

Uso de Software Libre (Scilab) en la Captura y Procesamientos de datos en Ingeniería Mecatrónica

Hernández García Edgar, Allende Chávez Edgar, Martínez Rodríguez Cándido,
García Limón Brenda Yazmin.

Instituto Tecnológico Superior de Zacapoaxtla
División de Ingeniería Mecatrónica
Carretera Acuaco-Zacapoaxtla, Km 8, Col. Totoltepec, Zacapoaxtla, Pue.
Tel: (233) 31 75000, Correo electrónico: virmin_462hotmail.com

Resumen

En el presente texto se analizan las ventajas del uso del software libre en las ingenierías así mismos se muestra una aplicación práctica y común dentro de la Mecatrónica, como lo es la captura y procesamiento de datos, en este caso se hace uso de Scilab 5.4.1, el cual es un software libre, así mismo se compara su eficiencia contra Matlab el cual es un software con licencia, Scilab 5.4.1 se trabajó sobre el sistema operativo Fedora 18, con la finalidad de realizar un análisis de vibraciones mecánicas a distancia.

Palabras claves: software libre, análisis de vibraciones, Mecatrónica, sistema operativo, captura, procesamiento.

Introducción

Actualmente todo a lo que se refiere a aplicaciones de computadora en la enseñanza en las ingenierías se encuentra copado, casi en su totalidad por software propietario (con uso de licencia comercial), los cuales resultan de difícil acceso para los estudiantes. Por otro lado, los planes de estudio que se restringen a este tipo de software obligan al estudiante de ingeniería a tener una sola perspectiva de solución en sus proyectos escolares. Un ejemplo palpable se presenta en la Ingeniería Mecatrónica, cuando se requiere hacer un análisis de vibraciones mecánicas, el cual normalmente se estudia con la conocida aplicación comercial de procesamiento numérico: MathWorks Matlab [1].

Una propuesta diferente pero equivalente, es la que representa el uso del software libre, en el cual no se requiere de una gran inversión económica, además de que en la actualidad existen varios proyectos de aplicaciones en diversas ramas de la ingeniería. Este es el caso del presente trabajo, en el que se ha optado por el uso de Scilab (Figura 1).

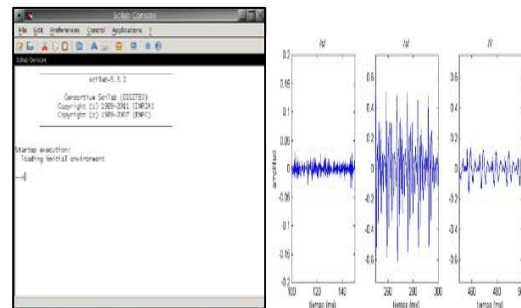


Figura 1: Entorno Scilab 5.4.1 , Espectros de frecuencia

Metodología

1. Elección de sistema operativo

Para la realización de este proyecto se usó como plataforma el sistema operativo GNU/Linux en su distribución Fedora 18(en la figura 2 se muestra el entorno de fedora 18) .



Figura 2: Escritorio de Fedora 18

Fedora es un proyecto apoyado por la empresa Red Hat, pionera y desarrolladora de su propia distribución (de paga). Esta distribución tiene como meta el de desarrollo de un sistema operativo completamente basado en software libre. Además, los propios ingenieros de Red Hat participan del continuo desarrollo de ésta distribución [2]. Otra ventaja sobre otras distribuciones es la posibilidad de tener un entorno dedicado al desarrollo técnico-educativo (Spins ElectronicLab, Robotics, Scientific-KDE).

2. Instalación de software y paqueterías sobre Fedora 18.

Instalar Scilab sobre Fedora 18 es sencillo, basta con escribir en la línea de comandos, de la terminal del sistema el comando “sudo yum –y install scilab”, como se ilustra en la figura 3, con esto, y escribiendo nuestra respectiva contraseña de superusuario, obtendremos la versión más reciente, como se muestra en las ilustraciones.

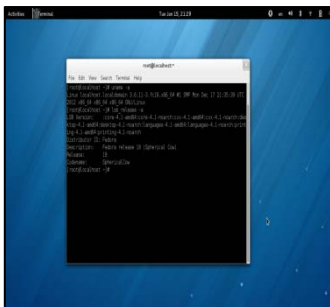


Figura 1. Instalación de Scilab

3. Programación en Scilab

Scilab no posee herramientas propias para el manejo de puertos seriales, para poder realizar este tipo de conexión es necesario recurrir a los llamados Toolboxes (cajas de herramientas), las cuales no son más que aditamentos que se han pre-programado para darnos funcionalidades extras de una manera más sencilla. Esta

característica nos recuerda a aditamentos equivalentes en Matlab. Una ventaja adicional es que, dado el carácter libre de nuestro entorno de trabajo, tendremos acceso a variados Toolboxes en Internet, los cuales están en continua actualización y desarrollo.

Para la gestión de estas cajas de herramientas, en la figura 4 se muestra como en Scilab se incorpora un gestor llamado “atoms”, al cual podemos acceder desde la pantalla principal. Esta utilidad nos permite agregar o quitar componentes de manera sencilla.

Así entonces, en nuestro caso particular instalamos el Toolbox llamado “Serial Communication Toolbox 0.4.1-2”. [3], como aparece en la figura 5.

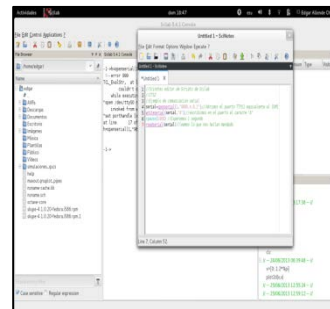


Figura 2: Instalación de Toolboxes

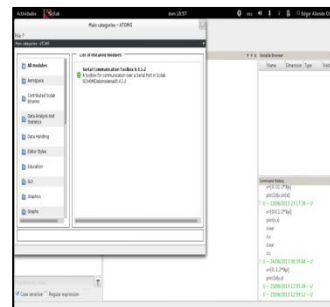


Figura 3: Puerto serial

La lógica de programación utilizada para el proyecto se muestra en el siguiente diagrama de flujo en la figura 6.

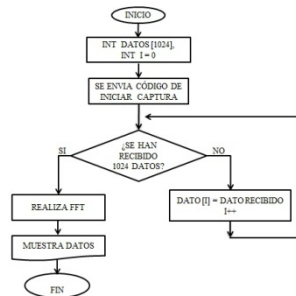


Figura 6: Diagrama de Flujo

4 Adaptación y comunicación con el hardware.

La comunicación se realizó usando un módulo Bluetooth serial en conjunto con una placa Arduino Mega, la cual se ha programado bajo un sistema operativo GNU/Linux en su distribución Ubuntu 10.10, a través de la librería AVR-GCC también de carácter libre, la descripción de la programación en el Arduino es la siguiente:

En primer lugar el Arduino espera un código para comenzar a hacer una lectura analógica de un acelerómetro en el eje deseado con su respectiva conversión de analógico a digital, esta lectura es realizada a razón de 5000 muestras por segundo, después dichos datos son enviados vía Bluetooth, a través del protocolo RS232 cuando termina de enviar 1024 vuelve a su estado de standby.[4]

5. Programación del Arduino sobre el sistema operativo GNU/Linux

Arduino es una plataforma de diseños de prototipos rápido (en la figura 6 se muestra e entorno grafico de Linux), basada en la ideología del hardware libre, que consiste en observar, analizar y mejorar libremente las partes físicas que componen la tarjeta. Lo anterior gracias al acceso a los diagramas esquemáticos de las diferentes tarjetas construidas hasta el día de hoy, dándonos la posibilidad de crearla uno mismo. Ahora bien, la idea de hardware libre es hermana de la idea de software libre, así que para desarrollar en este tipo de plataforma nada mejor que su contraparte lógica/informática, es decir GNU/Linux.

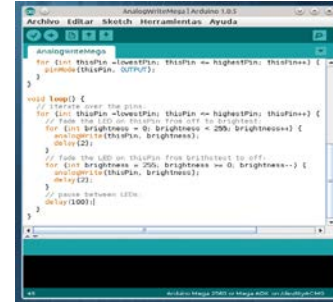


Figura 6:Entorno de Arduino en Linux

El entorno de desarrollo de Arduino, está totalmente escrito en java, lo cual hace que sea multiplataforma, esto se ve reflejado en el hecho de poder utilizar dicho entorno sobre plataformas GNU/Linux, pues la mayoría de ellas cuenta con una implementación de la máquina virtual de java, ya sea la versión libre en la figura 7 se muestra como se ha detectado el Kernel, desarrollada por la comunidad GNU o la versión oficial de Oracle (TM). La detección de la tarjeta Arduino es prácticamente trivial, dado que solo se conecta y el sistema la detecta automáticamente creando un puerto virtual de tipo serial, al que se accede fácilmente concediendo los permisos necesarios para utilizar dicho puerto, lo cual tampoco es demasiado difícil[5].

El entorno Arduino es sencillo, lo cual ayuda cuando se está aprendiendo a utilizarlo, y una vez adquirida la experiencia se puede volver una poderosa herramienta.

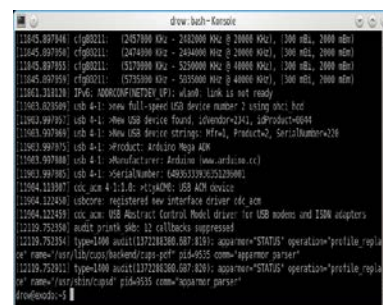


Figura 7:El Arduino detectado desde el Kernel del sistema

6. Pruebas

En Scilab se realiza lo siguiente como se muestra en las figuras 8 ,9 y 10 se inicializa el puerto a utilizar mediante el comando `h=openserial (numero_de_puerto,"bitrate, bit de paridad, numero de bits, bit de parada")`, posteriormente se envía el código al Arduino mediante el comando `datos[i]=readserial (h)` posteriormente con los

datos capturados en un vector podemos realizar la FFT directamente sobre el vector obtenido solo creando los vectores de frecuencia

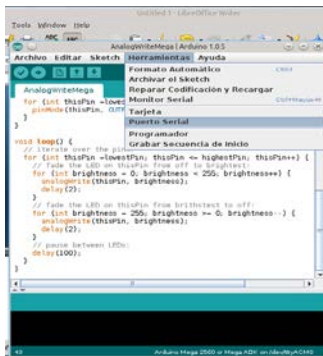


Figura 8:El puerto detectado desde el entorno de Programacion

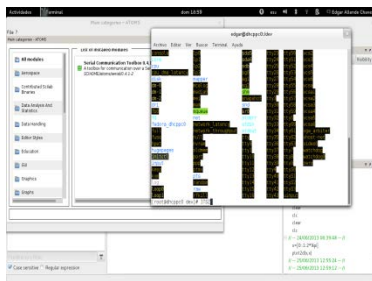


Figura 9: Terminal de Fedora ejecutando programa

A continuación se muestran algunas graficas obtenidas en el script escrito, figuras 10 y 11.

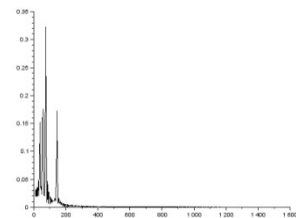


Figura 10:Vibraciones con respecto a la frecuencia

Haciendo uso puramente del software libre específicamente Scilab. Así mismo se logró realizar exitosamente la comunicación serial a través de Scilab con el modulo Bluetooth. De la misma forma se programó la placa Arduino Mega sobre un sistema operativo GNU/Linux específicamente Fedora 18.

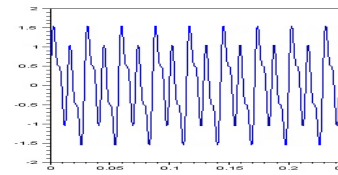


Figura 12: Vibraciones con respecto al tiempo

Al mismo tiempo con fines de comparacion se realizo todo el trabajo antes mencionado con el software Matlab, migrando exitosamente el código hecho en Sclab, obteniendo los mismos resultados, en las figuras 12 ,13y 14.

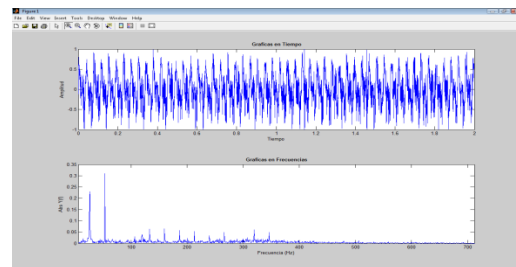


Figura 12: Vibraciones respecto a la frecuencia y al tiempo

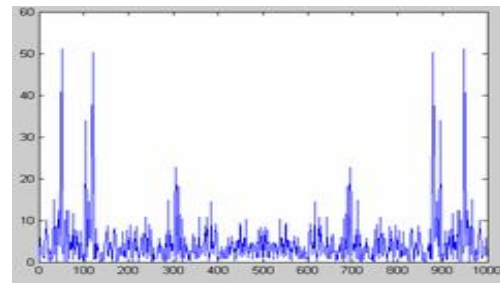


Figura 13: Vibraciones respecto a al tiempo Matlab.

Resultados:

En las figuras arribas mostradas podemos observar el análisis en tiempo real de los datos adquiridos mediante la tarjeta Arduino. Es posible apreciar una respuesta aceptable del sistema, señal que a su vez puede utilizarse para tratar los temas teóricos que normalmente se revisan en clase.

Al igual se ha realizado la comparación entre ambos software y se ha demostrado la eficiencia de Scilab la cual es una alternativa libre viable para estudiantes de ingenierías que no cuentan con la

solvencia económica de obtener una licencia de un software privativo como lo es Matlab.

Conclusiones.

Se ha logrado obtener un dispositivo medidor de vibraciones a distancia, capaz de capturar datos para posteriormente procesarlos con uso del software libre, específicamente Scilab. Así mismo se logró realizar exitosamente la comunicación serial a través de Scilab con el módulo Bluetooth. De la misma forma se programó la placa Arduino Mega sobre un sistema operativo GNU/Linux específicamente Fedora 18. Podemos agregar que Scilab es una opción viable para el análisis de vibraciones mecánicas, lo cual ya se comprobó mediante la realización de este mismo trabajo sobre Matlab, lo cual dio como consecuencia los mismos resultados, tanto en Matlab como en Scilab y particularmente útil en la enseñanza de este tópico en Ingeniería Mecatrónica. Recalcamos además el hecho de que la enseñanza de la Mecatrónica no debe estar restringida al uso del software propietario. Además, este tipo de software cubre perfectamente las necesidades de los estudiantes de ingenierías, además de desarrollar aún más sus competencias profesionales, aprovechando los nichos de oportunidad que el software libre ofrece como reto.

Referencias Bibliográficas

- [1] *Serial communication with scilab in Linux* [en línea]: stack exchange inc. 2012. [Fecha de consulta: 01 de junio de 2013]. Disponible desde internet: <<http://stackoverflow.com/questions/13710539/serial-communication-with-scilab-in-linux>>
- [2] *ATOMS: Serial Communication Toolbox details* [en línea]: Scilab Enterprises S.A.S. 2013. [Fecha de consulta: 10 de junio de 2013]. Disponible desde internet: <<https://atoms.scilab.org/toolboxes/serial/0.4>>
- [3] *Serial Terminal: Controlling RealTerm from Scilab* [en línea]: Simon Bridger, a Sourceforge user. 2013. [Fecha de consulta: 11 de junio de 2013]. Disponible desde <http://realterm.sourceforge.net/realterm_from_scilab.html>

- [4] Delicado, Bernardo A. (s.f.). *Manual de Introducción al Tratamiento de Señales con SCILAB para usuarios de MATLAB*. Recuperado el 15 de junio de 2013, del sitio web de la Dirección Nacional de Innovación Académica de la Universidad Nacional de Colombia: <<http://www.virtual.unal.edu.co/cursos/ingenieria/2001619/lecciones/descargas/senal.pdf>>