MANIPULACIÓN DE OBJETOS EN AMBIENTES TRIDIMENSIONALES PARA REALIDAD VIRTUAL

Autores:
Fausto Freire[1]
Fernando Jiménez
Darwin Meneses
Silvia García

1. F. Freire
Facultad de Ciencias de la Ingeniería, Universidad Tecnológica Equinoccial, Av. Occidental y Av. Mariana de Jesús, Apartado 17-01-2764, Quito – Ecuador
ffreire@ute.edu.ec

REVISTA DE INVESTIGACIÓN CIENTÍFICA

Resumen

El presente artículo trata sobre el desarrollo de un Sistema para la manipulación de objetos virtuales, en un ambiente tridimensional, para inversión virtual utilizando un guante de navegación y el casco virtual visor VR1280.

El Sistema está compuesto por dos subsistemas: navegación y software. El subsistema de navegación consta de un acelerómetro, microcontrolador y conversor análogo/digital; mediante el cual el usuario interactúa con el ambiente virtual. El subsistema software está desarrollado en Java 3D y simula el ambiente virtual donde se alojan los objetos tridimensionales; la comunicación de los subsistemas se realiza a través del puerto serial del computador.

El Sistema permite interactuar al usuario con un ambiente virtual, siendo uno de los principales problemas la velocidad de reacción del sistema ante la manipulación, en parte debido a la arquitectura y tecnologías utilizadas.

Palabras clave: Realidad virtual, Java 3d, navegación virtual.

Abstract

This article discusses the development of a system for manipulating virtual objects in a three-dimensional environment, using a glove and helmet visor VR1280.

The system consists of two subsystems: navigation and software. The navigation subsystem consists of an accelerometer, microcontroller and analog / digital converter, whereby the user interacts with the virtual environment. The software subsystem is developed in Java 3D and simulate the virtual environment, which contains three-dimensional objects, communication subsystems is done through the computer serial port.

The system allows the user to interact with a virtual environment, one of the main problems the speed of system response to manipulation, in part due to the architecture and technologies used.

Keyboards: Virtual Reality, Java 3D, virtual navigation

Recibido: Mayo 2011

Aprobado: Julio 2011

INTRODUCCIÓN

El avance de las Tecnologías de la Información marca nuevos retos, cambios en las costumbres y forma de vida de las personas que se harán más evidentes en un futuro próximo.

El desarrollo y la utilización de Realidad Virtual (RV) han demostrado ser el camino a seguir por las tendencias actuales de desarrollo tecnológico, que permitirá entre otras cosas maximizar la utilización de recursos con muy buenos resultados.

Realidad Virtual (RV), término atribuido a Jaron Lanier en 1989, investigador de VPL Research, según el cual, RV es un interface que permite sumergir al usuario en un ambiente artificial mediante el engaño de los sentidos y una interactividad natural y es una tecnología de simulación avanzada que utiliza una programación de imágenes o gráficos en tres dimensiones de alta velocidad y software especial que desarrolla la acción en tiempo real [1]. Sin embargo, otros autores difieren de esta posición, planteando que ya en 1987, David Zelzer (del Media Lab) utilizaba dicha expresión [2]

En la actualidad existen varias definiciones de RV, que defieren unas de otras en función del investigador y situaciones; a continuación se muestra una recapitulación de diferentes definiciones, donde cada una entrega una idea de lo que se debería entender por RV [2]:

Realidad Virtual es la experiencia de telepresencia, donde telepresencia es la sensación de presencia utilizando un medio de comunicación.

Realidad Virtual es una manera mediante la cual los humanos visualizan, manipulan e interactúan con computadoras y datos extremadamente complejos.

Realidad Virtual es un paso más allá de lo que sería la simulación por computadores, tratándose más bien de una simulación interactiva, dinámica y en tiempo real de un sistema.

Realidad Virtual consiste en simulaciones tridimensionales interactivas que reproducen ambientes y situaciones reales.

Realidad Virtual es un entorno de tres dimensiones sintetizado por computadora, en el que participantes acoplados de forma adecuada pueden manipular elementos físicos simulados en el entorno y, de alguna manera, relacionarse con las representaciones de otras personas pasadas, presentes o ficticias, o con criaturas inventadas.

Realidad Virtual es un ambiente altamente interactivo donde el usuario participa a través del uso de un computador en un mundo virtualmente real. Es una simulación tridimensional por computadora durante la cual el usuario resulta inmerso tan completamente que esta realidad, de origen artificial, aparenta ser real.

Realidad Virtual es simulación por computadora, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y variables complejas, durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a mundos que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial. Una nueva y sorprendente forma de navegar información.

Realidad Virtual es una simulación tridimensional interactiva por computador en la que el usuario se siente introducido en un ambiente artificial, y que lo percibe como real basado en estímulos a los órganos sensoriales.

Realidad Virtual es el medio que proporciona una visualización participativa en tres dimensiones y la simulación de mundos virtuales, siendo dichos mundos el elemento fundamental de un sistema de

realidad virtual. La realidad virtual es un entorno generado por computador en el que los participantes pueden entrar físicamente e interactuar con él, desplazándose por su interior o modificándolo de cualquier manera.

Realidad Virtual o mundo virtual es una base de datos gráficos interactivos, explorables y visualizables en tiempo real en forma de imágenes tridimensionales de síntesis capaces de provocar una sensación de inmersión en la imagen. En sus formas más complejas, el entorno virtual es un verdadero 'espacio de síntesis', en el que uno tiene la sensación de moverse 'físicamente'. Esta sensación de 'movimiento físico' puede conseguirse de diferentes formas, la más frecuente consiste en la combinación de dos estímulos sensoriales, uno basado en una visión estereoscópica total y el otro en una sensación de correlación muscular, llamada 'propioceptiva', entre los movimientos reales del cuerpo y las modificaciones aparentes del espacio artificial en que está inmerso.

El computador y el software especial que se utiliza para crear la ilusión de RV constituye lo que se ha denominado "máquina de realidad" ("reality engine").

Una "máquina de realidad" es el corazón de cualquier sistema de realidad virtual porque procesa y genera Mundos Virtuales, incorporando a ese proceso uno o más computadores. Una "máquina de realidad" obedece a instrucciones de Software destinadas al ensamblaje, procesamiento y despliegue de los datos requeridos para la creación de un mundo virtual, debiendo ser lo suficientemente poderosa para cumplir tal tarea en "tiempo real" con el objeto de evitar demoras entre los movimientos del participante y las reacciones de la máquina a dichos movimientos.

Los componentes básicos de un sistema de realidad virtual son: el modelo de simulación, la representación del ambiente virtual, la entrada/salida y el usuario.

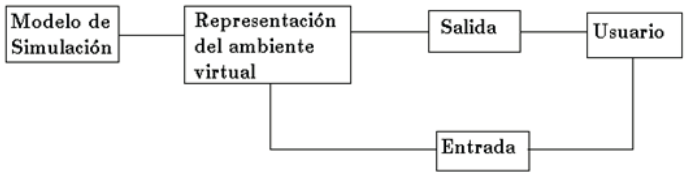

Figura 1. Componentes de RV

Los orígenes de RV se remontan a los años 60's cuando emergen los primeros simuladores de vuelo construidos para la fuerza aérea estadounidense, donde los estudiantes de piloto aprendían a manejar aviones entrenando en cabinas aéreas montadas en plataformas movibles las cuales subían, bajaban o movían hacia los lados dependiendo de las acciones que el piloto realizara sobre los controles [2,3,4].

En 1962, aparece la máquina de arcade creada por Morton Heiling; esta máquina simulaba las experiencias sensoriales de un paseo en motocicleta, al combinar imágenes, sonido, viento y aromas. En esta experiencia el usuario subía en el asiento de una motocicleta, tomaba los manubrios y usaba un visor parecido a unos binoculares donde podía pasear por dunas en California o las calles de Brookly, y donde unos pequeños ‹grills› cerca de la nariz del usuario emitían aromas auténticos.

En 1965, Ivan Sutherland, publicó un artículo denominado «The Ultimate Display», donde sentó las bases del concepto de realidad virtual, Sutherland estipulaba «La pantalla es una ventana a través de la cual uno ve un mundo virtual. El desafío es hacer que ese mundo se vea real, actúe real, suene real, se sienta real».

En 1966, el mismo Sutherland creó un casco visor de realidad virtual al montar tubos de rayos catódicos en un armazón de alambre.

Este instrumento fue llamado «Espada de Damocles» debido a que el aparato requería de un sistema de apoyo que pendía del techo.

En 1969, Myron Krueger crea ambientes interactivos que permiten la participación del cuerpo entero, en eventos apoyados por computadores. Por aquel entonces, diversos artistas ofrecían espectáculos que incluían imágenes, sonidos, vibraciones y hasta olores, que junto con cámaras que captaban sus movimientos sumergían al espectador en un mundo irreal, reproducido en una pantalla gigante.

En 1971, Redifon Ldt. en el Reino Unido, comienza a fabricar simuladores de vuelo con pantallas gráficas.

En 1972, General Electric, bajo mandato de la Armada Norteamericana, desarrolla el primer simulador computarizado de vuelo.

A fines de los 70, en el Media Lab del Instituto Tecnológico de Massachusetts (MIT), se obtiene el "Mapa Filmado" de Aspen, EE.UU, una simulación de video a través de la ciudad, donde el usuario puede recorrer sus calles y edificios.

Tom de Fanti fue el inventor del guante de datos en 1976, aunque su diseño fue mejorado posteriormente por Tom Zimmerman, dando origen al DataGlove, diseñado originalmente para poder tocar una guitarra virtual o imaginaria

La era de la realidad virtual inicia en los 80›s, cuando la NASA inició con el sistema de imágenes generadas por computadora.
En 1985 fue construido el primer sistema práctico de visores estereoscópicos para la NASA.

En 1989 el Departamento de defensa de los E.U crea a Simnet (Simulador de Red), una red experimental de estaciones de trabajo basadas en microprocesadores que habilitaban al personal a prácticas de operaciones de combate interactivamente en sistema de entrenamiento de tiempo real. De hecho este sistema se usó para entrenar al ejército en la Guerra del Golfo Pérsico en 1991.

A inicios de los 90, los sistemas de Realidad Virtual salen de los ambientes de laboratorios en busca de aplicaciones comerciales, apareciendo cabinas de entrenamiento para pilotos de guerra, simuladores de vuelo, recorridos virtuales, entre otras aplicaciones.

En la actualidad, estamos aún en presencia del crecimiento y consolidación de las técnicas y recursos de la Realidad Virtual, el cual ha sido posible gracias al esfuerzo e interés combinado de científicos, militares y visionarios, y porqué no decirlo, al dinero de las empresas que ven en ella una nueva y prometedora tecnología [2, 3].

En la presente investigación se plantea desarrollar en un ambiente virtual, constituido por modelos 3D, los cuales son manipulados mediante un guante, que está ubicado en la mano del usuario.

Materiales y Métodos

PROCEDIMIENTO

En la fase inicial se realizó un análisis del hardware e interfases existentes, mecanismos y metodologías de los protocolos disponibles, al mismo tiempo se realizó un estudio escrupuloso del entorno Java y sus bondades subyacentes a nuestro proyecto, para posteriormente proceder con el desarrollo del hardware y software.

ACELERÓMETRO ADXL322

El ADXL322 es un acelerómetro pequeño y delgado, de baja potencia, de doble eje X, Y, cuya salida es una tensión equivalente al ángulo de ubicación del acelerómetro, cuando éste está sometido a 0 gravedades (g), su salida será igual a la mitad de VCC, incrementando o disminuyendo este valor de acuerdo a la dirección de la inclinación a la que se lo someta, posee una resolución de ± 2 g (típico).

También se puede medir la aceleración dinámica (vibración) y la aceleración estática (por gravedad). Se puede seleccionar el ancho

de banda del acelerómetro utilizando condensadores CX y CY en los pines Xout y Yout. Puede ser alimentado con una tensión VCC de entre 2.4 a 6V

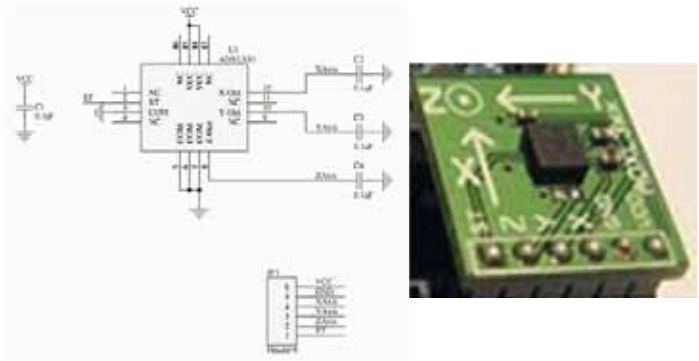

Figura 2. Acelerómetro ADXL 322

MICROCONTROLADORES

Los microcontroladores son circuitos integrados programables, capaces de ejecutar las instrucciones que existen en su memoria. Principalmente constan de un microprocesador, líneas de entrada y salida, memorias RAM y ROM. Para su funcionamiento requiere de alimentación, de un oscilador y de un programa de instrucciones.

La función principal del micro controlador es interpretar combinaciones de bits y generar señales digitales internas y/o externas, para ejecutar de manera continua una secuencia de instrucciones que permita controlar un sistema o subsistema electrónico. Estos dispositivos vienen con un juego de instrucciones reducido, además de su pequeño encapsulado con pocos pines y poco consumo, lo cual los hacen muy utilizables.

La mayoría de los micro-controladores constan con las siguientes características:

- Procesador o CPU: es quien procesa todos los datos que pasan por el bus.
- Memoria ROM: es la memoria no volátil, que es donde se guardan los programas
- Memoria RAM: o memoria volátil, que es donde se guardan los datos.
- Oscilador: que sincroniza todo el funcionamiento del sistema.
- Puertos de entrada y salida: Es por donde se comunica el microcontrolador con los periféricos externos.
- Convertidores análogo digital(A/D), digital análogo (D/A): como su nombre los dice convierten señales de análogas a digital y viceversa.
- Temporizadores: sirven para controlar periodos de tiempo.
- Comparadores: como su nombre lo indica comparan señales analógicas.
- Moduladores de ancho de pulso: esta función modula en PWM.
- Puerto USART: comunicación serie de transmisión y recepción.
- Controladores de interrupciones
- Watch dog: contador que resetea el microcontrolador cada vez que se desborda el stack.
- Protección ante fallo de alimentación: resetea el microcontrolador cada vez que la alimentación baja de un cierto límite.

MICROCONTROLADOR 16F870

Los microcontroladores de la familia 16f87x están diseñados con la arquitectura Harvard. La arquitectura Harvard dispone de dos memorias independientes; una que contiene instrucciones y otra donde se almacenan los datos.

El integrado 16f870 posee tres puertos de E/S; el puerto A con 6 pines, y el puerto B y C con 8. En total tiene 28 pines, 22 de estas configurables como entrada o salidas.

Las características generales del microcontrolador son:

- Voltaje de alimentación de 5V
- Frecuencia de operación máxima de 20Mhz
- Memoria flash de 4Kbytes
- Memoria de datos de 192Bytes
- Memoria de datos EEPROM de 128Bytes

- 2 temporizadores internos de 8 bites
- 1 temporizador interno de 16 bites
- 1 módulo de captura, comparación y PWM
- 1 módulo de comunicaciones USART
- 5 canales de conversión análoga digital
- 13 interrupciones

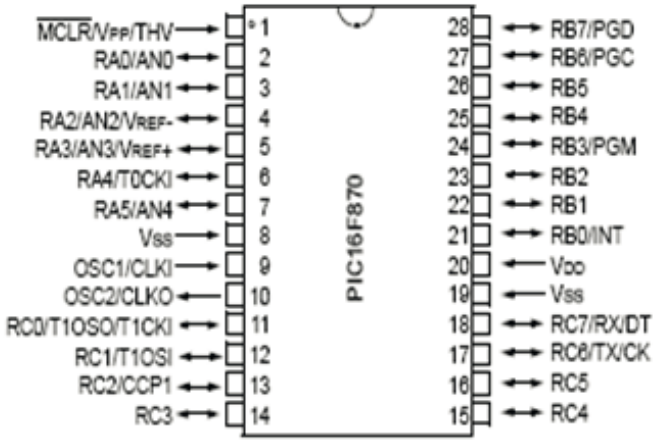

Figura 3. Microcontrolador 16F870, distribución de pines

CONVERSIÓN ANÁLOGA DIGITAL

La conversión análoga digital (A/D) se realiza en el microcontrolador 16f870, y éste toma la señal que viene del sensor. En este caso un voltaje continuo y lo trasforma a un número digital.

La conversión A/D consiste en la trascripción de una señal analógica en una señal digital, para su tratamiento como tal. Las señales análogas son señales continuas que tienen una frecuencia y amplitud; en cambio, una señal digital es discreta, que tiene tiempo y amplitud, esta señal toma un valor fijo en un tiempo determinado. Estos valores fijos corresponden al sistema binario (0,1).

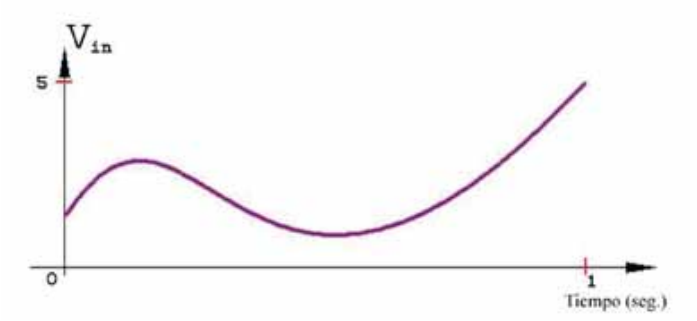

Figura 4. Señal analógica

En el proceso de conversión consiste en 4 etapas:

Muestreo: consiste en tomar muestras periódicas de la amplitud de la señal de entrada. La velocidad con que se realiza este proceso, el número de muestras por segundo, se le denomina frecuencia de muestreo.

Retención: las muestras han de ser retenidas durante un instante de tiempo para evaluar su nivel.

Éste más que un proceso matemático es un recurso técnico realizado por algún circuito de retención que cumple esta labor.

Cuantificación: en este proceso es donde se mide el nivel de voltaje de cada una de las muestras, consiste en asignar un margen de valor de la señal de entrada a un único valor de salida.

Codificación: consiste en traducir los valores obtenidos durante la cuantificación a código binario,

Hay que tener presente que el código binario es el comúnmente utilizado pero igual existe otro tipo de codificaciones.

Cabe destacar que en los dos primeros la señal aún es analógica, solo en los dos últimos se puede catalogar como señal digital.

REGISTROS DEL MÓDULO A/D DEL 16F870

El modulo A/D del microcontrolador tiene 4 registros, pero se utilizará dos, ADCON0 y ADCON1

REGISTER 10-1: ADCON0 REGISTER (ADDRESS: 1Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0**: A/D bits de selección de conversión de reloj

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = FRC

bit 5-3 **CHS2:CHS0**: bits de selección del canal analógico

000 = Channel 0, (RA0/AN0)

010 = Channel 2, (RA2/AN2)

011 = Channel 3, (RA3/AN3)

100 = Channel 4, (RA5/AN4)

101 = Channel 5, (RE0/AN5)**(1)**

110 = Channel 6, (RE1/AN6)**(1)**

111 = Channel 7, (RE2/AN7)**(1)**

bit 2 **GO/DONE**: A/D

If ADON = 1:

1 = A/D conversión en progreso

0 = A/D conversión no en progreso

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit

1 = A/D modulo de convertidor operando

0 = A/D converter module is shut-off and consumes no operating current

ADCON1

Este registro configura los pines del puerto.

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Bit 7, ADFM: Selecciona el formato del resultado de la conversión A/D

1 = >Pone en el registro **ARDESH** los seis bits de mayor peso a "0"

0 = >Pone los 6 bits de menor peso del registro **ADRESL** a "0"

Bits 6-4: No implementados

Bit 3-0: **PCFG3:PCFG0**: bits de configuración de los canales de entrada del convertidor A/D. Se utilizan para configurar las patillas como E/S digital o como entrada analógica de acuerdo con la siguiente tabla:

PCFG3: PCFG0	AN7[1] RE2	AN6[1] RE1	AN5[1] RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ REFS[2]
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

MAX232

El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa, necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. Mientras que otros integrados que manejan las líneas RS232 requieren dos voltajes, +12V y -12V.

El MAX232 brinda la comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento con base en señales de nivel TTL/CMOS.

El circuito integrado posee dos conversores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL.

Estos conversores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS.

-TX es la señal de transmisión de datos

-RX es la de recepción

-RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.

DESARROLLO DEL GUANTE DE NAVEGACIÓN

El guante está compuesto por un acelerómetro que entrega el ángulo de giro de un punto específico, para manipular los datos entregados por el acelerómetro se utilizan las entradas analógicas del PIC16f870, mismo que envía los datos hacia un computador de manera serial, a través del circuito integrado max232 que permite acoplar los voltajes, entre el puerto serial y el PIC.

El diagrama del circuito se presenta en la figura 5, donde se utiliza el PIC18f2550 que es similar al PIC16f870 en la distribución de pines, pero el primero posee 10 entradas analógicas lo que lo hace ideal para manejar tres acelerómetros, los que están simbolizados por SL1, SL3 y SL4,

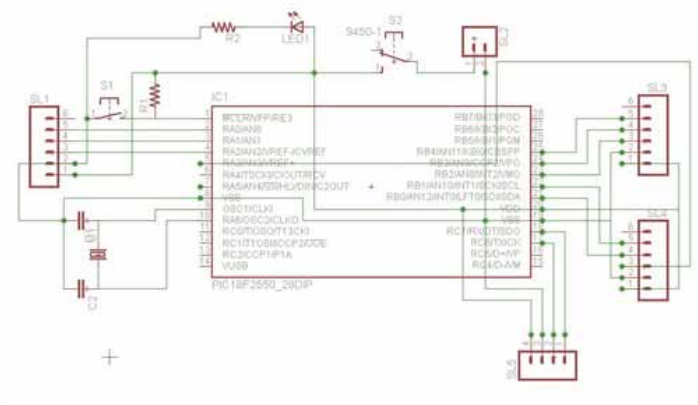

Figura 5. Diseño del circuito de adquisición de datos

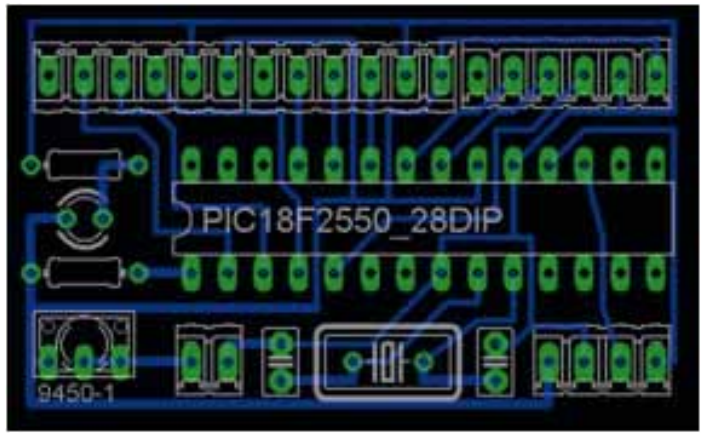

Figura 6. Diseño del PCB

El circuito de la comunicación del guante de navegación, con el computador se presenta en la figura 7.

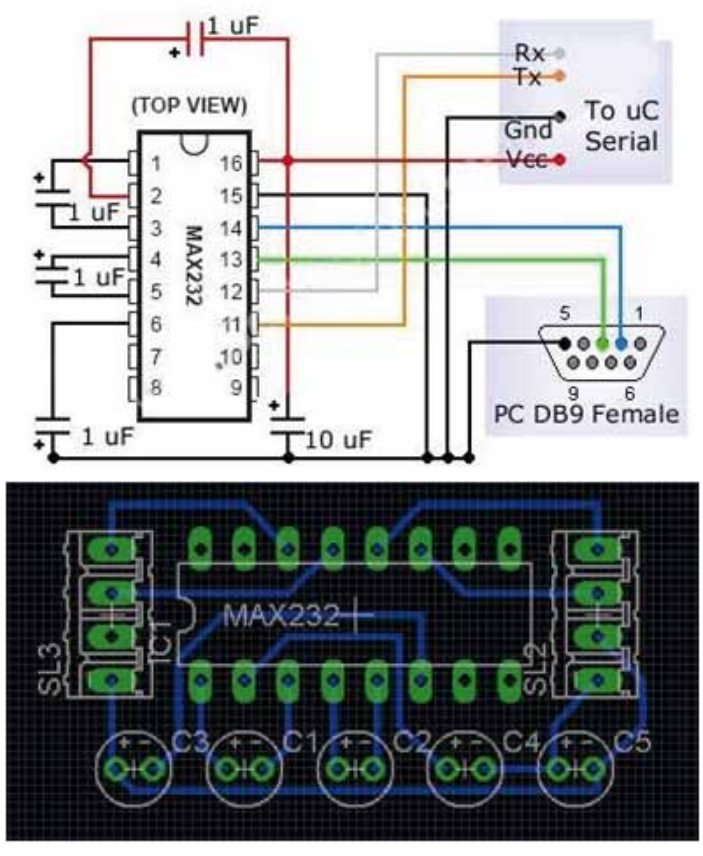

Figura 7. Diseño, esquema Max232.

El circuito completo del sistema de navegación se presenta en la figura 8.

Figura 8. Circuito de navegación

DESARROLLO DEL AMBIENTE VIRTUAL

JAVA

Java es una plataforma (lenguaje) para desarrollo de software desarrollada por Sun Microsystems, que permiten a los programas desarrollados, puedan ser ejecutados sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales, gracias a la implementación de máquina virtual (JRE)[6].

JAVA 3D

Java 3D es un proyecto que permite crear entornos tridimensionales en Java. El API Java 3D es una interface para escribir programas que muestran e interactúan con gráficos tridimensionales que corre sobre OpenGL o Direct3D. Java 3D es una extensión estándard del JDK 2 de Java y proporciona una colección de constructores de alto-nivel para crear y manipular geometrías 3D y estructuras.

Java 3D proporciona las funciones para creación de imágenes, visualizaciones, animaciones y programas de aplicaciones gráficas 3D interactivas [7,8].

Un programa Java 3D crea ejemplares de objetos Java 3D y los sitúa en una estructura de datos de escenario gráfico. Este escenario gráfico es una composición de objetos 3D en una estructura de árbol que especifica completamente el contenido de un universo virtual, y cómo va a ser renderizado.

Los programas Java 3D pueden escribirse para ser ejecutados como aplicaciones solitarias o como applets en navegadores web extendidos para soportar Java 3D [9].

Figura 9. Aplicaciones java 3D

3d Max

Autodesk 3ds Max, anteriormente 3D Studio MAX, es un modelador de animación, renderizado paquete desarrollado por Autodesk Media and Entertainment.

Tiene capacidades de modelado, una flexible arquitectura de plugins y es capaz de ser utilizado en la plataforma Microsoft Windows.

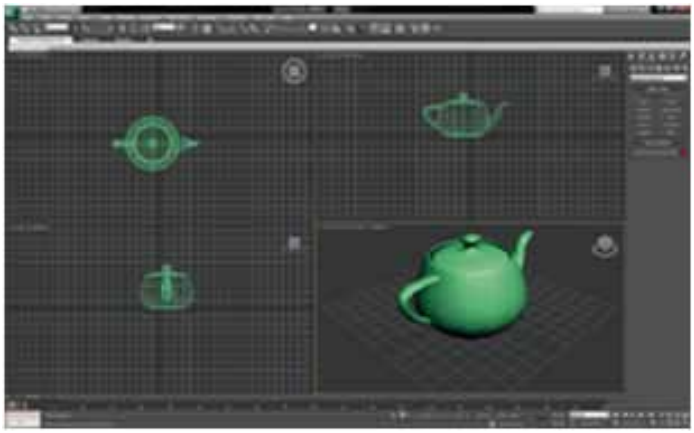

Figura 10 Aplicaciones en 3ds Max

Además de las herramientas de modelado y animación, las últimas versiones de 3ds Max cuentan con shaders (por ejemplo, oclusión de ambiente y del subsuelo de dispersión), simulación dinámica, sistemas de partículas, radiosidad, la creación de mapas normales y de representación, la iluminación global, una interfaz de usuario personalizable, y su propio lenguaje de scripts.

Para el desarrollo del ambiente tridimensional se utilizó Java como lenguaje de programación con la plataforma de desarrollo Eclipse.

Para la modelación de las piezas se utiliza 3DStudio Max™ 5 y un «loader» para ser utilizadas en Java3D™, que provee el NCSA (Centro Nacional de Aplicaciones de Supercomputación). Debido a la complejidad de coordinar el movimiento del pistón con los brazos, solo se diseñaron las siguientes piezas.

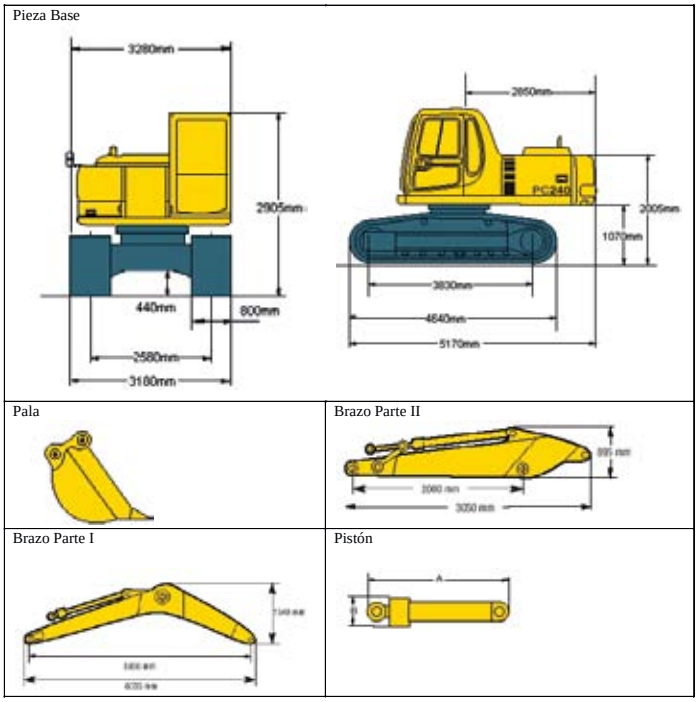

Figura 11. Estructura de la Máquina Excavadora

Movimiento

Los movimientos básicos de una excavadora son:

- Giro que hace la parte de la cabina en torno a la oruga.
- Giro vertical de la parte II del brazo en torno a la parte I de éste.
- Giro vertical que hace la pala en torno a la parte II del brazo.
- Giro vertical de la parte I con respecto a la base.

Modelamiento

El modelamiento se realizó utilizando 3D Studio Max™ 5, que permite modelar cada una de las piezas antes descritas, para luego generar archivos con los puntos correspondientes. Desde Java3D™ se utiliza un Loader, programa que permite leer los archivos generados por 3D Studio Max™ con formato .3DS

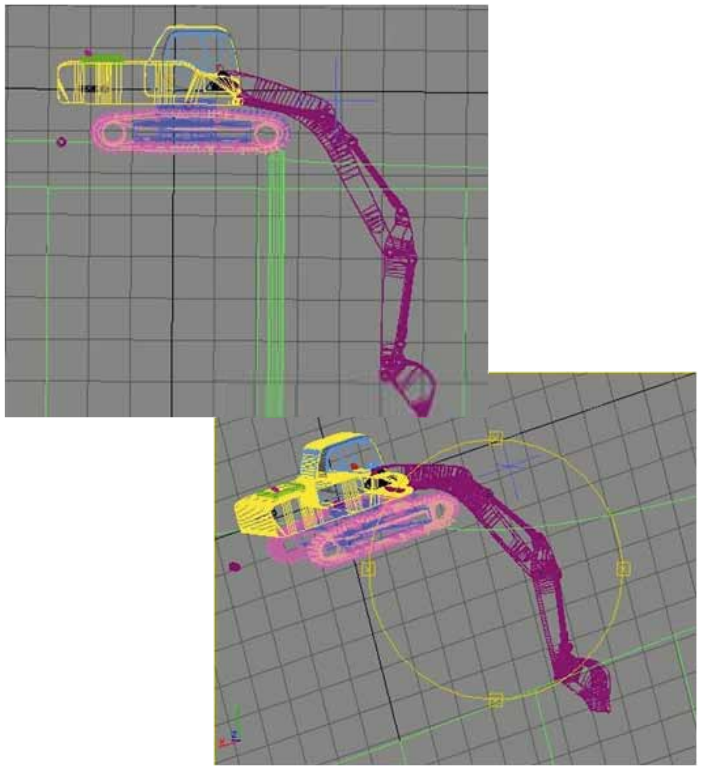

Figura 12. Máquina Excavadora en alambre

Para comprobar el funcionamiento del hardware se utilizó un ambiente desarrollado en la Universidad de Chile (Departamento de Ciencias de la Computación), donde el programa recibe el ángulo de giro de los elementos a través de una comunicación serial entre la PC y el PIC

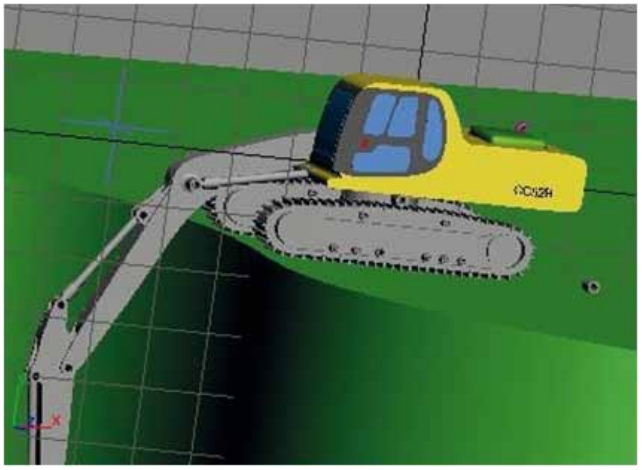

Figura 13. Máquina Excavadora en un ambiente 3D

Figura 14. Simulación de los movimientos de la máquina Excavadora

Conexión serial con Java

Java provee una librería para el manejo de comunicaciones a través de puertos como el serial y paralelo (la librería *comm*); sin embargo, el uso e implementación de esta librería está enfocado para sistemas Linux y Solaris, y su configuración en Windows es un tanto complicada, dicha librería provee de métodos para enviar y recibir datos a través de los puertos serial y paralelo y para usarla se debe crear un proyecto, en la clase *main* agregando las siguientes variables a nivel de clase:

static SerialPort puerto = new SerialPort(); *static List<String> listaPuertos;* *static Com com1;*	**SerialPort**: es un objeto del tipo de puerto **listaPuertos**: es la lista que contendrá los puertos libres que existen **com1**: es la interface que manejara el control del dispositivo.
listaPuertos = puerto.getFreeSerialPort(); *String recibido = "";* *String caracter = "";*	obtener la lista de puertos disponibles la variable tipo String contendrá todos los datos leídos desde el puerto serie y la otra contendrá el último carácter leído
for (String string : listaPuertos) { *System.out.println(string);* *}*	verificar los puertos que están disponibles creamos un ciclo que los imprima
Parameters settings = new Parameters(); *settings.setPort("COM1");* *settings.setBaudRate("9600");*	se establece las propiedades necesarias con el puerto, en este caso, para manejar 9600 bits por segundo a través del COM1…
com1 = new Com(settings);	con base a los parámetros establecidos se crea una instancia del objeto tipo interfaz de comunicación
while(!caracter.equals("\n")){ *caracter =com1.receiveSingleString();* *recibido += caracter;* *}*	con la instancia creada, se procede a leer desde el puerto serie, en este caso, mientras el carácter leído no sea un retorno de carro
System.out.println(recibido);	finalmente se imprime los datos recibidos

Cargadores (loaders)

Para hacer posible la lectura de archivos 3dmax desde java, necesitamos crear una clase Loader, que lee ficheros de escenas 3D (no ficheros Java 3D) crea representaciones Java 3D de sus contenidos que pueden ser añadidos selectivamente a un mundo Java 3D y argumentados por otro código Java 3D. El paquete com. sun.j3d.loaders proporciona el contenido principal para convertir los ficheros creados en otras aplicaciones en Java 3D. Las clases cargadoras implementan la interface Loader definido en el paquete com.sun.j3d.loaders [10].

Debido a la gran variedad de formatos de ficheros para propósitos de representación de escenas 3D (por ejemplo, .obj, .vrml, etc.) y siempre habrá más formatos de ficheros, el código real para cargar un fichero no forma parte de Java 3D o del paquete loaders; sólo se incluye el interface para el mecanismo de carga.

Con la definición de la interface; se puede desarrollar clases cargadoras de ficheros con la misma interface que las otras clases cargadoras.

Cargadores Disponibles Públicamente

En Java 3D existe una variedad de clases cargadoras. La siguiente tabla ilustra los formatos de ficheros cuyos cargadores están disponibles públicamente.

Formato de Fichero	Descripción
3DS	3D-Studio
COB	Caligari trueSpace
DEM	Digital Elevation Map
DXF	AutoCAD Drawing Interchange File
IOB	Imagine
LWS	Lightwave Scene Format
NFF	WorldToolKit NFF format
OBJ	Wavefront
PDB	Protein Data Bank
PLAY	PLAY
SLD	Solid Works (prt and asm files)
VRT	Superscape VRT
VTK	Visual Toolkit
WRL	Virtual Reality Modeling Language

Interfaces y Clases base del paquete Loader

Esta gran variedad de cargadores existe para hacer más sencilla la escritura de cargadores para los diseñadores Java 3D. Las clases Loader son implementaciones de la interface Loader que baja el nivel de dificultad para escribir un cargador.

Sumario de Interfaces de com.sun.j3d.loaders	
Loader	El interface Loader se usa para especificar la localización y los elementos de un formato de fichero a cargar
Scene	El interface Scene es un conjunto de métodos usado para extraer información de escenario gráfico Java 3D de una utilidad cargador de ficheros. Además de estas interfaces, el paquete com.sun.j3d.loaders proporciona implementaciones básicas de los interfaces
Sumario de Clases de com.sun.j3d.loaders	
LoaderBase	Esta clase implementa el interface Loader y añade constructores. Esta clase es extendida por los autores para especificar clases cargadoras
SceneBase	Esta clase implementa el interface Scene y añade métodos usados por los cargadores. Esta clase también es usada por los programas que usan clases cargadoras
Sumario de Métodos del Interface Loader	
Scene load(java.io.Reader reader)	Este método carga el Reader y devuelve el objeto Scene que contiene la escena
Scene load(java.lang.String fileName)	Este método carga el fichero nombrado y devuelve el objeto Scene que contiene la escena
Scene load(java.net.URL url)	Este método carga el fichero nombrado y devuelve el objeto Scene que contiene la escena
void setBasePath(java.lang.String pathName)	Este método selecciona el nombre del path base para los ficheros de datos asociados con el fichero pasado en el método load(String)
void setBaseUrl(java.net.URL url)	Este método selecciona el nombre de la URL base para los ficheros de datos asociados con el fichero pasado en el método load(String)
void setFlags(int flags)	Este método selecciona las banderas de carga para el fichero

DISCUSIÓN

El sistema cumple con los objetivos trazados al inicio de la investigación que fueron desarrollar el prototipo de un sistema de manipulación. El software utilizado en lo posible se trató que sea de código abierto y de fácil acceso para minimizar tiempos y costos en el desarrollo.

La arquitectura del Hardware es de fácil implementación y con base tecnológica disponible en el mercado nacional, lo cual de alguna manera interviene en el rendimiento global, específicamente en la velocidad de reacción de los modelos ante la manipulación, quedando abierta la posibilidad de mejorar, ya sea utilizando medios de comunicación de alta velocidad o mejorando los módulos software y hardware.

Conclusiones

La utilización de transmisión serial genera un retardo en la visualización de los movimientos, desde el momento en que se genera un cambio de posición en el acelerómetro, debido a los subprocesos utilizados para determinar dichas variaciones.

Los pics que poseen un puerto USB podrían mejorar la velocidad de transmisión; sin embargo, pero al momento de programar se crea un puerto serial virtual, lo que transforma a esta salida USB en una transmisión serial, por tanto la latencia de comunicación se mantiene.

La resolución entregada por el acelerómetro no es muy alta, entregando una variación de posición por cada 0.9 grados por lo que en movimientos pequeños menores a este valor no se detecta ningún cambio.

Resulta más conveniente utilizar 3D Studio Max, para el desarrollo de objetos tridimensionales, lo que permite ahorro de recursos económicos y tecnológicos, que pueden ser reasignados a otros procesos.

Para proyectos de este tipo se recomienda utilizar un computador con altas prestaciones, tanto de cálculo como de almacenamiento, lo cual agilizará el procesamiento del modelo matemático que se genera.

Se podría hacer uso de elementos de multiplexado para adquirir los datos del acelerómetro; sin embargo, podría incrementar el tiempo de respuesta del hardware.

Bibliografía

Contreras F. R. (1997), La Creación Inteligente en el Ciberespacio, Congreso de la AAS, Universidad de Sevilla

ISEA (2008), INTERNET 3D, Análisis prospectivo de las potenciales aplicaciones asociadas a los Mundos Virtuales

Cardozo H. J. (2004), Realidad Virtual, TAI 2, UCA

BOTELLA, Cristina [et al.] (2007). «La utilización de las nuevas tecnologías de la información y la comunicación en psicología clínica». En: E. HERNÁNDEZ y B. GÓMEZ-ZÚÑIGA (coords.as). «Intervención en salud en la Red». UOC Papers [artículo en línea].N.° 4. UOC. [Fecha de consulta: 12/03/11].<http://www.uoc.edu/uocpapers/4/dt/esp/botella.pdf> ISSN 1885-1541

Buss, Samuel R .(2003). *3-D Computer Graphics*. Cambridge University press, New York

Eberly, David (2003). 3D *Game Engine Architecture*. Morgan Kufmann publishers, Sam Francisco

Flawonn, Frank (2008). *Introduction to computer Graphics Using Java 2D and 3D*. Springer, London

Selman, Daniel (2006). *Java 3D programming*. Manning Publications

Sowizral Henry, Rushforth Kevin, Deering Michael, (1998). *The Java 3D API Specification*. JavaSoft, PaloAlto

Zhang Hong, Y. Daniel Liang – Armstrong (2006). *Computer Graphics Using Java™ 2D and 3D*. Prentice Hall