecto.cob a Lantronix

3.2.6 Circuito final

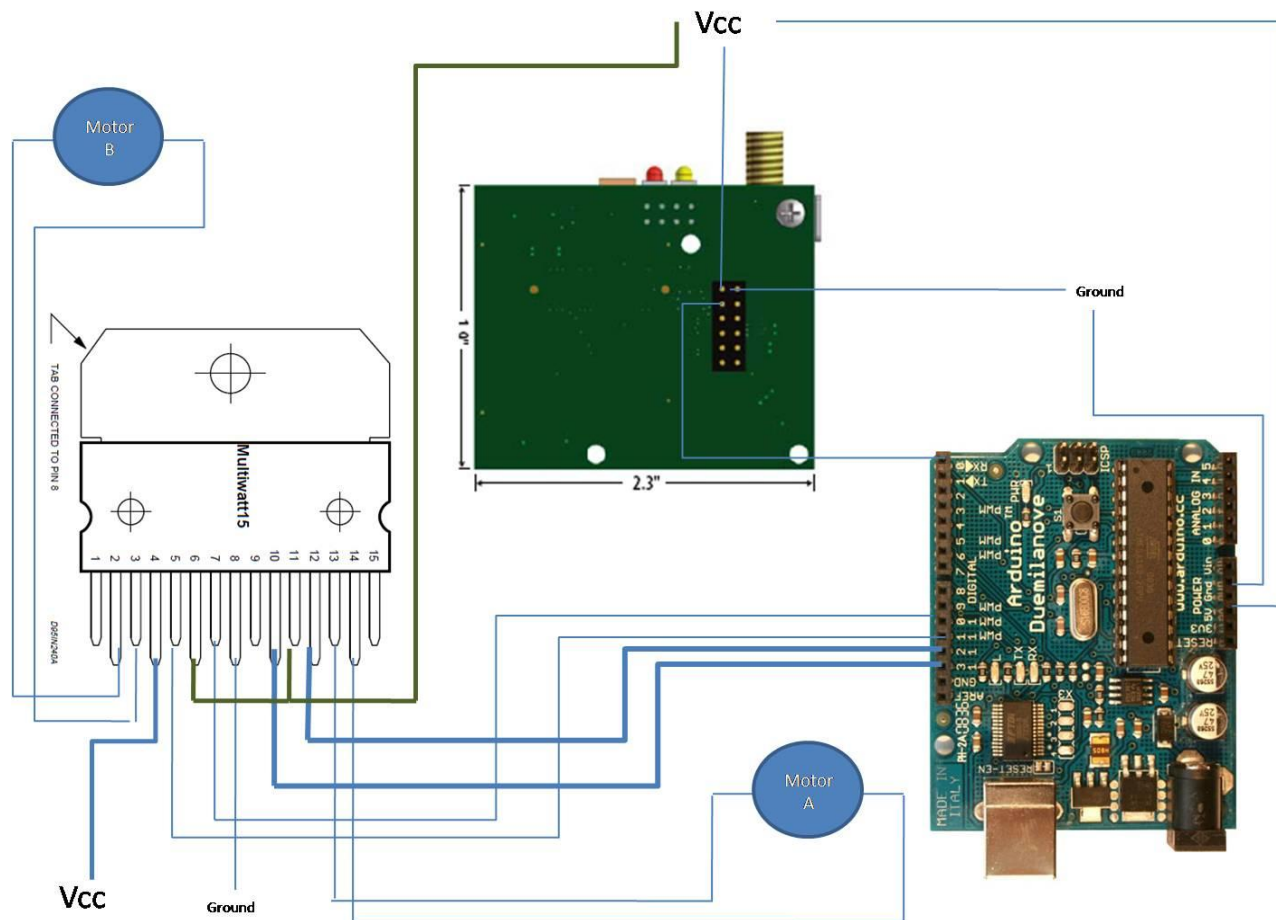


Figura 14. Circuito final para la manipulación del CJ

3.2.6.1 Alimentación del circuito final

En este diagrama podemos observar la interconexión de los diferentes pines del integrado L298N interconectados con *Arduino* y *Lantronix*, armonizados de tal forma que al ponerlo en operación nos da como resultado el movimiento del CJ. Por su parte *Arduino* se alimenta con una batería de 9V y un regulador de voltaje de 5V. *Lantronix* y la cámara web son alimentados a través de una batería de 6V 4A, para su funcionamiento, con sus respectivos reguladores L7805 de voltaje de 5V. Mientras que el integrado L298N utiliza una batería de 6V 1A y utiliza un voltaje lógico tomado desde *Arduino*.

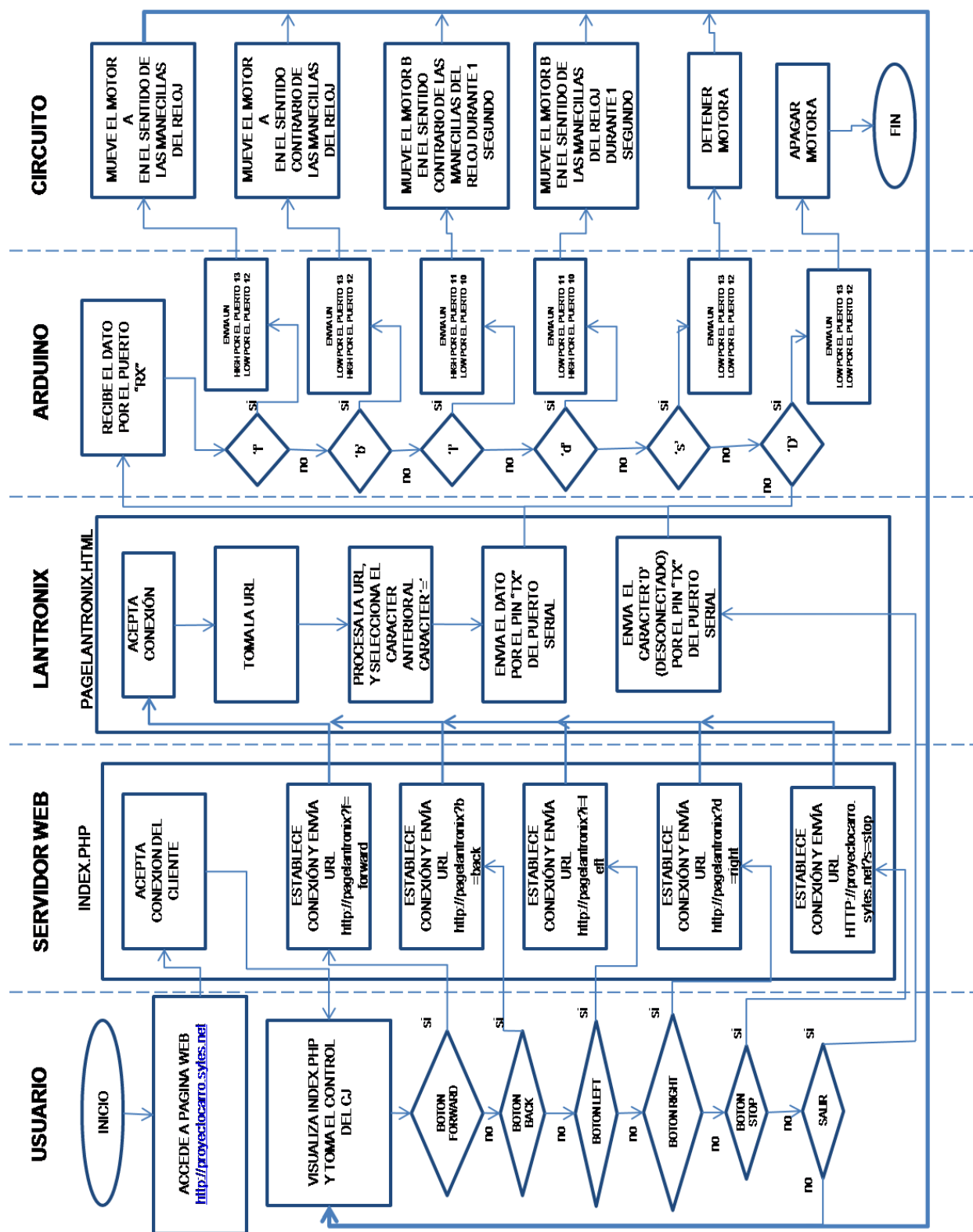


Figura 15. Diagrama del funcionamiento del sistema

3.2.7 Interfaz

- Boton “*Forward*”: Al dar click en este botón, el motor A del CJ gira en el sentido de las manecillas del reloj.
- Botón “*Back*”: Al dar click en este botón, el motor A del CJ gira en el sentido contrario de las manecillas del reloj.
- Botón “*Left*”: Al dar click en este botón, el motor B del CJ gira en el sentido contrario de las manecillas del reloj.
- Botón “*Right*”: Al dar click en este botón, el motor B del CJ gira en el sentido de las manecillas del reloj.
- Botón “*Stop*”: Al dar click en este botón, se detiene el motor A del CJ.

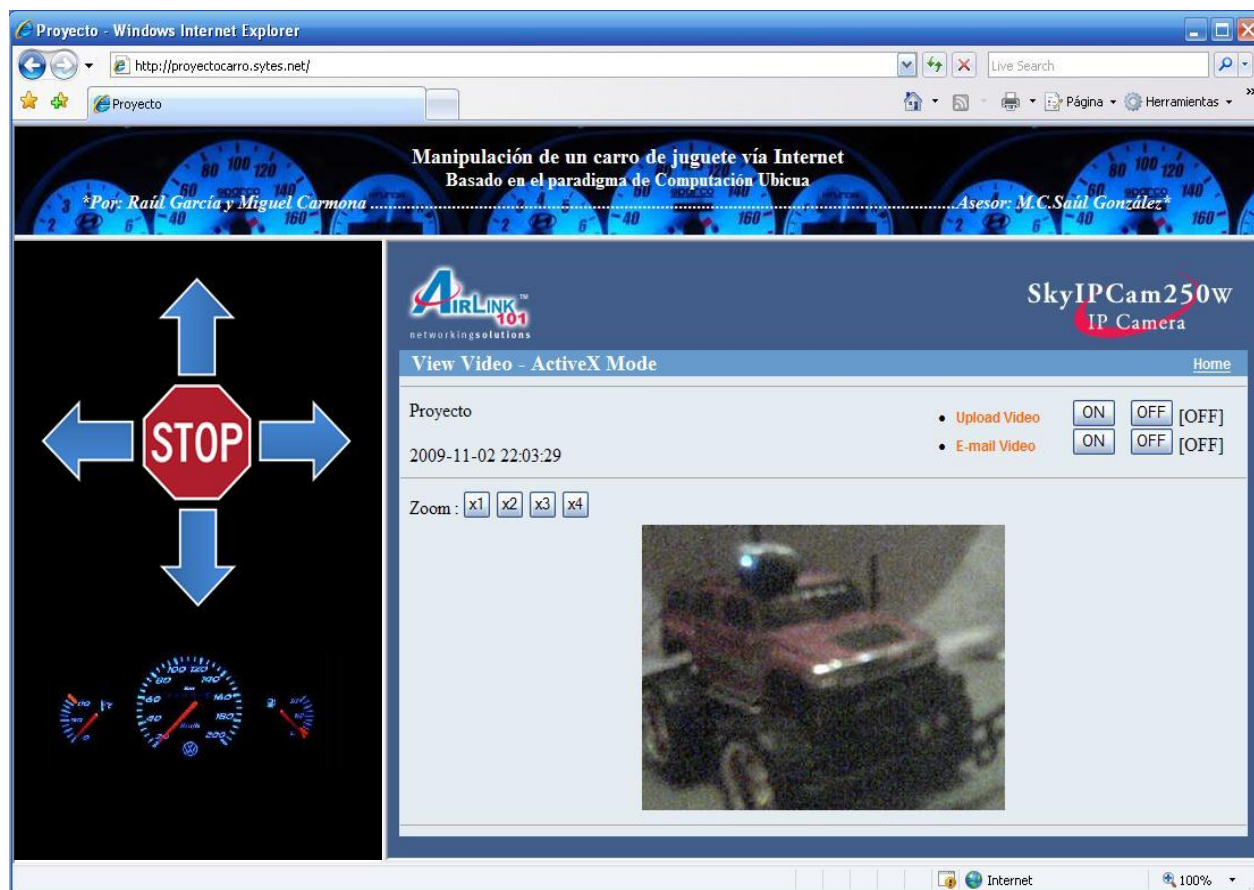


Figura 16. Interfaz

3.3 Experimentos

En esta sección se detallarán algunas pruebas que se realizaron durante el proyecto, antes de llegar al objetivo.

3.3.1 Circuito controlador de los motores

Antes, se mencionaron ya algunos integrados utilizados para tratar de armar el circuito, pero hubo muchos intentos fallidos. Utilizando solo el transistor PN2222 se logró mover un motor pero en un solo sentido y eso no se buscaba, porque es necesario mover los dos motores en los dos sentidos. Se investigó y encontramos el circuito *Puente H*, el cual se armó y funcionó exitosamente, solo que eran necesarios 2 de estos, armando los 2 funcionaría pero ocupaba mucho espacio en el CJ; se investigó para encontrar algún integrado que ya contara con un puente H en su interior. Del l293 y l298, se eligió el l298 por su capacidad y la potencia que entrega.

Ya con el l298 que cuenta con 2 puentes H en su interior, se tuvieron pruebas fallidas debido a la mal integración de los demás componentes del circuito (diodos, resistencias, capacitores) pero al final se lograron controlar las direcciones de los motores a través de las 4 entradas de control del L298N.

3.3.2 Servidor Web

Previamente no se contaba con el conocimiento de cómo utilizar Wamp Server, se recurrió a tutoriales de cómo realizar la instalación adecuada para el buen funcionamiento del servidor; después se realizó la clásica página “*Hola Mundo*” para

posteriormente visualizarla desde la Internet, lo cual ayudo a comprobar que el servidor es público y que se puede acceder a él desde cualquier lugar en la *Web*.

3.3.3 Código

Antes de llegar a esta solución para el problema inicial (controlar el CJ por medio del Internet) se tuvieron muchas opciones por las que se trató de llegar al objetivo. Se comenzó tratando con solo PHP para enviar y recibir los datos pero se encontró que esta versión de *Lantronix* no es compatible con PHP en su estructura interna. Posteriormente se llevó a cabo una prueba de que *Lantronix*, a través del puerto *serial* se conectara con el servidor Web y monitoreara el estado de una variable de la aplicación pero resultó no ser la mejor opción ya que desde éste punto de vista, *Lantronix* sí se conecta al servidor Web pero el *Servidor Web* no observa la dirección de éste, por lo consiguiente no se podrá enviar el dato.

Otra opción fue *Javascript*, por su compatibilidad con *Lantronix*, este parecía ser la solución al problema. Se realizó el siguiente código:

```
<SCRIPT LANGUAGE="JavaScript">

<!--

function tomarValor(){

url = location.href;

val = url.indexOf("?");

val = url.substring(val+1,url.length);

datos = new Array();

datos = val.split("&")
```

```

for (x=0;x<datos.length;x++){

document.write(datos[x] + "<BR>")

document.writeln('<form id="numero" action="" method="post"><p>\r\n<input
type="text" name="numero" id="txt_numero" value="" + datos[x] + "" style="vertical-align:
center; text-align: center"><br></form>');

}}

//-->

</SCRIPT>

```

Funcionó de manera que se podía recibir la *URL* de la interfaz de usuario, pero no se encontró la forma para enviar los datos necesarios al puerto serial de *Lantronix*.

Por lo cual se logró enviar estos datos estableciendo la conexión mediante un *Applet de Java* al puerto serial de *Lantronix*.

3.5 Guía de uso

Teniendo como requisitos mínimos:

- Internet Explorer 6.0 o superior.
- Conexión a Internet por parte del cliente.
- Maquina Virtual de Java instalado previamente en el cliente.
- Conexión de Internet inalámbrica (Wi-Fi), en la parte del CJ.

CAPITULO 4

4.1 Resultados y conclusiones

A continuación se mencionarán los resultados obtenidos y las conclusiones a las que se llegó durante el periodo de realización de este proyecto.

En este proyecto se logró demostrar la interacción humano-computadora, también conocida como *computación ubicua*, implementando hardware y software. Dentro del hardware ubicamos a los dispositivos *Lantronix* y *Arduino*, así como al integrado L298N, mientras que en software está *Arduino*, *Java*, *PHP* y *HTML*.

Dentro de este proyecto se observó que al acceder al *servidor web*, el cual envía datos a *Lantronix*, este a su vez también puede tomar el papel de *cliente*, ya que la página alojada en *Lantronix* espera peticiones y una URL para procesarla en el *applet de Java*.

El CJ funciona en base a las expectativas que se tenían, el tiempo de retardo que se obtiene para el movimiento, después que se da la instrucción en la intranet de la UACJ, es de .5 a 1.5 segundos aproximadamente.

También se logró establecer el control fuera de la red de la universidad; asignando el puerto 80 a *Lantronix* se obtiene una IP pública y se le hace trabajar como Servidor Web, aunque el retardo resultó mayor que en la intranet de la UACJ. Esto es debido a que cada vez que se da una instrucción (adelante, atrás, derecha, izquierda, stop) se realiza un *refresh* a la página de *Lantronix* (pagelantronix.html).

Las pruebas por *Internet* se realizaron con un Router que provee *Internet*, el cual puede ser modificado para tener acceso vía *Internet* a cualquier dispositivo de esa red.

La velocidad máxima del CJ registrada durante las pruebas fue de .7m/s. Mientras que el giro de la dirección del CJ es de 20 grados aproximadamente, tanto para la derecha como para la izquierda. Para mejorar la velocidad y el control de la dirección del CJ habría que sustituirlo por otro vehículo con motores que entreguen mayor potencia y donde la dirección sea mejor.

Tabla 4. Especificaciones físicas del CJ

Peso	2.650Kg.
Dimensiones	41cm de largo x 29cm de ancho
Llantas	Radio de 6.5cm
Motores	Frontal 6v 700mA, trasero 6v 300mA

Por último, otro de los puntos importantes es que objetos cotidianos como lo es un carro de juguete, que está diseñado para ser controlado a distancias pequeñas, se convierte en un juguete controlable desde cualquier lugar sin importar la distancia; se pueden tener múltiples funcionalidades y/o diferentes aplicaciones, tales como monitoreo y seguridad, donde no es necesario que exista una computadora cerca de tal objeto, de aquí el término de *Computación Ubicua*.

4.2 Trabajo relacionado

Un proyecto ya realizado en el Tecnológico de Monterrey, Campus Estado de México, que consiste en controlar el carro con su respectivo control conectado al puerto paralelo de una computadora [15].

Otro proyecto reconocido en México es el realizado por la Dra. Marcela D. Rodríguez Urrea, “*Agentes Autónomos en ambientes de cómputo colaborativos ubicuos*”, donde el proveer de plataformas que faciliten el desarrollo de sistemas de cómputo ubicuo es otro de los retos de esta área. En este sentido, la tesis de la Dra. Rodríguez explora el uso de los agentes autónomos como una abstracción para el diseño y construcción de sistemas de cómputo ubicuo y se proporciona un middleware para facilitar la implementación de los agentes. El objetivo en esa investigación es diseñar e implementar un *middleware* que permita a los desarrolladores manejar algunas de las complejidades asociadas con el desarrollo de sistemas de cómputo ubicuo por medio del uso de agentes autónomos [2].

4.3 Trabajo a futuro

Este proyecto puede ser retomado para subsecuentes mejoras, la visión que se tiene para que pueda ser complementado posteriormente puede resumirse en los siguientes puntos:

El CJ no precisamente debe de ser el que se utiliza en este proyecto, puede ser algún otro que posea motores eléctricos con las capacidades que el integrado L298N necesita para poder moverlos. El CJ incluso podría ser sustituido por un robot, por cámaras con servomotores o por algún otro dispositivo que contenga motores eléctricos que también puedan ser controlados con el integrado L298N.

Por otro lado, se observa en este proyecto que se lograron controlar 2 motores en forma remota, sin embargo el proyecto no está obligado a que sea solo esta la cantidad de motores; gracias a *Arduino*, complementado con el integrado L298N, se puede controlar un mayor número de motores en forma simultanea y con esto se abre la posibilidad para otro tipo de aplicaciones.

Si para 2 motores necesitamos un L298N, podemos controlar 4 motores agregando otro L298N a *Arduino* y así sucesivamente.

Observando el diagrama de secuencia hasta el punto de *Arduino*, se podrían realizar algunas modificaciones al sistema para que este pueda ser utilizado para controlar una diversidad de objetos, sin que necesariamente sean motores [figura 3].

El proyecto podría ser utilizado como base en diferentes campos tales como la industria, la minería, para la detección de minas explosivas que existen alrededor del mundo por causa de las diversas guerras, ayudar a personas con capacidades diferentes, etc. También, no solo para controlar un móvil sino tomar cualquier objeto electrónico y manipularlo remotamente.

Dentro de la industria, puede entrar como un sistema de seguridad y monitoreo, el cual podría ser utilizado para vigilar todas las áreas de una empresa desde cualquier lugar, no siendo necesario que el usuario deba de estar dentro del edificio que se desee monitorear.

En la minería, podría funcionar como un detonador de largo alcance, contribuyendo así a la seguridad de los empleados de un campo minado, previniendo accidentes.

En la detección de minas explosivas actualmente se usan móviles a través de radiofrecuencia por lo que existe un límite de alcance de control en estos móviles.

“Las minas, y los remanentes explosivos de las distintas guerras que han existido en el mundo, que incluyen la artillería inexplorada y artillería explosiva abandonada, representan una amenaza importante para el civil. El departamento de las naciones unidas de los asuntos humanos (UNDHA) determina que hay más de 100 millones de minas que se dispersan a través del mundo y plantean peligros significativos en más de 68 países.

El comité internacional de la cruz roja (CICR) estima que la tarifa de la muerte a causa de minas terrestres excede actualmente a 26000 personas cada año. Se estima que matan a mas de 800 personas y 1200 son mutilados cada mes por las minas en todo el mundo” [16].

Otra visión es para personas con capacidades diferentes, estos podrían tener la facilidad de usarlo para su uso personal como llevar y traer objetos de uso cotidiano y así no depender de otras personas.

ANEXO

Software Arduino

Estructura

- *void setup()*
- *void loop()*

Estructuras de control

- *if*
- *if...else*
- *for*
- *switch case*
- *while*
- *do... while*
- *break*
- *continue*
- *return*

Operadores Aritméticos

- *= (assignment)*
- *+ (addition)*
- *- (subtraction)*
- ** (multiplication)*
- */ (division)*
- *% (modulo)*

Operadores de comparación

- `==` (equal to)
- `!=` (not equal to)
- `<` (less than)
- `>` (greater than)
- `<=` (less than or equal to)
- `>=` (greater than or equal to)

Operadores Booleanos

- `&&` (and)
- `||` (or)
- `!` (not)

Operadores compuestos

- `++` (increment)
- `--` (decrement)
- `+=` (compound addition)
- `-=` (compound subtraction)
- `*=` (compound multiplication)
- `/=` (compound division)

Variables

Son expresiones que pueden ser usadas en programas para almacenar valores.

Constantes

Son valores en particular con especificos significados.

- *HIGH | LOW*
- *INPUT | OUTPUT*
- *true | false*
- *Integer Constants*

Tipos de datos

Las variables pueden tomar diferentes tipos, los cuales se mencionan a continuación:

- *char*
- *byte*
- *int*
- *unsigned int*
- *long*
- *unsigned long*
- *float*
- *double*
- *string*
- *array*
- *void*

Conversión

- *char()*
- *byte()*
- *int()*
- *long()*
- *float()*

Funciones

Puertos digitales I/O

- *pinMode(pin, mode)*
- *digitalWrite(pin, value)*
- *int digitalRead(pin)*

Puertos analogos I/O

- *int analogRead(pin)*
- *analogWrite(pin, value) - PWM*

Entrada y salida avanzados I/O

- *shiftOut(dataPin, clockPin, bitOrder, value)*
- *unsigned long pulseIn(pin, value)*

Tiempo

- *unsigned long millis()*
- *unsigned long micros()*
- *delay(ms)*
- *delayMicroseconds(us)*

Matemáticas

- *min(x, y)*
- *max(x, y)*
- *abs(x)*
- *constrain(x, a, b)*
- *map(value, fromLow, fromHigh, toLow, toHigh)*
- *pow(base, exponent)*
- *sq(x)*
- *sqrt(x)*

Trigonometría

- *sin(rad)*
- *cos(rad)*
- *tan(rad)*

Números aleatorios

- *randomSeed(seed)*
- *long random(max)*
- *long random(min, max)*

Comunicación

- *Serial*

Especificaciones Cámara IP

Connection	IEEE 802.11g
Protocol	TCP/IP, UDP, ARP, ICMP, BOOTP, RARP, DHCP, PPPoE, SMTP, FTP, HTTP, DDNS Support
Camera	Sensor: 1/3" Color CMOS, Resolution: 640x480, Focal Length: 6.0 mm, Aperture: 1.8, S/N Ratio: 57dBm
Video Format	Image Compression: Motion JPEG
Resolution	10fps @ 640x480 (VGA), 25fps @ 320x240 (CIF), 30fps @ 160x112 (QCIF)
Security	Admin/User protected, password authentication, WEP 64/128bit
Power Adapter	5V / 2.5 ^a
Dimensions	5" x 2.75" x 1.3"
Temperature	Operating: 5-40°C (41-104°F) Storage: -25-50°C (-13-122 °F)
Weight	235g
Certification	FCC, CE

Especificaciones del regulador de voltaje L7805

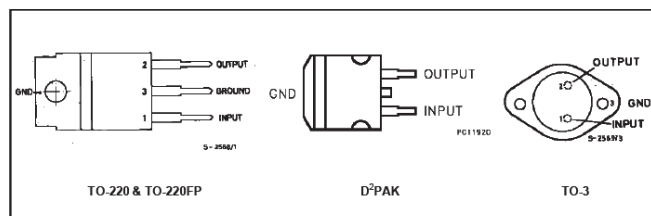
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_i	DC Input Voltage (for $V_o = 5$ to 18V) (for $V_o = 20, 24V$)	35 40	V V
I_o	Output Current	Internally limited	
P_{tot}	Power Dissipation	Internally limited	
T_{op}	Operating Junction Temperature Range (for L7800) (for L7800C)	-55 to 150 0 to 150	°C °C
T_{stg}	Storage Temperature Range	-65 to 150	°C

THERMAL DATA

Symbol	Parameter	D ² PAK	TO-220	TO-220FP	TO-3	Unit
$R_{th(jc)}$	Thermal Resistance Junction-case Max	3	3	5	4	°C/W
$R_{th(ja)}$	Thermal Resistance Junction-ambient Max	62.5	50	60	35	°C/W

CONNECTION DIAGRAM AND ORDERING NUMBERS (top view)



APÉNDICE A

Código Arduino

```
#include <SoftwareSerial.h>
```

```
#define rxPin 0
```

```
#define txPin 1
```

```
SoftwareSerial raul=SoftwareSerial(rxPin,txPin);
```

```
int ledPin = 13;          // Entradas de selección para motor trasero
```

```
int ledPin2=12;
```

```
int ledPin3=11;          //Entradas de selección para motor delantero
```

```
int ledPin4=10;
```

```
int edo=0;
```

```
void setup()
```

```
{
```

```
  pinMode(rxPin, INPUT);
```

```
  pinMode(txPin, OUTPUT);
```

```
  pinMode(ledPin, OUTPUT);
```

```
  pinMode(ledPin2, OUTPUT);
```

```
  pinMode(ledPin3, OUTPUT);
```

```
  pinMode(ledPin4, OUTPUT);
```

```
  raul.begin(9600);
```

```
}
```

```

void loop()
{
    Serial.flush();
    char inChar;

    int valor=0;
    valor=raul.read();
    inChar=(char) valor;
    if(inChar=='f')
    {
        forward();
    }
    if(inChar=='b')
    {
        back();
    }

    if(inChar=='l')
    {

        left();
    }
    if(inChar=='r')
    {
        right();
    }
    if(inChar=='s')
    {
        stop();

    }
}

```

```
if(inChar=='w')
{
  lforward();
}
```

```
if(inChar=='x')
{
  rforward();
}
```

```
if(inChar=='y')
{
  lback();
}
```

```
if(inChar=='z')
{
  rforward();
}
```

```
Serial.flush();
}
```

```
void forward()
{
```

```
  stop();
  if(edo==2)
  {
    digitalWrite(ledPin3,HIGH);
    delay(80);
    digitalWrite(ledPin3,LOW);
    edo=0;
  }
  if(edo==1)
```

```

    {
    digitalWrite(ledPin4,HIGH);
    delay(80);
    digitalWrite(ledPin4,LOW);
    edo=0;
    }
    digitalWrite(ledPin2,LOW);
    digitalWrite(ledPin,HIGH);
    }
void back()
{
    stop();
    if(edo==2)
    {
    digitalWrite(ledPin3,HIGH);
    delay(80);
    digitalWrite(ledPin3,LOW);
    edo=0;
    }
    if(edo==1)
    {
    digitalWrite(ledPin4,HIGH);
    delay(80);
    digitalWrite(ledPin4,LOW);
    edo=0;
    }

    digitalWrite(ledPin,LOW);
    digitalWrite(ledPin2,HIGH);

}
void left()
{

```

```
digitalWrite(ledPin3,HIGH);  
  delay(500);  
  digitalWrite(ledPin3,LOW);  
  digitalWrite(ledPin4,HIGH);  
  delay(80);  
digitalWrite(ledPin4,LOW);  
}
```

```
void right()  
{  
  digitalWrite(ledPin4,HIGH);  
  delay(500);  
digitalWrite(ledPin4,LOW);  
  digitalWrite(ledPin3,HIGH);  
  delay(80);  
  digitalWrite(ledPin3,LOW);  
}
```

```
void stop()  
{  
  digitalWrite(ledPin,LOW);  
  digitalWrite(ledPin2,LOW);  
  delay(1000);  
  
}
```

```
void lforward()  
{ stop();  
  edo=1;  
  digitalWrite(ledPin2,LOW);  
  digitalWrite(ledPin,HIGH);  
  
  digitalWrite(ledPin3,HIGH);  
  delay(1000);  
}
```

```

digitalWrite(ledPin3,LOW);
}
void rforward()
{ stop();
  edo=2;
digitalWrite(ledPin2,LOW);
  digitalWrite(ledPin,HIGH);

digitalWrite(ledPin4,HIGH);
  delay(1000);
  digitalWrite(ledPin4,LOW);
}
void lback()
{
  stop();
  edo=1;
  digitalWrite(ledPin,LOW);
  digitalWrite(ledPin2,HIGH);
  digitalWrite(ledPin3,HIGH);
  delay(1000);
  digitalWrite(ledPin3,LOW);
}
void rback()
{
  stop();
  edo=2;
  digitalWrite(ledPin,LOW);
  digitalWrite(ledPin2,HIGH);
  digitalWrite(ledPin4,HIGH);
  delay(1000);
  digitalWrite(ledPin4,LOW);
}

```

Index.php

```
<HTML>
```

```
<HEAD><TITLE>Proyecto</TITLE></HEAD>
```

```
<FRAMESET ROWS=15%,*>
```

```
<FRAME SRC="titulo.php" NAME="titulo">
```

```
<FRAMESET COLS=325,*>
```

```
<FRAME SRC="Control.php">
```

```
<FRAMESET ROWS="100%,0%">
```

```
<FRAME SRC=http://148.210.224.101/Aviwm.htm  
name="camara">
```

```
<FRAME name="paglantronix">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

```
</FRAMESET>
```

```
</HTML>
```

Control.php

```
<html>

<head>

<title>Control.php</title>

</head>

<body>

<?php

$formulario = <<<INICIO

<center><input type="image" src="/images/adelante.jpg" name="f"
value="Adelante" ></center>

<input type="image" src="/images/left.png" name="i" value="Left" >

<input type="image" src="/images/stop.png" name="s" value="Stop" >

<input type="image" src="/images/right.png" name="d" value="Right" >

<center><input type="image" src="/images/back.png" name="b" value="Back"
></center>

INICIO;

echo $formulario;

?>

</body>

</html>
```

PageLantronix.html

```
<HTML>

<BODY><CENTER>

<APPLET CODE="proyecto.class" WIDTH="862" CODEBASE="."
HEIGHT="512"></APPLET>

</CENTER>

</BODY></HTML>
```

Proyecto.class

```
String path = getDocumentBase().toString();

int aux=0;

for(int i = 0;path.charAt(i)!='';i++)

{

    aux=i;

}

gtp.send(Character.toString(path.charAt(aux)));
```

tcpip.class

```
import java.*;
import java.lang.*;
import java.net.*;
import java.util.*;
import java.io.*;

public class tcpip
{
    protected Socket s = null;
    public DataInputStream dis = null;
    protected DataOutputStream dos = null;

    public tcpip(InetAddress ipa, int port)
    {
        Socket s1 = null;
        try {
            // Abre el socket
            s1 = new Socket(ipa.getHostAddress(), port);
        }
        catch (IOException e) {
            System.out.println("Error al abrir el Socket");
            return;
        }
        s = s1;
        try {
            dis = new
DataInputStream(new BufferedInputStream(s.getInputStream()));
        }
        catch(Exception ex) {
            System.out.println("Error al crear el stream de salida ");
        }
    }
}
```

```

        try    {                                // Creacion de stream de salidas
            dos = new DataOutputStream(new
BufferedOutputStream(s.getOutputStream()));
        }
        catch(Exception ex) {
            System.out.println("Error al crear el stream de salida);
        }
    }

    public synchronized void disconnect()
    {
        if (s != null) {
            try {
                s.close();
            }
            catch (IOException e){}
        }
    }

    public synchronized void send(byte[] temp)
    {
        try {
            dos.write(temp, 0, temp.length);
            dos.flush();
        }
        catch(Exception ex) {
            System.out.println("Error sending data : " + ex.toString());
        }
    }
}

```

```

public synchronized void send(byte[] temp, int len)
{
    try {
        dos.write(temp, 0, len);
        dos.flush();
    }
    catch(Exception ex) {
        System.out.println("Error sending data : " + ex.toString());
    }
}

```

```

public synchronized void send(String given)
{
    int length = given.length();
    byte[] retvalue = new byte[length];
    char[] c = new char[length];
    given.getChars(0, length, c, 0);
    for (int i = 0; i < length; i++) {
        retvalue[i] = (byte)c[i];
    }
    send(retvalue);}

```

```

public synchronized byte[] receive()
{
    byte[] retval = new byte[0];

    try {
        while(dis.available() == 0); }
    catch (IOException e){}
    try {

```

```

        retval = new byte[dis.available()];
    } catch (IOException e){}
    try {
        dis.read(retval);
    } catch (IOException e){}
    return(retval);
}
public int available()
{
    int avail;
    avail = 0;
    try {
        avail = dis.available();} catch (IOException e) {}

    return(avail);}
}

```

BIBLIOGRAFÍA

- [1] <http://computacionubicuayagentes.blogspot.com>
- [2] Rodríguez Urrea, Marcela D., “Agentes autónomos en ambientes de cómputo colaborativos ubicuos”, *CICESE Research Center*
- [3] Pomares Hernández Saúl E., “Computación Ubicua; un gran desafío”, INAOE
- [4] <http://www.webtaller.com/maletin/articulos/que-es-tecnologia-wi-fi.php>
- [5] PHP Solutions “Dynamic Web Design Made Easy”, David Powers, Editorial Board, Edition
- [6] <http://wampserver.com>
- [7] <http://www.lantronix.com>
- [8] <http://www.arduino.com>
- [9] Programación En Java Para Internet, Editorial Paraningo, Michael D. Thomas, Pratik R.Patel, Alan D. Hudson. Donald A. Ball Jr.
- [10] Java curso de Programación 3ª edición, Fco. Javier Ceballos. Alfaomega
- [11] Sistemas Digitales, Principios y Aplicaciones, Ronald J. Tocci, 6ta ed. Pág. 4, 1996
- [12] <http://www.larevista.com.mx>

[13] El Imperio Digital, Leandro Zanoni, Pág. 22, 2008

[14] Mark Weiser, The Computer for the 21st Century, 1991

[15] <http://webdiee.cem.itesm.mx/web/servicios/archivo/trabajos/digitales/carro/menu.html>

[16] Habib Dr. Maki K., "Humanitarian Demining", Saga University , Japan

[17] Real Academia de la Lengua Española

Herrera Lozada M. C. Juan Carlos, "Control de iluminación vía Internet," Investigador del
CIDETEC IPN

http://www.airlink101.com/download/download_links/aic250w_manual.pdf

Colorado Enrique, "Implementacion de un puente serie usando convertidores serie-red",
Septiembre 2008