ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"PRINCIPIOS DE MANEJO DE SEÑALES CON HERRAMIENTAS OPEN SOURCE"

INFORME DE PROYECTO DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN TELEMÁTICA

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

GRACE MARICELA BERMEO QUEZADA

CARLOS EDUARDO VELÁSQUEZ RUBIO

GUAYAQUIL – ECUADOR

2011

AGRADECIMIENTO

A Dios, a nuestros padres, a nuestras familias, a nuestros profesores, en especial a nuestra directora de proyecto de tesis, Ing. María Antonieta Álvarez y al Ing. Juan Del Pozo por su gran apoyo durante el desarrollo de nuestro proyecto, a la Ingeniera Rebeca Estrada, y a nuestros amigos por su apoyo a lo largo de nuestras carreras.

DEDICATORIA

A Dios, a mi familia, de manera especial a mi Madre y mi Padre, ejes fundamentales en mi formación como persona y por su incondicional amor y apoyo durante mis años de estudio.

Grace

A Dios, a mi abuelo Eduardo, mi inspiración, a mi abuela Anita quien me ha formado y guiado, a mi madre por quien he llegado a cumplir mis metas, y al padre Gaetano Francischini por su apoyo en mi formación.

Carlos

TRIBUNAL DE SUSTENTACIÓN

_____	_____
Ing. Jorge Aragundi	Msc. María Antonieta Álvarez
SUBDECANO DE LA FACULTAD	DIRECTOR DE PROYECTO DE GRADUACIÓN

_____

Msc. Juan Carlos Avilés

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este trabajo, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL)

.

_____ _____

Grace Bermeo Quezada Carlos Velásquez Rubio

RESUMEN

El proyecto está estructurado de secciones que abarcan el análisis de las herramientas necesarias para la creación de aplicaciones de tiempo real, estudio de la viabilidad de instalación en los sistemas operativos Ubuntu, Fedora y Centos, pruebas de eficiencias de desempeño y pruebas de funcionamiento de la herramienta de visualización remota JRtaiLab.

Se ha analizado las ventajas de un sistema de código abierto, frente a sistemas privativos, las definiciones de tiempo real, y los tipos de tiempo real que existen. Para sistemas de alta precisión es necesario manejar un tiempo real estricto. La herramienta encargada de este tipo de procedimientos es RTAI, herramienta que es analizada a lo largo del desarrollo de los temas.

Se realiza la descripción y justificación de herramientas de fines gráficos como MesaLib y EFLTK, de librerías de drivers como Comedi, y el proceso de alteración del kernel de Linux. Se detalla el proceso de instalación y los factores imperativos para la instalación correcta de cada herramienta. En el desarrollo se justifica la necesidad de realizar la resolución de dependencias en cada sistema operativo para poder realizar la instalación de todo el sistema. Se muestra el proceso necesario para realizar la instalación en el sistema operativo Ubuntu y las alteraciones necesarias en el listado de paquetes requeridos para poder realizarlo en los sistemas operativos Fedora y Centos.

También se presentan las estadísticas de los resultados de pruebas de emisión de señales, y de control de una planta de nivel de fluidos, las cuales han sido realizadas sobre cada sistema para evaluar su desempeño. Las pruebas realizadas sobre los sistemas están diseñadas para medir los posibles retrasos presentados por el sistema, la calidad gráfica y facilidad de manejo bajo ciertas tasas de trabajo, y las alteraciones que puede sufrir el control de una planta dependiendo del sistema operativo encargado del control y la justificación de la selección del sistema operativo más apropiado para condiciones de trabajo más precisas.

Luego de finalizado el proceso de instalación y las pruebas de desempeño de los sistemas se detallan las pruebas y el análisis de la eficiencia de la herramienta remota de control de señales JRtaiLab. Se han realizado pruebas de frecuencia máxima de trabajo del sistema conformado por el servidor, controla la planta y el cliente, como también la calidad de las gráficas de las señales, y el análisis de los protocolos utilizados para comunicar el servidor con la aplicación cliente.

ÍNDICE GENERAL

ABREVIATURAS

DAQ: *Data Acquisition*
Adquisición de Datos

FLTK: *Fast Light Toolkit*
Grupo de Herramientas Rápidas y Ligeras

GCC: *GNU Compiler Collection*
Colección de compiladores GNU

GLUT: *OpenGL Utility Toolkit*
Biblioteca de utilidades para programas OpenGL

GPUs: *Graphics Processing Unit*
Unidad de procesamiento de gráficas

HAL: *Hardware Abstraction Layer*
Capa de Abstracción de Hardware

INRIA: National Institute for Research in Computer Science and Control.
Instituto Nacional de Investigación en Ciencias de la Computación y Control.

IPC: *Inter-Process Communication*
Comunicación entre Procesos.

LXRT: *Linux Real Time*
Tiempo Real Linux

NRSE: *Nonreferenced Single-Ended*
Medida simple sin referencia

PCI: *Peripheral Component Interconnect*
Interconexión de Componentes Periféricos

PGIA: *Programmable Gain Instrumentation Amplifier*
Amplificador de Instrumentación de ganacia programable

RAM: *Random-access memory*
Memoria de Acceso Aleatorio.

RPC: *Remote Procedure Call*
Llamada a Procedimiento Remoto

RSE: *Referenced Single-Ended*
Medida simple referenciada

RTAI: *Real Time Interface Application*
Interfaz para Aplicaciones en Tiempo Real

RTAI-XML: *Extensible Markup Language*
Lenguaje de Marcas Extensible

SOAP: *Simple Object Access Protocol*
Protocolo de Acceso a Objeto Simple

STR: *Real Time System*
Sistema de tiempo real

TCP/IP: *Transmission Control Protocol/Internet Protocol*
Protocolo de control de transmisión/Protocolo de Internet

XML-RPC: *eXtensible Markup Language - Remote Procedure Call*
Lenguaje de Marcas Extensible - Llamada a Procedimiento Remoto

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El predominio de soluciones propietario para sistemas de control de maquinarias y de plantas industriales, significan una voluminosa inversión para su implementación y utilización; por lo tanto, este proyecto constituye un inicio del proceso de migración de aplicaciones licenciadas hacia herramientas Open Source, es decir de código libre; también está enfocado en reducir la complejidad que involucra la instalación, configuración y manejo de soluciones Open Source vinculadas al proyecto.

El proyecto tiene como objetivo examinar la eficiencia de la herramienta de código libre RTAI, diseñado para Debian para el control de aplicaciones de tiempo real, probar la viabilidad de su instalación en los sistemas operativos Ubuntu, Fedora y Centos.

La investigación realizada está estructurada por 4 capítulos, cada uno de ellos con enfoques individuales basados en análisis y presentación de resultados.

El desarrollo de los capítulos de este trabajo comprende el capítulo Uno con la descripción de las herramientas de código libre a utilizar, el capítulo Dos con el detalle de los procesos de instalación de las herramientas en cada sistema operativo.

El contenido del capítulo Tres abarca la mecánica de evaluación de desempeño y su análisis, de las herramientas utilizadas por el usuario durante la ejecución de un proceso de tiempo real. El capítulo Cuatro muestra el desempeño de las herramientas de control remoto de aplicaciones de tiempo real.

La valoración del desempeño de las herramientas, RTAI y Scilab, y de los controladores de Comedi es realizada mediante pruebas con aplicaciones de tiempo real, de generación de onda seno, aplicaciones de control automático de una planta de control de nivel de fluido de lazo abierto y de lazo cerrado.

Se calificará el desempeño de las herramientas en cada sistema operativo cuantificando la capacidad de respuesta gráfica de los sistemas operativos durante la ejecución de las aplicaciones de tiempo real, los cambios del tiempo de estabilización de la planta y los tiempos de respuesta de los sistemas operativos adaptados mediante las herramientas de medición provistas por la herramienta RTAI. También se comprobará las capacidades y limites de funcionamiento del sistema de control remoto de aplicaciones de tiempo real, por criterio de máximo rango de frecuencia de trabajo, el cual está conformado por el servidor RTAI-XML y la herramienta cliente JRtaiLab.

CAPITULO 1

DESCRIPCIÓN DE HERRAMIENTAS UTILIZADAS EN INSTALACIÓN DE LA INTERFAZ PARA APLICACIONES EN TIEMPO REAL (RTAI)

1.1. Interfaz para Aplicaciones en Tiempo Real-RTAI.

1.1.1. Sistemas en Tiempo Real.

Definición de Sistema de Tiempo Real.

Un sistema de tiempo real (STR) es definido como todo sistema capaz de garantizar que los procesos o tareas que se encuentran bajo su control sean ejecutados o atendidos dentro de intervalos mínimos de tiempo relativamente invariables [1].

En términos informáticos al tratar con sistemas de tiempo real hay que considerar dos aspectos, la latencia, o tiempo transcurrido desde la ocurrencia de un evento hasta que el mismo sea atendido, y el jitter, o variaciones de tiempo en el período normal de ocurrencia de eventos periódicos, como es el casos de actividades programadas.

Típicamente un sistema de tiempo real representa una computadora de sistema de control que maneja y coordina las actividades de un sistema controlado, que puede ser visto como el ambiente con el que interactúa el computador. La interacción en estos sistemas es bidireccional, dígase a través de varios sensores (comunicación ambiente-computador) y actuadores (comunicación computador-ambiente), y se caracterizan por las limitaciones en los tiempos de corrección, siendo esto último, el tiempo que le toma al sistema en responder ante un evento.

En los sistemas de tiempo real los intervalos de tiempo en que se ejecutan las tareas se definen por un esquema de activación y por un plazo de ejecución. En lo que respecta al esquema de activación puede ser periódico, es decir en intervalos regulares o también puede ser aperiódico, es decir en respuesta a sucesos externos no regulares.

Todos los sistemas tienden a tener baja latencia, pero un sistema de tiempo real requiere que la latencia y el jitter sean tiempos predeterminados y constantes sin importar la cantidad de carga en el procesamiento. La cualidad de tiempo real se la puede clasificar de tres formas [2]:

Tiempo Real Estricto: Todas las acciones deben ocurrir dentro de un plazo especificado (Hard Time).

Tiempo Real Flexible: Se pueden perder plazos de vez en cuando, el valor de la respuesta decrece con el tiempo.

Tiempo Real Firme: Se pueden perder plazos ocasionalmente, el valor de una respuesta tardía no tiene valor.

Concepto de Adaptación de un Sistema para Funcionamiento en Tiempo Real.

La adaptación de un sistema para que trabaje en tiempo real, depende exclusivamente de su velocidad de respuesta, por lo tanto obedece a dos parámetros, la velocidad de procesamiento del sistema y la priorización de procesos que debe atender el sistema.

El método aplicado consiste en colocar una capa interfaz entre el hardware y el kernel estándar. Esta capa controla la ejecución de las tareas de tiempo real y ejecuta el kernel estándar como una tarea en background, es decir, el kernel estándar sólo se ejecuta cuando no hay tareas de tiempo real pendientes. Otra de las finalidades de esta adaptación es disminuir la carga de procesamiento, mediante la reducción de la carga de algunos módulos del kernel que para fines de operación de un sistema de tiempo real son esencialmente innecesarios.

Fundamentos de Funcionamiento de Sistema en Tiempo Real

Algunas de las funcionalidades de un sistema operativo general son la gestión de procesos, gestión de memoria, interacción con el hardware, servidor de ficheros, servidor de comunicaciones. Todo sistema operativo de propósito general trabaja de tal forma que sus tareas manejan prioridades que pueden cambiar a lo largo del tiempo, y que pueden ser interrumpidas por algún proceso y ser postergadas hasta que lo establezca su nueva prioridad. Esto se debe a que estos sistemas operativos están diseñados primordialmente para atender de forma inmediata cualquiera petición del usuario. La figura 1.1 nos detalla de forma gráfica la Arquitectura de un Sistema Operativo de Uso General.

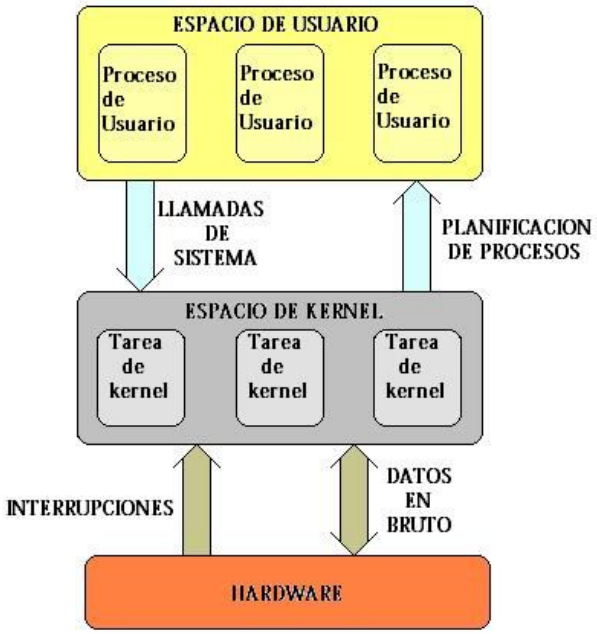


Figura 1.1 Arquitectura de Kernel de uso general

Por otra parte, un sistema de tiempo real tiene como funcionalidad principal el servicio de aplicaciones para una respuesta en un intervalo de tiempo predeterminado y se centra en atender de forma primordial dos tipos de interrupciones, las interrupciones del procesador y las interrupciones de los periféricos. Esta característica está orientada a garantizar que ninguna tarea diseñada para tiempo real, sea postergada por alguna interrupción producida por alguna otra tarea.

El funcionamiento básico de las herramientas utilizadas consiste en colocar una capa, la cual es el nuevo administrador central de las gestiones del sistema operativo. Esta capa, discrimina las tareas en: tareas comunes y en tareas de tiempo real. Esta categorización es realizada disminuyendo la

prioridad del kernel y de todo proceso dependiente de él .Las tareas de tiempo real tienen la más alta prioridad y no pueden ser postergadas por tareas comunes. Comúnmente el manejo de interrupciones es trabajo del kernel, pero, ya que el kernel tiene una prioridad baja y las tareas de tiempo real están vinculadas con el tiempo y el hardware, las interrupciones también pasan a ser manejadas por esta nueva capa. Caso contrario las interrupciones solo podrían ser atendidas luego de la finalización de una tarea de tiempo real.

El parche colocado dentro del proceso de adaptación del sistema, permite crear aplicaciones que corran bajo órdenes de usuario, es decir dentro del espacio de usuario el cual es manejado por el kernel, y que a pesar de esto, se ejecuten como aplicaciones de tiempo real.

1.1.2. Tareas de Tiempo Real

Sobre los sistemas de tiempo real se pueden ejecutar varias actividades, las cuales pueden ser categorizadas como: actividades con tiempo crítico y actividades sin tiempo crítico. Es deseable que las actividades de tiempo crítico y las actividades sin tiempo crítico coexistan en un mismo sistema de tiempo real. Ambos tipos de actividades son conocidas como tareas, las cuales requieren ser ejecutadas en un plazo de tiempo límite conocidas como tareas de tiempo real [3].

Las tareas de tiempo real tienen los siguientes atributos:

-Requerimiento de tiempo: Las tareas de tiempo real pueden ser periódicas o no periódicas. Una tarea periódica tiene que ser repetida una vez cada cierto

periodo, pero las tareas no periódicas pueden ser llamadas a ejecución por un evento dinámico.

-Requerimiento de recursos: Una tarea de tiempo real puede requerir acceso a ciertos recursos, como dispositivos de entrada y salida, estructuras de datos, archivos y bases de datos.

-Requerimientos de Comunicaciones: Las tareas deben de tener permiso de comunicarse con mensajes.

-Limitaciones de Concurrencia: Las tareas deben de contar con el permiso de acceso concurrente a recursos, garantizando que la consistencia de los recursos no sea violada.

1.1.3. Interfaz de Aplicación de Tiempo Real-RTAI

Interfaz de Aplicación de Tiempo Real o RTAI, por sus siglas en ingles REAL TIME APPLICATION INTERFACE, es la implementación de un sistema operativo de tiempo real estricto para GNU/Linux de forma que añade un pequeño kernel de tiempo real bajo el kernel estándar de GNU/Linux y trata al kernel de éste como una tarea de menor prioridad.

RTAI proporciona una opción llamada LXRT para facilitar el desarrollo de aplicaciones de tiempo real en el espacio de memoria del usuario. RTAI trata el kernel estándar como una tarea de prioridad menor, lo que hace posible que se ejecute cuando no hay ninguna tarea con mayor prioridad ejecutándose. Las tareas básicas de tiempo real son implementadas como módulos del kernel. RTAI se ocupa de manejar de forma inmediata las interrupciones de

periféricos antes de que lo haga el kernel, luego de ejecutar las posibles acciones de tiempo real que hayan podido ser lanzadas por efecto de la interrupción, pasan a ser atendidas por el kernel como cualquier otro proceso dentro de un sistema general.

RTAI utiliza el método Micro-Kernel de modificación de kernel, método que añade otro kernel al sistema, aparte del kernel propio de GNU/Linux. Este kernel añadido en realidad es una interfaz entre el hardware y el kernel estándar, lo que se llama tradicionalmente HAL (capa de abstracción de hardware o Hardware Abstraction Layer). Esta capa controla la ejecución de las tareas de tiempo real y ejecuta el kernel estándar como una tarea en background, es decir, el kernel estándar solo se ejecuta cuando no hay tareas de tiempo real pendientes. La Figura 1.2 nos muestra la arquitectura de un microkernel.

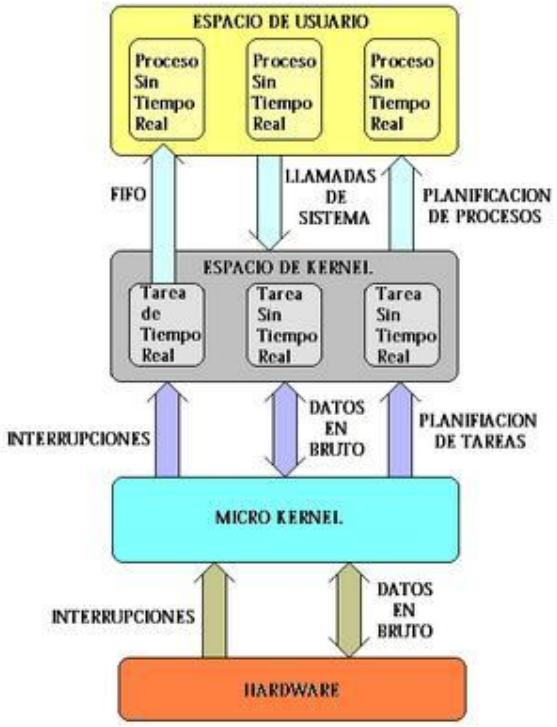


Figura 1.2 Arquitectura Micro kernel

Este micro-kernel capta las interrupciones de hardware y se asegura que las tareas de tiempo real se ejecuten con la mayor prioridad posible para minimizar la latencia. Este mismo sistema es utilizado por RTLinux, otro sistema de tiempo real.

La figura 1.3 nos permite visualizar el esquema una vez que ha sido agregado los módulos de RTAI para su comportamiento en tiempo real.

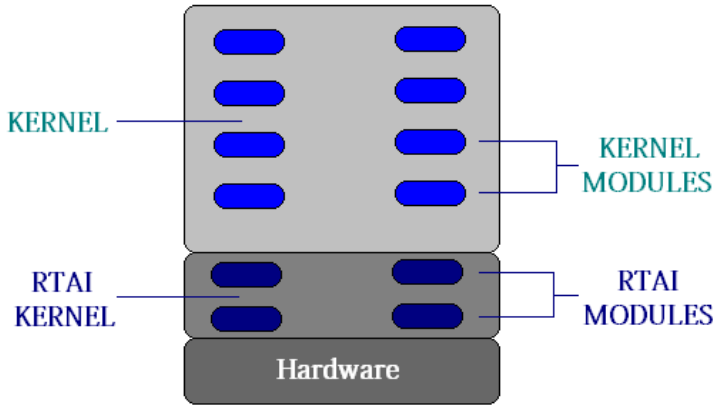


Figura 1.3 Esquema General del sistema operativo Linux con RTAI

RTAI está orientado a módulos, por lo que es necesario tratar acerca de los módulos para Linux de carga dinámica. Un módulo es un controlador de un dispositivo o servicio que pueden cargarse o descargarse cuando el usuario o el dispositivo lo solicitan. El uso de módulos dinámicos de kernel, permite escribir porciones de kernel como objetos separados que pueden ser cargados y suprimidos mientras el sistema este ejecutándose. Para usar RTAI es necesario cargar los módulos que implementan cualquier capacidad de RTAI que se pueda necesitar.

Compréndase a la clasificación de las aplicaciones realizadas en RTAI como aplicaciones en tiempo real, debido a que son capaces de responder inmediatamente a los eventos externos durante su ejecución.

La Figura 1.4 muestra con mayor detalle la arquitectura básica de RTAI, que es muy similar a la de RTLinux. Las interrupciones se originan en el

procesador y en los periféricos. Las originadas en el procesador (principalmente señales de error como división por cero), son manejadas por el kernel estándar, pero las interrupciones de los periféricos (como los relojes) son manejadas por RTAI Interrupt Dispatcher. RTAI envía las interrupciones a los manejadores del kernel estándar de linux cuando no hay tareas de tiempo real activas. Las instrucciones de activar/desactivar las interrupciones del kernel estándar son reemplazadas por macros que se enlazan con las instrucciones de RTAI. Cuando las interrupciones están desactivadas en el kernel estándar, RTAI encola las interrupciones para ser repartidas después de que el kernel estándar haya activado las interrupciones de nuevo.

Adicionalmente, se puede ver en la Figura 1.4 el mecanismo de comunicación entre procesos (IPC), que está implementado de forma separado por Linux y por RTAI. También existe un planificador (scheduler) distinto para Linux y para RTAI.

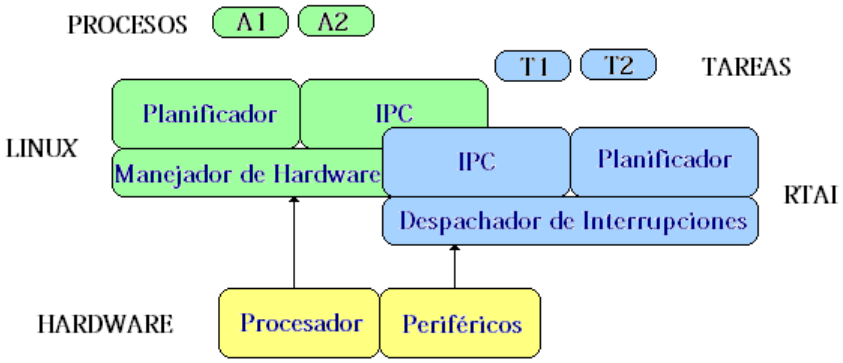


Figura 1.4 Arquitectura de RTAI

1.1.4. Aplicaciones en Tiempo Real e Interacción con el Medio

Una de las finalidades para los sistemas de aplicaciones en tiempo real, es el uso industrial, esencialmente el control automático de dispositivos dentro de una planta, por lo que este tipo de sistemas debe interactuar con equipos, de tal forma que esta interacción no constituya un tiempo de retraso que haga que el sistema pierda su calidad de acción en tiempo real.

Los mecanismos necesarios para la comunicación del sistema con el hardware de los dispositivos de control, son librerías estándares conocidas como drivers. En un sistema de administración de plantas por medio de computadores, el control de los dispositivos industriales como motores, bombas, o pruebas de bajo nivel de señales eléctricas moduladas, se las realiza con un hardware conectado directamente a la tarjeta madre de la computadora. Este hardware es la tarjeta de adquisición de datos, la cual requiere de librerías de control para poder ser manejada.

1.1.5. Driver de Dispositivo

Un aplicativo de un sistema de tiempo real suele involucrar el manejo de circuitería o sistemas eléctricos de control o emisión de datos y señales, el sistema también deberá de contar con un método de comunicación con el hardware que interactuará directamente con los medios físicos, como por ejemplo la planta de una industria. Para esto, el sistema debe tener la capacidad de manejar aplicaciones personalizables, y librerías de comunicación con periféricos, por ejemplo tarjetas de adquisición de datos.

Un driver de dispositivo es una parte de un software que interactúa con una pieza particular de hardware, como una impresora, tarjeta de sonido, un driver de motor, etcétera. Éste traduce lo primitivo, es decir comandos dependientes con los cuales el productor del hardware nos permite configurar, leer y escribir la electrónica de la interfaz del hardware en llamadas de funciones genéricas más abstractas para las aplicaciones de programadores.

1.2. Tarjeta de Adquisición de Datos NI 6024E

1.2.1. Introducción

La tarjeta PCI-6024E tiene16 canales de entradas analógicas, dos canales de salidas analógicas un conector de 68 pines y 8 líneas digitales de Entrada/Salida. Esta tarjeta utiliza el sistema de control de tiempo DAQ-STC para funciones de tiempo relacionado, la cual consiste en tres grupos de temporizadores que controlan las entradas analógicas, las salidas analógicas, y las funciones de tiempo y conteo de propósito general. Estos grupos incluyen un total de siete contadores de 24 bits y tres contadores de 16 bits, y una máxima resolución de tiempo de 50 ns. DAQ-STC hace posible operaciones como almacenamiento de generación de pulso, muestreo de tiempo equivalente, y cambio de tasa de muestreo sin ruptura. [4].

La tarjeta es completamente compatible con las Especificaciones para Bus Local PCI Revisión 2.1. Esta especificación permite al sistema PCI colocar la dirección de memoria de la tarjeta base y el canal de interrupción sin que el

usuario intervenga. Para instalar la tarjeta PCI-6024E, con la computadora apagada, se debe retirar la tapa del case e insertar la tarjeta en el socket PCI, si es posible ajustar la tarjeta con los tornillos del case, y cerrar el case. Una vez que se encienda la computadora, la tarjeta podrá ser reconocida como hardware instalado [4]

1.2.2. Entradas Analógicas

La tarjeta posee tres tipos diferentes de entrada en lo que respecta a las entradas analógicas: referenciadas con final simple (RSE), no-referenciadas con final simple (NRSE), y entradas diferenciales (DIFF). Como se puede observar en la Figura 1.5 la tarjeta cuenta con una entrada programada como diferencial usa dos entradas analógicas, una entrada conectada a la entrada positiva del amplificador de instrumentación de ganancia programable (PGIA) y la otra entrada conectada a la entrada negativa de PGIA.

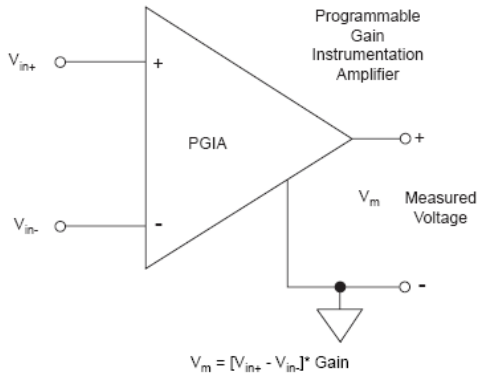


Figura 1.5 Diagrama de Entrada Diferencial de Tarjeta PCI-6024E

Una entrada configurada como RSE, usa una entrada analógica la cual conecta a la entrada positiva de PGIA. La entrada negativa del PGIA es internamente vinculado con la entrada analógica de tierra (AIGND). Una entrada configurada como NRSE, usa una entrada analógica la cual la conectada a la entrada positiva del PGIA. La entrada negativa del PGIA la conecta a la entrada analógica (AISENSE).

La tabla I nos presenta los Rangos de precisión de la Tarjeta, se puede notar que los rangos de entrada son bipolares y que cambian con la ganancia programada. Cada entrada puede ser programada con una ganancia única, que puede tomar valores como 0.5, 1.0, 10 o 100 para maximizar la resolución del convertidor analógico-digital de 12 bits.

Ganancia	**Rango de Entrada**	**Precisión**
0.5	-10 a 10 V	4.88 mV
1.0	-5 a 5 V	2.44 mV
10.0	-500 a 500 mV	244.14 uV
100.0	-50 a 50 mV	24.41 uV

Tabla I Rangos de Precisión de la Tarjeta PCI-6024E

La tarjeta puede hacer la lectura de múltiples canales a la misma velocidad que hace la lectura de un solo canal, sin embargo la velocidad de asentamiento de datos puede alterarse cuando se cuenta con cambios drásticos de ganancia entre canal y canal, es decir para no alterar este tiempo las ganancias deben ser constantes y la impedancia de la fuente de la señal que se lee debe ser baja.

1.2.3. Salidas Analógicas

Esta tarjeta posee dos canales de salidas analógicas. Tiene un rango desde -10V a +10V

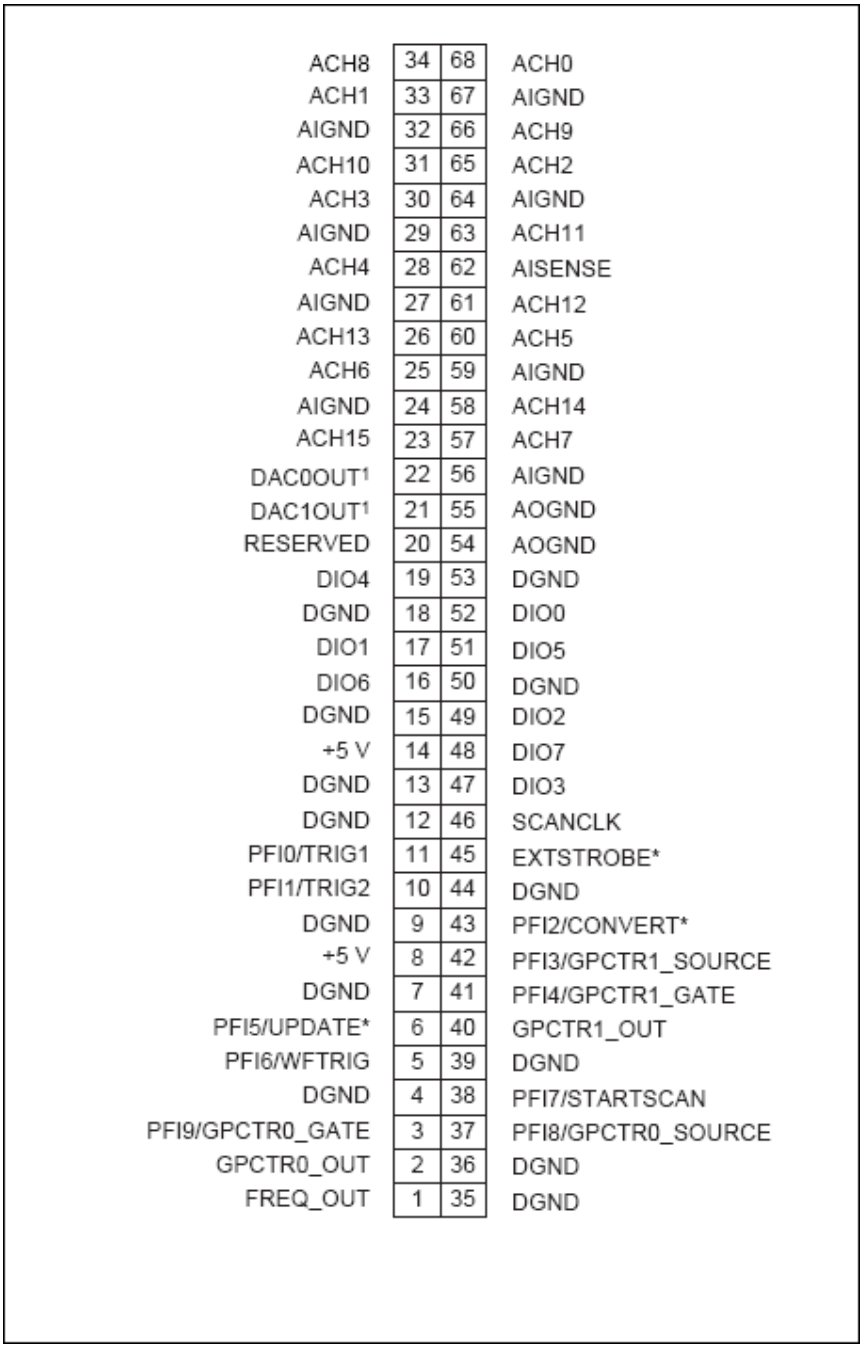
1.2.4. Entradas/Salidas Digitales

La tarjeta contiene 8 líneas de entradas/salidas digitales (DIO0, DIO1,…, DIO7). Cada línea puede ser configurada individualmente por software como entrada o como salida. Al iniciar el sistema, todas estas entradas/salidas se setean a alta impedancia.

La tarjeta requiere frecuencias de funcionamiento para poder generar las señales de tiempo para el control de los convertidores A/D, actualizaciones del DAC, y para propósitos generales de las señales de entrada y de salida. La tarjeta en este caso puede utilizar a su frecuencia de 20 MHz o las señales de tiempo de la tarjeta madre recibidas por el bus RTSI.

1.2.5. Conexión de Señales

En la siguiente Figura 1.6 se muestra la distribución de pines de la tarjeta PCI NI-6024E.



ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7
DAC0OUT[1]	22	56	AIGND
DAC1OUT[1]	21	55	AOGND
RESERVED	20	54	AOGND
DIO4	19	53	DGND
DGND	18	52	DIO0
DIO1	17	51	DIO5
DIO6	16	50	DGND
DGND	15	49	DIO2
+5 V	14	48	DIO7
DGND	13	47	DIO3
DGND	12	46	SCANCLK
PFI0/TRIG1	11	45	EXTSTROBE*
PFI1/TRIG2	10	44	DGND
DGND	9	43	PFI2/CONVERT*
+5 V	8	42	PFI3/GPCTR1_SOURCE
DGND	7	41	PFI4/GPCTR1_GATE
PFI5/UPDATE*	6	40	GPCTR1_OUT
PFI6/WFTRIG	5	39	DGND
DGND	4	38	PFI7/STARTSCAN
PFI9/GPCTR0_GATE	3	37	PFI8/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND
FREQ_OUT	1	35	DGND

Figura 1.6 Distribución de Pines de la Tarjeta PCI-6024E

1.3. KERNEL DE LINUX

1.3.1. Introducción al Kernel Linux

Hablar de Linux es referirse al Kernel, el núcleo del sistema operativo. El kernel o núcleo de Linux se puede definir como el corazón del sistema operativo. Es el principal responsable de facilitar a los distintos programas el acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado.

Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador. En sencillas palabras es el encargado de que el software y el hardware del ordenador puedan trabajar juntos [5].

Las fuentes en C de cada versión de kernel cuentan con controladores para diversos dispositivos. Cuando se compila una versión, algunos de esos controladores pueden unirse al kernel estáticamente, otros pueden dejarse como módulos para cargarse o descargarse cuando la parte estática del kernel esté operando, y otros pueden ser excluidos del proceso de compilación (por lo tanto no podrán ser utilizados ni cuando el kernel esté

operando). Por lo tanto todo módulo que va a usarse en algún momento debe ser incluido previo a la configuración y la compilación del kernel. El proceso de agregar una funcionalidad o capacidad de manejo al kernel, se lo conoce como proceso de parchado.

1.3.2. Módulos

Un módulo de kernel es un simple objeto, o programa informático que permite al sistema operativo interactuar con un periférico que contiene rutinas y datos para cargar en el kernel mientras opera. Técnicamente un módulos puede ser programado como rutina, con la única restricción de que dos funciones deben ser provistas, `init_module()` y `cleanup_module()`, la primera es ejecutada una vez que el módulo este cargado y la segunda antes de que el módulo sea descargado del kernel.

Los módulos que se distribuyen con el kernel están ubicados en el directorio `/lib/modules/version` donde *`version`* es la versión del kernel, con la extensión .o, organizados en directorios que indican el tipo de dispositivo o propósito, por ejemplo *`fs`*, para sistema de archivos, *net*, para protocolos y hardware para redes.

Las funciones principales para el manejo de módulos son las siguientes:

- `Insmod`: inserta el módulo en el kernel que se encuentra corriendo.
- `Rmmod`: remueve el módulo del kernel que se encuentra corriendo.
- `Lsmod`: lista los módulos que se encuentran corriendo en el kernel.

- `Modprobe`: Emplea información de las dependencias de un módulo para cargar el módulo especificado, cargando antes todos los módulos de los cuales dependa.
- `Depmod`: Permite calcular las dependencias entre módulos.
- `Modconf`: Permite listar, cargar y descargar los módulos con menú.

1.3.3. Clasificación de Tipos de Kernel

No necesariamente se necesita un kernel para usar una computadora. Los programas pueden cargarse y ejecutarse directamente en una computadora, siempre que sus autores quieran desarrollarlos sin usar ninguna abstracción del hardware ni ninguna ayuda del sistema operativo.

A medida que los kernels se fueron desarrollando, se convirtieron en los fundamentos de lo que llegarían a ser los primeros núcleos de sistema operativo.

Existen cuatro grandes tipos de kernel:

- **Monolíticos**, aquellos que facilitan abstracciones del hardware subyacente realmente potentes y variadas.
- **Microkernel,** proporcionan un pequeño conjunto de abstracciones simples del hardware, y usan las aplicaciones llamadas servidores para ofrecer mayor funcionalidad.

- **Híbridos** (*microkernels modificados*) son muy parecidos a los microkernels puros, excepto porque incluyen código adicional en el espacio de kernel para que se ejecute más rápidamente.

- **Exokernel,** no facilitan ninguna abstracción, pero permiten el uso de bibliotecas que proporcionan mayor funcionalidad gracias al acceso directo o casi directo al hardware.

1.3.4. Arquitectura

La mayoría de los núcleos de Unix son monolíticos: cada una de las capas de núcleo está integrado en el núcleo y se ejecuta en modo kernel. Por el contrario, los sistemas operativos micro-núcleo demandan un conjunto muy reducido de funciones del kernel, en general, incluyendo sincronizaciones primitivas, un planificador simple, y un mecanismo de comunicación entre procesos. Aunque la investigación académica sobre los sistemas operativos es orientada hacia micro-núcleos, tales sistemas operativos son en general más lentos que el monolítico. Sin embargo, sistemas operativos micro-núcleo pueden tener algunas ventajas teóricas por encima de los monolíticos.

Los micro-núcleos comprometen a los programadores a adoptar un sistema de enfoque modular, ya que cualquier capa del sistema operativo es un programa relativamente independiente que debe interactuar con los demás capas a través de un software bien definido y limpio de interfaces. Por otra parte, un

sistema operativo micro-núcleo existente es fácilmente portado a otras arquitecturas, ya que todos los componentes de hardware dependientes suelen ser encapsulado en un código micro núcleo. Por último, estos sistemas tienden a hacer un mejor uso de memoria RAM que los núcleos monolítico.

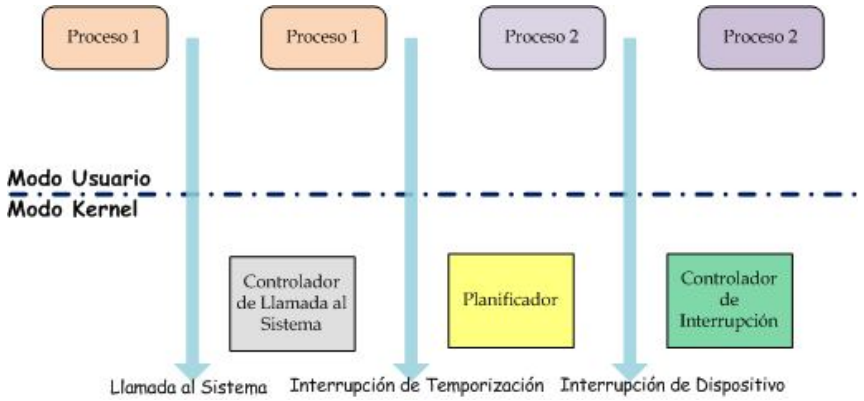
Hablar de la Arquitectura del Kernel es referirse a dos niveles de ejecución:

- El espacio de Usuario.
- El espacio del Kernel.

El espacio de usuario es donde las aplicaciones de usuario se ejecutan. Como parte de ésta sección de usuario está la biblioteca C de GNU (glibc); la cual proporciona la interfaz de llamada al sistema que se conecta con el núcleo y a su vez suministra el mecanismo para la transición entre la aplicación del espacio de usuario y el kernel. Es una de las funciones más importantes debido a que el núcleo y la aplicación de usuario ocupan diferentes espacios de direcciones protegidas. Y si bien cada proceso en espacio de usuario ocupa su propio espacio de direcciones virtuales, el núcleo ocupa un espacio de dirección única.

El espacio de kernel es el encargado de gestionar los recursos de hardware de la máquina de una forma eficiente y sencilla, ofreciendo al usuario una interfaz de programación simple y uniforme. El kernel, y en especial sus drivers, constituyen así un puente o interface entre el programador de aplicaciones para el usuario final y el hardware. Toda subrutina que forma parte del kernel tales como los módulos o drivers se consideran que están en el espacio del kernel.

La Figura 1.7 ilustra ejemplos de transiciones entre el espacio usuario y espacio kernel. El Proceso 1 realiza una llamada al sistema, donde el proceso pasa del modo usuario al modo Kernel y la llamada al sistema es atendida. Por lo tanto el proceso 1 reanuda la ejecución en modo usuario hasta que la interrupción de temporizador se produce y el planificador se activa en modo kernel. Un proceso de interrupción se lleva a cabo, y el Proceso 2 inicia la ejecución en modo usuario hasta que el dispositivo de hardware programa una interrupción. Como consecuencia de la interrupción, el Proceso 2 cambia a modo kernel y a servicios de interrupción [6].



. Figura 1.7 Proceso de Transición de Modo Usuario a Modo Kernel

1.3.5. Funcionalidades

Inmediatamente sobre el hardware se sitúa el kernel. A continuación un detalle de sus principales funciones:

- Administrar el hardware de manera coherente y justa mientras se le otorga un nivel de abstracción familiar, a través de las APIs, a las aplicaciones de nivel de usuario.
- Maneja dispositivos, administra los accesos de Entrada/Salida.
- Controla los procesos.
- Administra el uso compartido de memoria.

Aparte de las funcionalidades básicas, el conjunto de las funciones necesariamente no son proporcionadas por un kernel de sistema de explotación. Pueden establecerse estas funciones del sistema de explotación tanto en el espacio usuario como en el propio kernel. Su implantación en el núcleo se hace con el único objetivo de mejorar los resultados. En efecto, según la concepción del kernel, la misma función llamada desde el espacio usuario o el espacio núcleo tiene un coste temporal obviamente diferente. Si esta llamada de función es frecuente, puede resultar útil integrar estas funciones al núcleo para mejorar los resultados.

1.3.6. Versiones de kernel

En la actualidad se continúa lanzando nuevas versiones del núcleo Linux, estos son llamados núcleos “vanilla”, distribuidos por Linus Torvalds y el equipo de programadores; que significa que no han sido modificados por nadie. Muchos desarrolladores de distribuciones Linux modifican dicho núcleo en sus productos, principalmente para agregarle soporte a dispositivos o herramientas que no fueron oficialmente lanzadas como estables, mientras que algunas distribuciones, como Slackware, mantienen el núcleo vanilla [7].

Hasta que empezó el desarrollo de la serie 2.6 del núcleo, existieron dos tipos de versiones del kernel:

- *Versión de producción:* La versión de producción, era la versión estable hasta el momento. Esta versión era el resultado final de las versiones de desarrollo o experimentales.
- *Versión de desarrollo:* Esta versión era experimental y era la que utilizaban los desarrolladores para programar, comprobar y verificar nuevas características, correcciones, etc. Estos kernels solían ser inestables y no se debían usar sin saber lo que se hacía.

Numeración de Versiones Linux

Es importante saber interpretar los números de las versiones de las *series por debajo de la 2.6*:

Las versiones del núcleo se numeraban con 3 números, separados por puntos; de la siguiente forma: XX.YY.ZZ:

XX: Indica la serie/versión principal del núcleo. Solo han existido la 1 y 2.

YY: Indica si la versión es de desarrollo ó de producción. Un número impar, significaba que es de desarrollo, uno par, que es de producción.

ZZ: Indica nuevas revisiones dentro de una versión, en las que lo único que se había modificado eran fallos de programación.

La Figura 1.8 nos presenta un ejemplo de numeración del núcleo de la serie debajo de la 2.6

Figura 1.8. Numeración de Núcleo de Serie debajo de la versión 2.6

Con la *serie 2.6* del núcleo, el sistema de numeración así como el modelo de desarrollo han cambiado. Las versiones han pasado a numerarse con 4 dígitos y no existen versiones de producción y desarrollo. Las versiones del núcleo se numeran hoy en día con 4 dígitos, de la siguiente forma: AA.BB.CC.DD.

AA: Indica la serie/versión principal del núcleo.

BB: Indica la revisión principal del núcleo. Números pares e impares no tienen ningún significado hoy en día.

CC: Indica nuevas revisiones menores del núcleo. Cambia cuando nuevas características y drivers son soportados.

DD: Este dígito cambia cuando se corrigen fallos de programación o fallos de seguridad dentro de una revisión.

La figura 1.9 nos presenta un ejemplo de numeración del núcleo de la serie 2.6

Figura 1.9 Numeración de Núcleo de Serie 2.6

1.4. MesaLib, EFLTK, Comedi, ComediLib

1.4.1. MESALIB

Mesa es una implementación de código abierto de OpenGL; es una biblioteca de gráficos interactivos en 3-D. A pesar de no ser una implementación oficial su uso se ha extendido y su estructura, sintaxis e incluso la semántica de su API es como la de OpenGL. Mesa es una librería que se puede utilizar en prácticamente todas las plataformas y soporta varios tipos de aceleradores gráficos, también puede ser compilado como un renderizador de software solamente [8].

Una variedad de controladores de dispositivo permiten que Mesa sea utilizado en diferentes plataformas que van desde la emulación de software hasta su integración en hardware en los GPUs más actuales, procesadores que están ubicados en las tarjetas gráficas. Mesa fue diseñado originalmente para sistemas Unix/X11 y sigue siendo el mejor apoyo en esos sistemas. Todo lo

que se necesita es un compilador de ANSI C y el entorno de desarrollo X a usar Mesa.

Incluye controladores para una variedad de otras plataformas: Apple Macintosh, BeOS, NeXT, OS / 2, MS-DOS, VMS, Windows 9x/NT, y Direct3D.

Los detalles sobre algunos drivers tenemos a continuación:

- DRI: Controladores de hardware para el sistema X Window.
- Xlib: Controlador de software para el sistema X Window y sistemas operativos tipo Unix y Microsoft Windows.
- SLB.
- Dependencias: Xorg Libraries, Xorg Utilities, libdrm-2.4.14,expat-2.0.1 ,
- Opcionales: libxcb-1.4 and LessTif-0.95.2.

1.4.2. EFLTK

Es una herramienta desarrollada en base de librerías de software libre y de código abierto, es una extensión de la librería FLTK. Esta biblioteca es una mejora FLTK 2.0 con muchos cambios y modificaciones fundamentales como: soporte para XML, red, y base de datos. Para compilarlo, no es necesario el código fuente FLTK 2.0, EFLTK ya lo incluye. La biblioteca EFLTK, soporta varios sistemas operativos: Windows/ Unix / Mac, entre otros, es de licencia LGPL (Licencia Pública General Reducida de GNU); proporciona moderna funcionalidad GUI, soporte en gráficos a través de OpenGL que es el primer

ambiente para desarrollar aplicaciones portables, que muestren gráficos 2D y 3D, integra una emulación de GLUT (OpenGL Utility Toolkit).

Dentro de los requerimientos antes de su instalación tenemos los siguientes paquetes (Ubuntu):

XOrg Development packages (`xorg-dev`, `xserver-xorg-dev`, `libglu1-xorg-dev`, `libXext-dev`), `g++` (`g++-3.4`), `svn` (`subversion`), `gettext` [9].

1.4.3. COMEDI

Comedi es un proyecto de software libre; es una colección de drivers [10]. Estos drivers están implementados como una combinación de los siguientes elementos:

- Un módulo del kernel de Linux que proporciona las funciones más comunes.
- Módulos individuales para las funciones de bajo nivel para cada dispositivo.

Estos módulos son la combinación de tres elementos de software complementarios: un elemento genérico, independiente del dispositivo de la API; una colección de módulos del kernel Linux que implementan esta API para una amplia gama de tarjetas, y un usuario de Linux para la colección de espacio con una interfaz de programación orientada a desarrolladores para configurar y usar las tarjetas.

Los controladores se implementan como un módulo de memoria del núcleo Linux que proporcionan una funcionalidad común e individual como módulos de controladores de bajo nivel; además proporciona las herramientas y bibliotecas para un montón de tarjetas de adquisición de datos (DAQ). Los controladores de dispositivos normalmente deben tener acceso a direcciones específicas en el bus de datos, y por lo tanto se ejecutan en espacio de kernel.

Jerarquía de Dispositivos

Comedi organiza todo el hardware de acuerdo con la siguiente jerarquía genérica [11]:

Canal: Componente de hardware de más bajo nivel, que representa las propiedades de un canal único de datos, por ejemplo, una entrada analógica o una salida digital. Cada canal tiene varios parámetros, tales como: el rango de tensión, la tensión de referencia, la polaridad del canal (unipolar, bipolar), un factor de conversión entre las tensiones y las unidades físicas, los valores binarios "0" y "1", entre otros.

Sub-dispositivo: Conjunto de canales funcionalmente idénticos que se ejecuta físicamente en la misma (chip en una tarjeta) interfaz. Por ejemplo, un conjunto de 16 salidas analógicas idénticos. Cada sub-dispositivo tiene parámetros para: el número del canal y el tipo de los canales.

Dispositivo: Conjunto de sub-dispositivos que son físicamente implementados en la tarjeta misma de la interfaz. Por ejemplo, el dispositivo de National Instruments 6024E tiene un sub-dispositivo con 16 canales de

entrada analógica, otro sub-dispositivo con dos canales de salida analógica, y un tercer sub-dispositivo que es un conjunto de ocho entradas / salidas digitales. Cada dispositivo tiene parámetros para: la etiqueta de identificación de dispositivos del fabricante, etiqueta de identificación asignado por el sistema operativo (para discriminar entre las múltiples tarjetas de interfaz del mismo tipo), el número de sub-productos, etc.

Algunas tarjetas de interfaz tienen componentes adicionales que no encajan en la clasificación antes mencionada, como una EEPROM para almacenar la configuración, o insumos de calibración. Estos componentes especiales están también clasificados como "sub-dispositivos" en Comedi. Existen diferentes aplicaciones haciendo uso de Comedi, entre ellas tenemos:

- comedirecord : Un sencillo programa para grabar datos de un dispositivo Comedi. Utiliza la biblioteca Qt de Trolltech widget.
- Qt-Wave : Generador de forma de onda Trolltech librería de widgets Qt.
- RTXI :Interfaz para experimentos en tiempo real.
- Xoscope: Un osciloscopio digital para Linux.
- RTIC-Lab :Controles de laboratorio en tiempo real (RTIC-Lab), implementación real para controlador de tiempo y ambiente de simulación para Linux.
- ComediClientServer :Sistema de control y prueba de dispositivos de forma remota COMEDI (o local).
- Comedi2puredata : DAQ Análogo de entrada para el procesador de audio en tiempo real.

- Comedi-Lib : API de Perl para Comedilib.

1.4.4. ComediLib

Comedilib es una biblioteca en espacio de usuario que proporciona una interfaz amistosa con el desarrollador de dispositivos Comedi. En la distribución Comedilib encontramos la documentación, configuración y calibración de los servicios de acceso público y programas de demostración. Esta librería viene incluida en las distribuciones Linux.

1.5. Scilab-Scicos

Scilab es un lenguaje de programación de alto nivel para cálculo científico, interactivo de libre uso y disponible en múltiples sistemas operativos: Mac OS X, GNU/Linux, Windows; desarrollado por INRIA (Institut National de Recherche en Informatique et Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990 [12].

Desde el punto de vista del programa, Scilab es un lenguaje interpretado. En general, esto permite que se vuelvan más rápidos los procesos de desarrollo, ya que el usuario accede directamente a un lenguaje de alto nivel, con un rico conjunto de características proporcionadas por la biblioteca. El lenguaje Scilab fue desarrollado para ser extendido, de tal forma que los tipos de datos definidos por el usuario pueden ser definidos con sobrecargas de operaciones. Los usuarios Scilab

pueden desarrollar su propio módulo para que puedan resolver sus problemas particulares.

Posee cientos de funciones matemáticas y la posibilidad de integrar, compilar y enlazar dinámicamente otros lenguajes como Fortran y C: de esta manera, las bibliotecas pueden ser utilizados como si fueran una parte de Scilab características incorporadas. Desde el punto de vista de licencia, Scilab es un software libre en el sentido de que el usuario no paga por lo que Scilab es un software de código abierto.

1.5.1. Características de Scilab

En el comienzo mismo de Scilab, las características se han centrado en álgebra lineal. Pero, rápidamente, el número de funciones avanzadas para abarcar muchas áreas de la computación científica. La siguiente es una lista corta de su capacidad:

- Álgebra lineal, matrices dispersas.
- Polinomios y funciones racionales.
- Interpolación, aproximación lineal, cuadrática y no de optimización lineal.
- Solucionador de ecuaciones diferenciales ordinarias y diferenciado
- Solucionador de ecuaciones algebraicas.
- Clásico y control robusto, Lineal Matrix desigualdad de optimización, diferenciables y no diferenciables de optimización.
- Procesamiento de señales.
- Estadísticas.
- Programación con lenguaje simple y fácilmente asimilable.

- Posibilita al usuario la creación y definición de funciones propias.
- Soporta la creación y utilización de conjuntos de funciones destinadas a aplicaciones específicas denominados “Toolboxes”, por ejemplo: Control, Optimización, Redes Neurales, etc.

1.5.2. Scicos

Scilab ofrece funciones gráficas, incluyendo un conjunto de funciones de trazado, que permiten crear gráficos en 2D y 3D, así como interfaces de usuario. El entorno gráfico de Scilab denominado Scicos proporciona un ambiente sistema gráfico dinámico, modelador y simulador.

Con Scicos se puede crear diagramas de bloque para modelar y para simular el funcionamiento de sistemas dinámicos híbridos y para compilar sus modelos en código ejecutable. Scicos se utiliza para el proceso de señal, el control de sistemas, sistemas que hacen cola, y además para estudiar sistemas físicos y biológicos.

Sobre la base de Scicos desarrollada por INRIA, Xcos reemplaza Scicos en la distribución de Scilab desde la versión 5.2. Xcos proporciona cualidades nuevas, un plan de trabajo sincronizado con Scilab, y una continuidad garantizada por el Consorcio Scilab.

Características de Scicos [13]

Scicos ofrece eficientes soluciones para necesidades industriales y académicas. Proporciona funcionalidades para el modelado de sistemas

mecánicos, circuitos hidráulicos, de control, etc. Entres sus principales características tenemos:

- Modela, compila, y simula gráficamente los sistemas dinámicos.
- Combina los comportamientos continuos y del tiempo discreto en el mismo modelo.
- Selecciona elementos de las gamas de colores de bloques estándares.
- Permite añadir nuevos bloques del programa escritos en lenguajes como C, FORTRAN, o Scilab.
- Funcionalidad para simulaciones en proceso discontinuo de ambiente Scilab.
- Genere el código de C del modelo de Scicos usando un generador de código.
- Funcione las simulaciones en tiempo real con los dispositivos verdaderos usando Scicos-HIL.
- Genera los ejecutables en tiempo real para el control con Scicos-RTAI.
- Simula los sistemas de Comunicaciones Digitales con Scicos-ModNum.
- Utiliza los bloques implícitos desarrollados en Modelica.

1.6. Software Instalado en Ubuntu

Ubuntu

Antes de comenzar con el proceso de instalación de las herramientas principales, se debe instalar paquetes, es decir programas, librerías o códigos pequeños que son requeridos por los programas principales. En software libre estos paquetes son

utilizados en la elaboración y ejecución de varios programas, es decir, pueden ser utilizados por varias aplicaciones sin que sea exclusivo de alguna de ellas, de esta forma se reduce espacio lógico y procesamiento. Por lo tanto los programas que son construidos en base a estos paquetes no podrán funcionar sin que estos estén instalados dentro del sistema operativo, por este motivo se los conoce como dependencias.

Las dependencias, pueden ser utilizadas durante el proceso de instalación, como lo es en este caso las dependencias de uso general, como también pueden ser para el uso de herramientas específicas como son las dependencias de RTAI.

Entre los tipos de dependencias que se requieren en el proceso de instalación están las dependencias generales [14], utilizadas por los proceso de configuración e instalación de herramientas, las cuales son: `cvs, svn, build-essential, checkinstall`.

También son necesarias dependencias para el proceso de levantamiento las cuales son: `libncurses5-dev, kernel-package y fakeroot.`

Luego se encuentran las dependencias exclusivas para cada herramienta las cuales se detallan a continuación.

Dependencias de RTAI

`libxmu-dev, libxi-dev, doxygen, autoconf, automake, libtool`

Dependencias de COMEDILIB

bison, flex, python-dev

Dependencias de COMEDI-CALIBRATE

libboost-program-options-dev, libgsl0-dev,

Dependencias de SCILAB-4.1.2

g77, gfortran, sablotron, tcl8.4, tk8.4, tcl8.5-dev, tk8.5-dev, xaw3dg-dev, libpvm3, libgtkhtml2-dev, libzvt-dev, libvte-dev

Dependencias de SCILAB-5

swig, pvm-dev, gettext,

Dependencias de MESALIB

X11proto-xext-dev, xlibs-static-dev, libxext-dev, libxt-dev, libglu1-mesa-dev

Dependencias de QRTAILAB

```
libqt4-dev, libqwt5-qt4-dev
```

Estas dependencias, también poseen dependencias para poder ser instaladas por lo que el proceso de instalación, constituye la instalación de una red de paquetes predeterminados y en un orden especifico. Estas sub-dependencias, se detallan en el ANEXO A

1.7. Software Instalado en Fedora y Centos

De la misma forma que con el sistema operativo Ubuntu, en Fedora y en Centos, para poder proceder con la instalación de herramientas principales es obligatorio contar con las dependencias de todas las herramientas. Dado que las herramientas principales fueron construidas en base un grupo específico de dependencias, las dependencias instaladas en Ubuntu no son diferentes en funcionalidad respecto a las dependencias instaladas en Fedora y en Centos. Esencialmente los cambios que existen entre las dependencias de estos sistemas, en ciertos casos son exclusivamente solo cambios en los nombres, como también se da el caso en que un paquete deba ser reemplazado por dos o más paquetes o dos o más paquetes puedan ser reemplazados con un solo paquete.

Muchos de las dependencias que son instaladas individualmente en el proceso de instalación de las herramientas en Ubuntu, sin instalados como un grupo de

paquetes cuando se realiza la instalación de `Development Tools`, en el proceso de instalación en Fedora y en Centos.

A continuación se muestra el listado de dependencias instaladas en Fedora y en Centos. Para un detalle más profundo refiérase al ANEXO A

Dependencias GENERALES, KERNEL Y RTAI

```
cvs subversion Development Tools checkinstall intltool ncurses
ncurses-devel libXi-devel libXmu-devel
```

Dependencias de COMEDILIB

```
python-devel python flex
```

Dependencias de COMEDICALIBRATE

```
gsl-devel boost
```

Dependencias de SCILAB-4.1.2

```
sablotron pvm compat-gcc-34-g77 tcl-devel tk-devel gtkhtml2-
devel Xaw3d-devel
```

Dependencias de MESALIB

```
xorg-x11-proto-devel libXext-devel libXt-devel
```

Dependencias de QRTAILAB

```
qt-devel
```

CAPITULO 2

JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROCESO DE INSTALACIÓN DE RTAI EN DISTRIBUCIONES LINUX: UBUNTU, FEDORA Y CENTOS.

En este capítulo se describe las características generales del tipo de sistema orientado para el control de dispositivos en tiempo real, y se argumenta el uso de las herramientas orientadas a atender cada requisito que demanda la puesta en funcionamiento de un sistema de tiempo real.

Posteriormente, se argumenta el orden del proceso de instalación de cada bloque de herramientas que es requerido para poner en línea cualquiera de los sistemas operativos, Ubuntu, Fedora o Centos adaptados para trabajar en tiempo real. Se presenta un análisis de las limitantes en la selección de los sistemas operativos bajo los cuales se han realizados las pruebas de instalación, y la factibilidad del reemplazo o descarte del uso de ciertos paquetes de software en los sistemas operativos de Fedora y Centos, en relación a Ubuntu.

2.1. Justificación de Proceso de Instalación

El problema a tratar es la elaboración de un sistema Open Source para el control de equipos y mecanismos que requieren respuestas inmediatas y automáticas. El primer punto a abordar en este tipo de problemas es la necesidad de contar con un mecanismo de manejo personalizable, adaptable dependiendo del tipo de trabajo, por ejemplo la forma de manejo de las señales que se envían y se reciben de los equipos que conforman un planta , el tipo de respuesta de los equipos a los cuales se desea controlar, por lo tanto, se deduce que el sistema debe de contar con una herramienta que permita al usuario elaborar aplicaciones que se adapten a sus necesidades.

El segundo punto es permitir al usuario interactuar con dispositivos e instrumentos, mediante interfaces de comunicación como tarjetas de adquisición de datos,

dispositivos tales como equipos electrónicos de lectura de magnitudes físicas, bombas, motores, válvulas u otros equipos sin verse obligado a manejar herramientas propietario.

El tercer punto es garantizar la respuesta inmediata ante algún efecto externo, es decir que el sistema sea capaz de generar respuestas ante perturbaciones del medio, en fracciones mínimas de tiempo preestablecidas y que no se alteren pese a la carga del sistema.

El cuarto punto es brindar al usuario una herramienta para visualización en tiempo real de de entrada, de salida o señales internas al sistema que el usuario mismo haya diseñado, y otorgarle la capacidad de alterar las magnitudes de las señales que estén bajo su control.

En respuesta a cada uno de los requerimientos para este tipo de sistemas, se realiza la instalación ordenada de cada módulo para atender cada uno de los requerimientos.

Con respecto a la herramienta de elaboraciones de aplicaciones para el usuario, dentro del listado de paquetes instalados, el software que permite la elaboración de estas aplicaciones personalizadas es Scicos, el cual es uno de los módulos del programa Scilab, el cual se instala en el último bloque del proceso de instalación. El programa Scilab presenta mucha similitud con el programa Matlab [15]. Scicos es el módulo de Scilab que permite la elaboración de diagramas y esquemas de forma gráfica, lo cual permite abstraer la programación que desea hacer el usuario.

En relación las interfaces físicas de interacción con el medio, por ejemplo una tarjeta de adquisición de datos, los paquetes instalados Comedi y ComediLib, son respectivamente el bloque de controladores de la tarjeta de adquisición de datos que se carga como un módulo en el kernel, y el conjunto de librerías que permite al usuario mediante programación la lectura y escritura de datos en la tarjeta. Ambos paquetes brindan soporte para varias clases de tarjetas.

Para poder asegurar la priorización de proceso de control de señales, y establece que las velocidades de respuesta del sistema no se alteren, se instala el paquete de software RTAI el cual también es responsable de que se puedan realizar lecturas y escrituras de las aplicaciones de control en tiempo real.

Para que el usuario pueda visualizar señales y realizar cambios en las magnitudes de las mismas, se instala de forma paralela a RTAI, el programa RtaiLab, el cual está incluido dentro del paquete de instalación de RTAI. Es por este software que es necesaria la instalación de algunas librerías para el diseño de aplicaciones gráficas. En la Figura 2.1 se muestra la disposición lógica que tendrá cada paquete instalado dentro del computador en el cual se colocará la tarjeta de adquisición de datos.

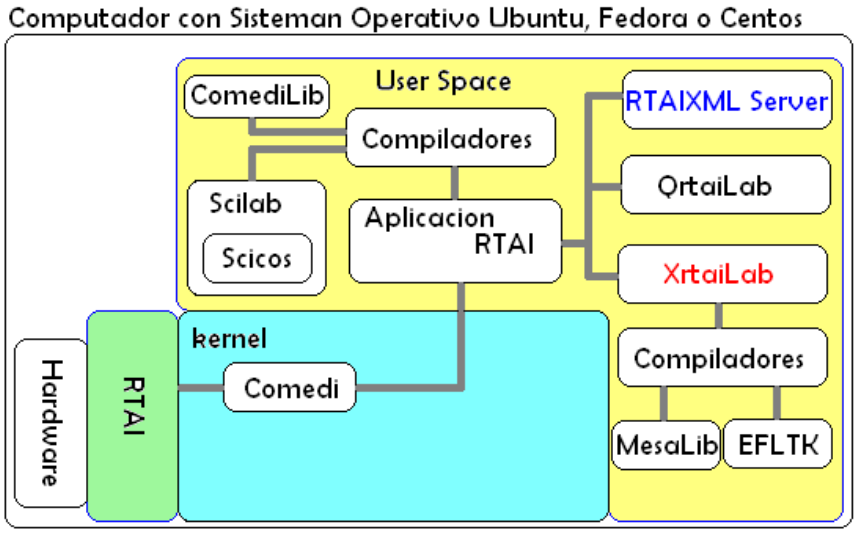


Figura 2.1 Esquema de Herramientas dentro de un Sistema Operativo con RTAI.

El proceso de instalación posee un orden específico ya que algunas de las herramientas requieren comprobar la instalación previa de otra herramienta, como es el caso de esencial de la sección de instalación de los paquetes RTAI y Comedi, los cuales requieren el uno del otro para poder ser instalados.

La Figura 2.2 muestra el diagrama del proceso de instalación de los paquetes.

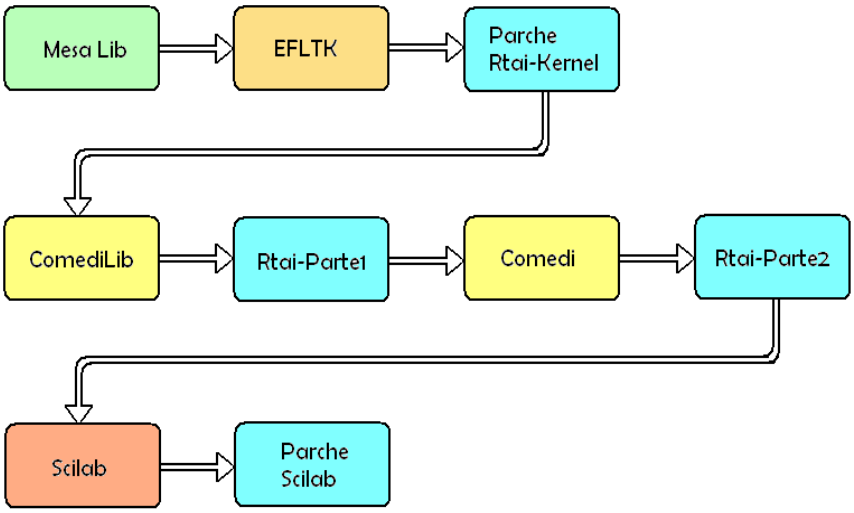


Figura 2.2 Esquema de Proceso de Instalación de Herramientas para RTAI.

Las versiones de los programas, parche y bloque de drivers señalados en la figura 2.2, están detalladas en la sección 2.2 y en los Apéndices A y B.

Para poder realizar el proceso de instalación de las aplicaciones y software detallados en la Figura 2.2, se debe de instalar con anterioridad un bloque de dependencias, las cuales han sido catalogadas en este documento como Dependencias Previas, las cuales son diferentes para Ubuntu, Fedora y Centos tienen la misma dependencia y la misma herramienta de instalación llamada yum.

Las Dependencias Previas están detalladas en el Apéndice A, para el sistema operativo Ubuntu y en el Apéndice para los sistemas operativos de Fedora y Centos, las versiones de las mismas son las cuales se instalan por defecto con yum en Fedora 8 y en Centos 5.5. Las versiones de las Dependencias Previas en la

instalación en Ubuntu son las versiones por defecto que se instalan utilizando la herramienta de instalación `apt-get`, en Ubuntu versión 8.4.3.

2.1.1. Viabilidad de instalación por Distribución Linux: Ubuntu, Fedora, Centos.

2.1.1.1. Ubuntu

Dado que RTAI fue desarrollado en base a la arquitectura e infraestructura del sistema operativo Debian, la instalación en Ubuntu, en términos de paquetes requeridos, puede ser resuelto de forma inmediata ya que el sistema de archivos, métodos de instalación y paquetes de Ubuntu son creados en base a Debian.

Ya resueltas las dependencias previas, y ya habiendo constatado que Ubuntu posee los paquete indispensables para la instalación, es imprescindible comprobar que las versiones de cada paquete sean las versiones requeridas que deben de tener cada uno de ellos para poder realizar correctamente la instalación.

Dado que gran parte de los paquetes a instalar precisan ser configurados y compilados, las herramientas cuyas versiones son las más importantes, son la de los compiladores y de las herramientas de configuración. El lenguaje de programación que más frecuentemente se usa durante el proceso de instalación es el lenguaje C, por este motivo es obligatorio instalar un compilador de este lenguaje, el compilador apropiado es GCC, que por requerimientos de algunos

paquetes y en especial, por requerimiento de Scilab debe de ser de versión 4.1.2.

Mediante pruebas realizadas para este proceso de instalación y configuración, se ha constatado que Ubuntu 8.4.3 posee la versión correcta de compilador GCC, como también las versiones apropiadas para cada uno de los paquetes del bloque de Dependencias Previas. Una vez finalizadas las pruebas, se ha comprobado cada uno de los paquetes para compilación y soporte, con sus respectivas versiones y se ha concluido que Ubuntu 8.4.3, permite la instalación completa y funcional de las herramientas RTAI, Comedi y Scilab. Versiones diferentes de Ubuntu poseen compiladores con versiones distintas a las indicadas anteriormente, por lo que si se usa las versiones de RTAI, Comedi y Scilab que han sido usadas en el proceso de instalación, por lo que el bloque de herramientas no puede ser correctamente instalado.

Es importante destacar que debido a la versión con la que se ha trabajado para este proceso de instalación de RTAI, se ha seleccionado inalterablemente las versiones de todos y cada uno de los paquetes y software. Si se altera una de estas versiones, la instalación no puede ser concluida o en caso de que la versión errónea sea la de un paquete, cuya versión no precise comprobación durante el proceso de instalación., las herramientas no producen aplicaciones correctamente funcionales. Es decir, si se desea alterar alguna de las

versiones de alguno de los paquetes instalados, es probable que sea necesario altear todas las versiones de los paquetes.

2.1.1.2. Fedora y Centos

El sistema RTAI ha sido diseñado originalmente para Debian, por lo que para sistemas operativos como Fedora y Centos, que provienen del sistema operativo Red Hat, se presenta como obstáculo inicial, la identificación de paquetes que puedan substituir a los paquetes que se instalan durante un proceso de instalación en Debian o sistema operativos provenientes de Debian, y las versiones de los paquetes que se requieren instalar.

Mediante investigación y pruebas se ha comprobado que algunos paquetes de Ubuntu pueden ser instalado por el comando *yum* en los sistemas operativos de Fedora y de Centos. Otros paquetes que no pueden ser instalados directamente pueden ser reemplazados por conjuntos de herramientas y paquetes homólogos propios de estos sistemas operativos. Algunos paquetes que no pueden ser encontrados directamente en los repositorios o no han podido ser reemplazados, han sido instalados desde su código fuente, como es el caso de Sablotron y Checkinstall, ambos con versiones específicas. Otros paquetes propios de Ubuntu y que no pueden ser instalados en Fedora y Centos han podido ser obviados gracias al soporte que brindan las otras herramientas y paquetes que se han podido instalar,

que son homólogas a las herramientas de Ubuntu y que son propias de Fedora y de Centos.

Una vez establecidos los paquetes que pueden reemplazar a los originalmente necesarios, es imprescindible abordar el problema de las versiones de los compiladores y del resto de paquetes. Mediante pruebas se ha podido determinar que Fedora 8, Fedora 9, Centos 5.1, 5.2, 5.3, 5.4 y 5.5 cuentan con la versión correcta del compilador GCC. Para el sistema operativo Centos se ha podido determinar que la versión 5.5 es la versión apropiada para instalar las herramientas correctamente, ya que las otras versiones de este sistema operativo no cuentan con la versión obligatoria del software python.

Ya resuelto este problema, y realizadas las pruebas con las versiones correctas de las dependencias y de los paquetes se ha podido concluir que es posible la instalación de todo el bloque de herramientas de RTAI, Comedi y Scilab para sistemas operativos Centos, utilizando exclusivamente la versión 5.5. En caso de cambiar una de las versiones de los paquetes o la versión de Centos, la instalación no puede ser realizada correctamente las herramientas no funcionan.

Respecto al caso Fedora se presentan discrepancias en las versiones de las dependencias Automake y Autoconf, de acuerdo a los resultados de las pruebas obtenidas se ha determinado que la única versión de Fedora que cuenta con las versiones requeridas para la instalación de cada uno de los paquetes y dependencias es la versión 8.

Luego de realizar pruebas se ha podido determinar que Fedora 8 soporta la instalación de todas las herramientas anteriormente mencionadas y al igual que los otros dos sistemas operativos la funcionalidad del mismo depende de la instalación de las versiones correctas indicadas en el Apéndice B.

Las versiones correctas de los paquetes de las dependencias previas para Ubuntu, Fedora y Centos, son las que se instalan por defecto con las herramientas de cada sistema operativo, en las versiones Ubuntu 8.4.3, Fedora 8 y Centos 5.5 respectivamente.

Por lo que se ha concluido, que respetando las versiones indicadas en este documento, es viable la instalación en los tres sistemas operativos y que el sistema operativo que más ventajas y soporte brinda para la instalación es Ubuntu.

2.1.2. Parámetros de configuración del Kernel.

Cuando iniciamos nuestro sistema Linux, lo que hacemos es cargar el núcleo con algunos parámetros previamente definidos. Sin embargo, si dichos parámetros no se ajustan completamente a nuestras necesidades, agregamos algunas opciones más para la ejecución del sistema que se enfoque más a nuestros requerimientos.

Antes de comenzar, es necesario tomar en cuenta muchas de las características propias del kernel. Estas características pueden ser compiladas dentro del mismo o como módulos. La diferencia radica

principalmente en que, mientras más modularizado sea nuestro núcleo, más pequeño será el tamaño de su imagen, lo que nos lleva a una mejor utilización de la memoria. Por otro lado, la utilización de módulos tiene otra gran ventaja [16]: un núcleo dinámico hace más fácil la tarea de agregar o eliminar hardware, puesto que solamente es necesario compilar e instalar o eliminar el módulo correspondiente, respectivamente, y no reconfigurar todas las opciones del sistema y compilar un nuevo núcleo. Además, los módulos son cargados en memoria y eliminados de ella en demanda, haciendo nuevamente un manejo más eficiente de los recursos.

Por último, cabe decir que hay quienes ven en la existencia de módulos un gran riesgo [17] de seguridad: un módulo malicioso puede ser capaz de esconder procesos, usuarios, etc. Por supuesto, los módulos requieren de privilegios de administrador para ser instalados, por lo que el ordenador debe tener un compromiso de seguridad previo para que esto suceda.

Existen varias ventajas por lo que se compila un nuevo núcleo, entre las principales tenemos:

- Administración de la memoria del computador.
- Establecimiento de la comunicación entre aplicaciones y dispositivos de hardware.
- Administración de procesos (o tareas).
- Establecimiento y uso de Sistemas en Tiempo Real.

Es muy importante verificar que tenemos las fuentes completas del nuevo núcleo a compilar, el cual se debe encontrar en el directorio `/usr/src`. También se recomienda hacer un enlace simbólico de la fuente del kernel que han descargado.

Como se mencionó con anterioridad el núcleo de GNU/Linux es modular y para cada módulo se puede especificar si se quiere que este compilado estáticamente, dinámicamente o que no se incluya en el mismo. Esta tarea de elección manual puede muchas veces a ser un poco tediosa, por lo que es indispensable generar un fichero de configuración. Además a la hora de tratar este fichero es posible hacerlo a través de una de las herramientas que se detalla a continuación.

Make oldconfig

Es la herramienta más simple en modo texto y la menos usada por el usuario final, consiste en responder con un carácter para cada uno de los módulos. Las opciones permitidas son tres:

[Y]: Permite compilar estáticamente los módulos a elegir.

[N]: No compilar los módulos.

[M]: Compilar como módulo de carga dinámica.

La Figura 2.3 muestra el resultado del comando `make oldconfig`.

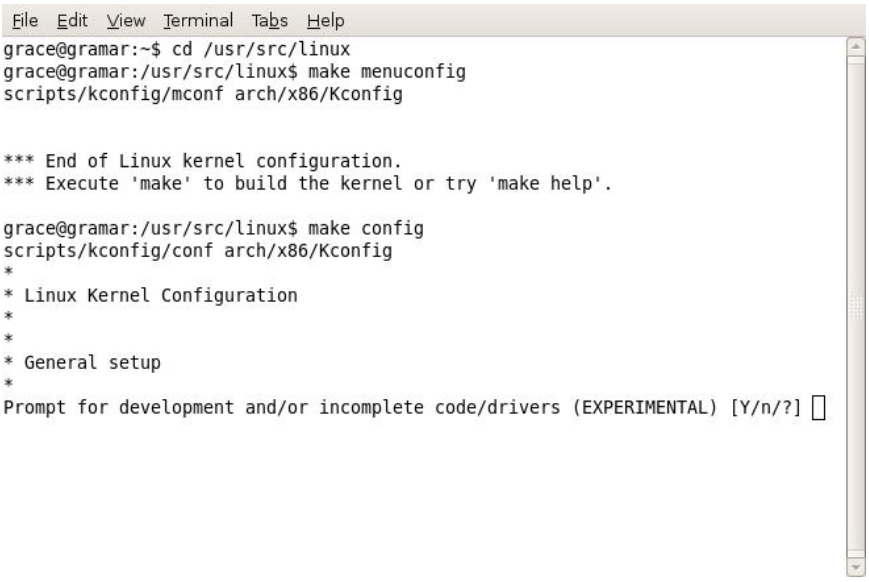


Figura 2.3 Ventana de Configuración de Make Oldconfig.

Make menuconfig

Es la herramienta más usada de modo texto y amigable para el usuario que configura el núcleo.

Es muy cómodo hacer la configuración por este método porque posee un menú muy sencillo e interactivo. Utiliza la librería ncurses para simular un entorno de ventanas que nos permite visualizar y recorrer las diferentes listas anidadas utilizando los cursores del teclado.

En la Figura 2.4 se muestra el resultado del comando `make menuconfig`

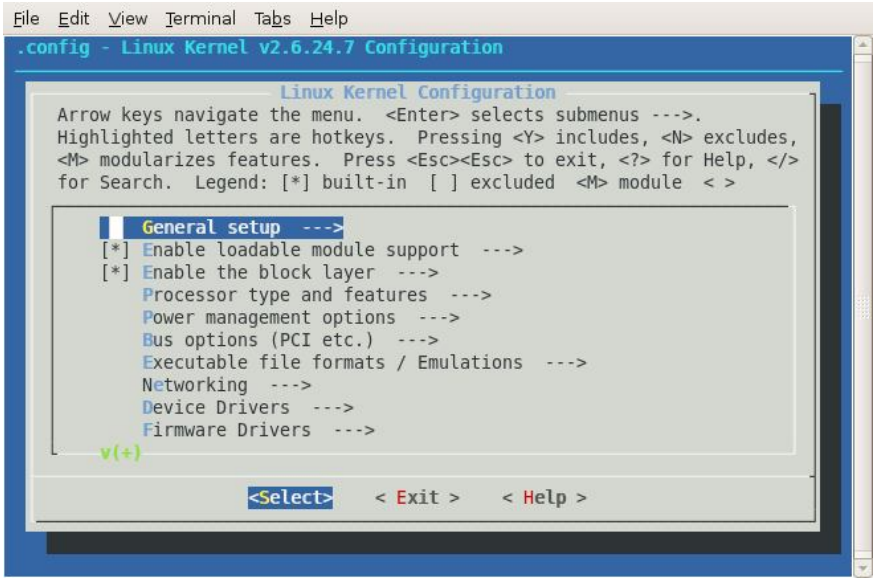


Figura 2.4 Ventana de Configuración de Make Menuconfig.

Make xconfig

Es una herramienta en modo gráfico que requiere Servidor X y de la biblioteca QT para su funcionamiento. Se usa principalmente en el entorno KDE, aunque no es obligatorio. Su interfaz, al ser gráfica, es la más agradable que las de modo texto y usa menús con ventanas y manejo de ratón. La Figura 2.5 se muestra el resultado del comando `make Xconfig`.

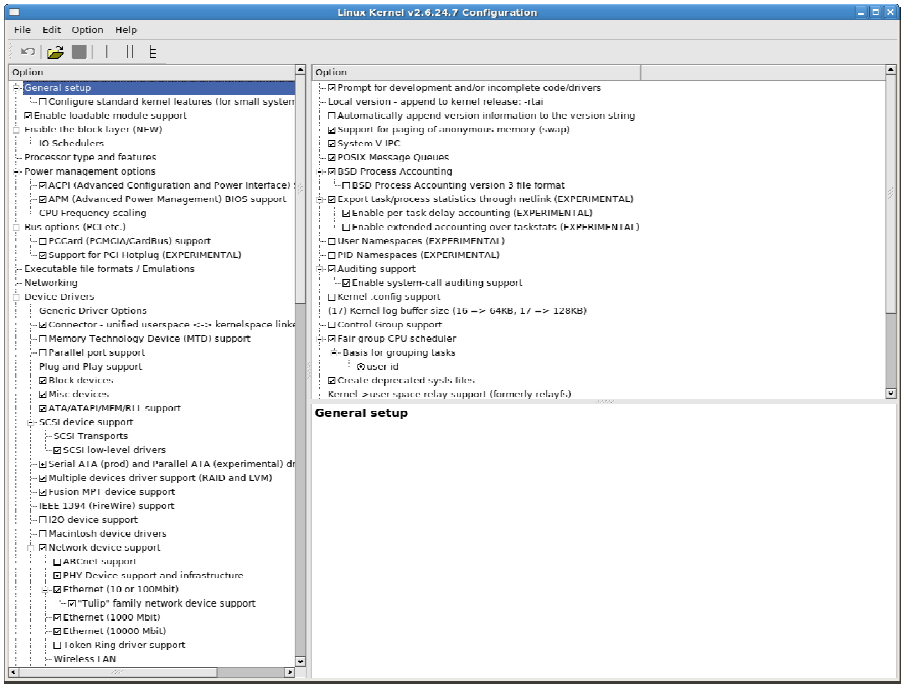


Figura 2.5 Ventana de Configuración de Make Xconfig.

Una vez que se ejecuta la herramienta de compilación seleccionada se muestra una lista de opciones que permiten crear un núcleo a medida.

Inicialmente, el núcleo que se distribuye libremente viene con una configuración genérica por defecto para cada arquitectura, por lo que configurarlo para nuestra arquitectura puede llegar a incrementar mucho el rendimiento.

El núcleo de Linux ofrece la funcionalidad no solo de deshacerse de las opciones que no van a hacer útiles en el sistema, sino que nos permite cargar aquellas funciones que nos interesen mediante los tipos de compilación.

Tipos de compilación

En ocasiones es deseable desarrollar un núcleo especifico para un dispositivo de limitadas características, en estos casos, lo habitual es compilar un núcleo monolítico reducido, es decir un nuevo núcleo en el que solamente se da soporte a aquellos elementos necesarios para el funcionamiento de dicho dispositivo y además todo el soporte reside en el propio núcleo, para la realización de tareas en tiempo real.

Linux permite también ejecutarse en sistemas de tiempo real, donde la carga del núcleo debe ser mínima para que pueda atender todas las peticiones entrantes en tiempo critico. Para estos casos, es interesante que el núcleo mantenga en su espacio de memoria el soporte a aquellos elementos que reciban información del exterior. Sin embargo, puede que este tipo de sistemas estén instalados en un PC donde sea necesario operar mediante la introducción de una tarjeta digital de acceso. No tiene mucho sentido habilitar el soporte del núcleo a este tipo de dispositivos para que resida permanentemente en la memoria ya que se está intentando que el núcleo sea el más ligero posible.

Para ordenadores donde el tiempo de inicio sea vital es apropiado que se cargue la menor cantidad de módulos al inicio del sistema pero también requiere que el sistema operativo de soporte a una gran cantidad de hardware.

Existen dos modos de compilación del núcleo, los cuales son:

Compilación Estática

El módulo seleccionado se compila junto al núcleo. Cada vez que el núcleo se cargue en memoria los módulos se cargan también.

Compilación Modular

El módulo seleccionado se compila de forma separada del núcleo, solamente cuando sea necesario, se cargará el módulo en memoria.

Antes de comenzar con la configuración de los drivers que componen el núcleo sería conveniente conocer qué tipos de dispositivos y driver están asociados a la PC; para ello tenemos la posibilidad de ejecutar dos comandos muy útiles:

`lspci` --> Lista todos los dispositivos pci.

A continuación detallamos los principales parámetros de configuración para el levantamiento de un nuevo Kernel, que permita transformar un sistema operativo de uso general en un sistema operativo en tiempo real.

GENERAL SETUP

Este grupo recoge las opciones más comunes de la configuración del propio núcleo.

- **Local version - append to kernel release**

  Añade un texto al final de la versión del núcleo de forma local. Útil para diferenciar distintas implementaciones de un mismo kernel. Este parámetro se lo utiliza para referenciar o indicar que se trata de un kernel

adaptado, de preferencia se coloca un sufijo representativo del parche, en nuestro caso -rtai; lo que permitirá añadir al final la palabra rtai a la versión del nuevo núcleo a compilar, como se muestra a continuación: 2.6.24.7-rtai.

- **Automatically append version information to the version string**

  Habilita la opción anterior.

- **Support for paging of anonymous memory(swap)**

  Soporte para dispositivos o archivos de intercambio

LOADABLE MODULE SUPPORT

Activación del soporte de módulos y opciones de los mismos.

- **Enable loadable module support**

  Habilita la capacidad del núcleo para cargar módulos dinámicamente, permite la activación del soporte de módulos y opciones de los mismos. Y si se activa esta opción, manejadores, código ejecutable o sistemas de ficheros pueden compilarse como módulos. Además permite añadirlos o quitarlos dinámicamente del núcleo. Se consigue el ahorro de memoria.

- **Set information alla modules simbols**

  Lo usual es que los módulos se compilen con el núcleo actual. Si se activa esta opción, es posible que los módulos compilados en otra versión del núcleo funcionen con la nueva versión del núcleo.

- **Kernel loader module**

  El administrador puede cargar (insmod) o quitar (rmmod) módulos del núcleo. Con esta opción activada el núcleo (modprobe) decide cuando cargar o descargar un módulo en función de sus necesidades.

- **Module unloading**

  Permite la descarga de los módulos de memoria siempre que el módulo lo permita. Es útil para que el kernel ocupe lo menos posible en memoria.

- **Forced module unloading**

  Permite el uso de herramientas de manejo de módulos para forzar la descarga de alguno.

- **Module versioning support**

  Permite usar módulos compilados para otras versiones del kernel siempre que se incluya suficiente información en estos para conocer las limitaciones. Esta opción debe estar deshabilitada para la Instalación de RTAI.

- **Source checksum for all modules**

  Añade un nuevo campo a las estructuras de descripción del modulo donde se codifica la fuente utilizada para construir el modulo. Útil solamente en caso de que se usen varias versiones distintas del mismo modulo.

- **Automatic kernel module loading**

  Permite que el kernel cargue módulos a medida que los necesite.

PROCESSOR TYPE AND FEATURES

Soporte para distintos tipos de procesadores y características especificas de los mismos.

- **Symmetric multi-processing support**

  Habilita el soporte para ordenadores con más de un núcleo (Hyper-Threading, Core Duo, QuadCore). Esta opción puede ser habilitada o deshabilitada de acuerdo a las características propias del procesador del computador en el cual se va a instalar el nuevo núcleo. Hacer uso del siguiente comando: `lspci`

- **Subarchitecture Type**

  Permite seleccionar la arquitectura que más se adapta a la nuestra. Habitualmente se selecciona siempre PC-compatible.

- **Processor family**

  Aquí se recogen todas las posibles configuraciones de la arquitectura del procesador para las que se puede configurar el kernel x386(compatible con todos los pcs), 486,586(procesadores Pentium y similares). Hacer uso del comando para verificar el modelo del procesador: `cat /proc/cpuinfo`

- **Generic x86 support**

  Implementa optimizaciones genéricas para todos los procesadores basados en la arquitectura x86 independientemente de la opción seleccionada en el parámetro anterior.

- **SMT(Hyperthreading) scheduler support**

  Mejora el planificador del CPU en máquinas P4 con Hyper threading.

- **Multi-core scheduler support**

  Mejora el planificador de la CPU para máquinas Multi-Core .

- **Preemption model**

  Permite seleccionar el modelo de manejo de aplicaciones por parte del procesador. En general, los equipos configurados como servidores deben utilizar la opción `'No forced preemption'`, donde los procesos tienen mayor cantidad de tiempo antes del cambio de contexto mientras mayor sea su prioridad, pero en el caso de un sistema en tiempo real seleccionaremos `Preemptible Kernel (Low-Latency Desktop)` para que el parche aplicado (Rtai) presente la menor latencia posible en el procesamiento de las tareas en el sistema.

- **Interrupt pipeline**

  También se la conoce como I-pipe. Esta es basada en una propuesta técnica para un sistema llamado ADEOS, i-pipe es una implementación simplificada de una propuesta que se basa en una modificación de las fuentes del núcleo de Linux para la virtualización de la máscara de interrupción, en lugar de depender sobre las características del chip x86. Este parámetro debe ser activado en la configuración del núcleo caso contrario en surgirá un error el momento de la compilación.

- **Local APIC support on uniprocessors**

Permite el uso de APIC por parte del procesador, si el procesador tiene PIC solamente, esta opción no influye en el rendimiento del sistema.

IO-APIC Support on uniprocessor

Habilita el uso de IO-APIC presente en muchos sistemas SMP como sustituto del APIC habitual.

- **Toshiba laptop support / Dell laptop support**

  Habilita el soporte para los modos especiales del procesador que permiten usar ambos equipos portátiles, siempre que la BIOS sea Toshiba o Dell respectivamente, su utilidad está restringida a este tipo de equipos e incluso puede generar errores en aquellos equipos Toshiba o Dell que no soporten las características para las que fueron programados estos módulos.

- **High memory support**

  Permite seleccionar que soporte debe incluirse en el núcleo para grandes cantidades de memoria.

POWER MANAGEMENT OPTIONS

Soporte para el manejo de alimentación, útil cuando se utilizan sistemas con batería como los portátiles; ahorra energía en períodos de inactividad desactivando dispositivos. Se realiza activando APM O ACPI.

- **ACPI (Advanced Configuration and Power Interface) Support**

  ACPI permite el control de la administración de energía desde el sistema operativo. El estándar del anterior para la administración de energía, administración avanzada de energía (APM), se controla a nivel del BIOS. APM se activa cuando el sistema se vuelve inactivo - el más largo es el sistema de ralentí, la energía que consume menos (por ejemplo, protector de pantalla frente a frente a suspender el sueño). En APM, el sistema operativo no tiene conocimiento de cuando el sistema va a cambiar estados de energía.

- **APM (Advanced Power Management) BIOS Support**

  Administración avanzada de energía (APM) permite que el BIOS del equipo de soporte a la gestión de ahorro energía en los períodos de inactividad del computador apagando el monitor, parando los discos o suspendiendo el sistema.

  Si se está depurando el núcleo o en sistemas SMP debe desactivarse esta opción, el fichero `/proc/apm` proporciona información del estado de las pilas.

- **CPU Frequency scaling**

  La forma más fácil de ahorrar energía es con la escala de frecuencia de la CPU. Esta es una tecnología que le permite ajustar la velocidad de una CPU corriendo procesos, mientras se está ejecutando. Cuando la CPU

se ejecuta a velocidades más lentas, que consumen menos energía. La mayoría de las CPUs se ponen a su velocidad máxima todo el tiempo, incluso cuando el sistema no los está usando. Linux tiene soporte para mantener la CPU a la velocidad máxima a menos que esté inactivo. Mediante la activación de esta característica, se puede ahorrar energía, virtualmente sin costo para el rendimiento. La característica de Linux para manejar la escala de frecuencia de CPU se llama cpufreq.

BUS OPTIONS

Este grupo de opciones recoge todo el soporte para los buses PCI, PCMCIA, ISA EISA Y MCA. El soporte para ISA y EISA se puede desactivar para las maquinas más modernas (Pentium 4 en adelante) y para portátiles obtenidos a partir del 2005. Además los ordenadores de sobremesa no suelen tener puertos PCMCIA/CardBus por lo que se puede desactivar completamente.

NETWORKING

Agrupa todas las opciones de conectividad del sistema mediante el soporte del núcleo. Estas opciones controlan la capacidad TCP/IP del ordenador así como la implementación del protocolo IPv6, el soporte para infrarrojos, bluetooth y las opciones para conexiones wifi.
En general las opciones pueden ser seleccionadas en función de la configuración hardware del equipo y acorde a las necesidades del usuario,

por lo que se recomienda desactivar el soporte para ***Bluetooh, Amateur Radio Support e Irda*** infrarrojos ya que consumen memoria y tiempo tanto en el momento de la compilación como en la inicialización del sistema, mientras menor sea la carga de módulos innecesarios mejor será el rendimiento.

DEVICES DRIVERS

Esta sección recoge las opciones de soporte de dispositivos que se pueden conectar al equipo y que serán reconocidos por el núcleo. Los parámetros más relevantes son:

- **Parallel port support:** soporte de puerto paralelo

  Se basa en el modulo partport que da soporte a dispositivos conectados al puerto paralelo de 25 pins, como impresoras, unidades zip.

- **I2C Support**

  El I2C es un protocolo de bus serial lento, usado por muchos micro controladores, es habitual encontrarlo en sensores de hardware como sensores de temperatura o movimiento. La activación o no de este soporte viene determinado por el tipo de máquina sobre la que se vaya a desplegar el núcleo.

- **Multimedia devices**

  Aquí encontramos las opciones referentes al soporte de video, estos dispositivos son conectados a través del video4linux y video4linux2, se sugiere desactivar las demás opciones.

- **Sound**

  ALSA (Advanced Linux Sound Architecture) es el actual sistema de sonido para el núcleo de Linux. Se sugiere desactivar el sonido para la instalación de RTAI.

Si desea determinar qué tipo de controlador de sonido está presente en su máquina, y de qué tipo es, ejecute el siguiente comando: `/usr/sbin/lspci | grep -i audio`

- **Virtualización**

  Esta opción permite habilitar/deshabilitar los módulos que permiten al ambiente Linux la instalación de herramientas de virtualización. La Desactivación de este parámetro permite que el tiempo de compilación en el levantamiento del nuevo núcleo disminuya notablemente especialmente si hablamos de un ordenador de doble núcleo, ya que este opción posee varios módulos que emplean tiempo más o menos significante en el procesamiento.

- **File Systems**

  Se encuentran todas las opciones del sistema de ficheros. Cabe destacar las siguientes opciones:

  - Soporte para ext2, el sistema de ficheros nativo de Linux.

  - Soporte para ext3, que es la extensión al sistema nativo del Linux ext2 con journalling, si su sistema raíz es /ext3 no puede montarse como modulo con la utilidad tune2fs se puede pasar de ext2 a ext3 y viceversa.

- Soporte para establecer cuotas de disco soso se aplica al sistema de ficheros ext2 y ext3

- Soporte para uto FS montar sistemas de ficheros remotos bajo demanda, existe una nueva versión v4

- Soporte para el sistema de ficheros reiserfs sistema de ficheros que utiliza un árbol balanceado y utiliza journaling

- Soporte para sistema de fichero MSDOS-FAT y como M permiten montar el sistema de ficheros FAT y poder ser accedidos desde Linux en lectura y escritura, permite ejecutar Linux dentro de una partición MSDOS.

- **CD-ROM/DVD FILESYSTEMS**

  Soporte para sistema de ficheros ISO 9600 CD-ROM. Permite almacenar música y datos, además brinda soporte para Joliet la extensión de Microsoft del ISO 9660 cdrom que permite almacenar caracteres con formato de 16 bits Unicode.

  Soporte para rockridge extensión transparente de Linux que permite descompresión de los datos almacenados en CD-ROM. Soporte para el sistema de ficheros UFS, utilizado por sunos, FreeBSD, NetBSD, OpenBSD, NextTstep.

- **DOS/FAT/NT FilesYSTEMS**

  Soporte para distintos sistemas de archivo, se debe dejar las opciones que vienen por defecto en la configuración.

2.1.3. Parámetros de configuración en Herramientas instaladas.

Mesa3D

Es un sistema Open Source de procesamiento interactivo de gráficos 3D implementado con soporte de OPENGL la cual es una librería gráfica Open Source que abstrae un conjunto de instrucciones y brinda un conjunto de funciones y procedimientos para la creación de aplicaciones que permitan generar gráficos en 2 y en 3 dimensiones.

Parámetros configurados en sección de Instalación de Mesalib

Parámetros principales

```
make realclean
```

Esta línea de comando se la usa cada vez que se desea asegurarse de que no existan datos de una compilación anterior, que puedan alterar las compilaciones que se vayan a especificar durante esta instalación.

```
make linux-x86-static
```

Esta línea de código le indica al comando make, que se realice una compilación independiente, es decir sin vinculaciones con método de aceleración de gráficos, es decir un método stand-alone, que produzca la configuración necesario para un sistema operativo proveniente de Linux, para procesadores tipo 386, 486, 586,686, etcétera, y produciendo librerías de forma estática como libGL.so

```
make install
```

Instalará la conflagración previamente indicada.

Parámetros Comunes para levantar un archivo de configuración de las compilaciones de EFLTK, ComediLib, Comedi y Scilab

`--prefix=PREFIX`

Especifica el prefijo del directorio de instalación. Es decir las siguientes variables son construidas a partir de esta especificación. Todas las otras carpetas con las que vayan a interactuar los archivos importantes de los programas, tendrán como directorio raíz, el directorio especificado por la variable prefix. Es un estándar de GNU que todo paquete de instalación tenga por defecto la variable prefix la siguiente ruta: /usr/local/, directorio que por defecto, contiene datos que pertenecen a los programas instalados en la raíz del equipo local. Ningún paquete o subprograma puede ser instalado directamente en este directorio, de utilizarse los subdirectorios que se encuentran en su interior.

`--exec_prefix=EPREFIX`

Es el prefijo de la raíz donde el programa va a instalar los paquetes compilados, posee el mismo valor de prefix. En la compilación es exactamente lo mismo que prefix, ya que exec_prefix es una copia de prefix, por lo que se le es designado automáticamente el valor de: EPREFIX=PREFIX.

`--bindir=DIR`

Especifica el directorio donde se encuentran los ejecutables de los programas que el usuario puede ejecutar, durante una instalación este directorio es automáticamente asignado, en caso de que no se especifique esta instrucción, como: [EPREFIX/bin].

`--sbindir=DIR`

Especifica el directorio donde se encuentran los ejecutables de los programas que se pueden ejecutar desde el terminal pero que solo son generalmente útiles para los usuarios administradores del sistema (super-usuarios), durante una instalación este directorio es automáticamente asignado, en caso de que que no se especifique esta instrucción, como:[EPREFIX/sbin].

`--libexecdir=DIR`

Especifica el directorio en el cual se instalaran los programas que van a ser ejecutados por otros programas en lugar de ser ejecutados por el usuario, durante una instalación este directorio es automáticamente asignado, en caso de que no se especifique esta instrucción, como: [EPREFIX/libexec]

`--libdir=DIR`

Es el directorio donde se instalan los archivos de objetos, es decir archivos que contienen instrucciones en lenguaje de maquina e información de tal forma que mediante un vínculo se puedan crear programas ejecutables a partir de los mismos, y librerías con el código de los objetos, los códigos de objetos contienen una serie de instrucciones que pueden ser entendidas por el procesador y que son muy difíciles de leer o modificar por personas, durante

una instalación este directorio es automáticamente asignado, en caso de que no se especifique esta instrucción, como: [EPREFIX/lib]

Parámetros configurados en sección de Instalación de EFLTK

Parámetros principales

```
./configure --disable-mysql --disable-unixODBC
```

La instrucción. /configure, nos da a entender los siguiente: dentro del directorio actual (lo cual es indicado por. /) Existe un script llamado configure, que debido a su nombre se trata de un script, o texto que contiene comandos de shell, encargado de realizar la configuración de la herramienta antes de ser compilada. Todos los elementos escritos posteriormente pueden ser catalogados como banderas, o instrucciones que le da la persona al script para que realice la configuración.

Usualmente todo Script de configuración suele necesitar ser ejecutado como super-usuario (root).

```
--disable-mysql
```

No producir soporte para MySQL. Para su correcto funcionamiento EFLTK no requiere de este soporte.

```
--disable-unixODBC
```

No producir soporte para unixODBC. Para su correcto funcionamiento EFLT no

requiere de este soporte.

OTROS PARAMETROS CONFIGURABLES

`--x-includes=DIR`

Si se usa X especifica el directorio donde se encuentran los archivos incluidos de X

`--x-libraries=DIR`

Si se usa X especifica el directorio donde se encuentran los archivos de las librerías de X

`--enable-static`

Crea librerías estáticas

`--enable-opengl`

Activa el soporte para OpenGL, activado por defecto

`--enable-plugins`

Habilita soporte para plug-ins de FLTK

`--disable-utf8`

Deshabilita soporte para UTF8

`--with-x`

Especifica el uso de sistemas de ventanas X

Parámetros configurados en sección de Instalación de COMEDILIB

Parámetros principales

`./configure --sysconfdir=/etc`

--sysconfdir=/etc

Instalara archivos de solo lectura para la maquina en el directorio indicado, y que pertenecen a una sola máquina, como por ejemplo archivos de configuración del host en el que se está instalando el software. El directorio /etc. es el directorio por defecto

OTROS PARAMETROS CONFIGURABLES

`--enable-shared`

Crea librerías compartidas, por defecto activado

`--enable-static`

Crea librerías estáticas, por defecto activado

`--enable-fast-install`

Optimiza la instalación para que sea más rápido, por defecto activado

`--disable-libtool-lock`

Deshabilita compilaciones paralelas incrementando la velocidad de compilación

`--enable-maintainer-mode`

Crea reglas independencias no útiles para el instalador

`--disable-python-binding`

Deshabilita el levantamiento de vinculación con Python

`--disable-ruby-binding`

Deshabilita el levantamiento de vinculación con la librería Ruby

`--enable-etc-hotplug`

Habilita hotplug en /etc./hotplug, es decir la capacidad para habilitar o deshabilitar un dispositivo sin tener que apagar el computador

Parámetros configurados en sección de Instalación de RTAI primera parte

instrucción: `make menuconfig`

General ->Installation directory: /usr/realtime

Carpeta donde se desea instalar la aplicación de RTAI.

General -> Linux Source Tree: /usr/src/linux

Ubicación del código fuente del kernel parchado con RTAI

Parámetros configurados en sección de Instalación de COMEDI

Parámetros principales

```
. /configure --with-linuxdir=/usr/src/linux --with-
rtaidir=/usr/realtime --enable-kbuild --disable-pcmcia
```

--with-linuxdir

Indica la ruta de la carpeta donde se encuentra la fuente del kernel en el cual se ha parchado Rtai.

--with-rtaidir

Indica el directorio donde se encuentra el código fuente de rtai

--enable-kbuild

Forza el uso del sistema kbuild del kernel para levantar módulos.

`--disable-pcmcia`

Deshabilita el soporte para tarjetas tipo PCMCIA, diferente al tipo de tarjeta PCI que se ha utilizados para las pruebas para la redacción de este documento.

OTROS PARAMETROS CONFIGURABLES

`--disable-rtai`

Deshabilita el uso de RTAI

`--disable-usb`

Deshabilita el soporte para trabajo con dispositivos USB

`--enable-kbuild`

Forza o no el uso del sistema de levantamiento de módulos del kernel kbuild

`--with-kernel-release=kernel`

Especifica el nombre del kernel para el cual se va a hacer la compilación

`--with-machine=MACHINE`

Especifica el nombre de la máquina para la cual se va a hacer la compilación

`--with-linuxdir=DIR`

Especifica la ruta del directorio fuente del kernel Linux

`--with-linuxconfig=FILE`

Especifica la ruta del archivo de configuración del kernel Linux

`--with-modulesdir=DIR`

Especifica la ruta del directorio donde se encuentran instalados los módulos del kernel

`--with-rtaidir=DIR`

Especifica la ruta del directorio de instalación de RTAI

`--with-rtlinuxdir=DIR`

Especifica la ruta del directorio fuente de RTLinux, otro software de Tiempo Real

`comedi_config -v /dev/comedi0 ni_pcimio`

Comedi_config asigna el siguiente orden a las instrucciones escritas después de comedi_config : `<opciones> <dispositivos> <drivers> <parámetros>`

Imprime en pantalla los mensajes que retorna el comando `comedi_config`

`/dev/comedi0`

Vincula el hardware(tarjeta de adquisición de datos) instalado con el dispositivo comedi0 de la carpeta /dev

`ni_pcimio`

Se configurara cualquier tarjeta de National Instruments de la serie E tipo PCI-MIO, Por ejemplo las siguientes :

National Instruments PCI-6023E PCI-6024E PCI-6025E PCI-6031E PCI-6032E PCI-6033E PCI-6034E PCI-6035E PCI-6036E PCI-6040E PCI-6052E PCI-6071E PCI-6110 PCI-6111

Parámetros configurados en sección de Instalación de RTAI segunda parte

instrucción: `make menuconfig`

`Add-ons-> Real Time COMEDI support in user space: habilitado`

`Habilita a RTAI para trabajar con soporte para COMEDI`

Parámetros configurados en sección de Instalación de Scilab

`./configure --without-java --with-tcl-library=/usr/lib --`

`with-tcl-include=/usr/include/tcl8.4`

`--without-java`

No se compila con interfaz Java

`--with-tcl-library=/usr/lib`

Ruta donde se encuentra los archivos de la libreria tal

`--with-tcl-include=/usr/include/tcl8.4`

Ruta donde se encuentra los archivos incluidos de tal

OTROS PARAMETROS CONFIGURABLES

`--x-includes=DIR`

Si se usa X especifica el directorio donde se encuentran los archivos incluidos de X

`--x-libraries=DIR`

Si se usa X especifica el directorio donde se encuentran los archivos de las librerías de X

`--enable-shared[=PKGS]`

Crea librerías compartidas, por defecto activado

`--enable-static[=PKGS]`

Crea librerías estáticas, por defecto activado

`--enable-fast-install[=PKGS]`

Optimiza la instalación para que sea más rápido, por defecto activado

`--disable-libtool-lock`

Deshabilita compilaciones paralelas incrementando la velocidad de compilación

`--with-x`

Especifica el uso de sistemas de ventanas X

`--with-gcc`

Utilizar para la compilación el compilador de C, gcc

`--with-g77`

Utilizar para la compilación el compilador de Fortran 77, g77

`--with-gfortran`

Utilizar para la compilación el compilador de Fortran 95, gfortran

`--with-gnu`

Utilizar compilar gcc para C y utilizar o g77 o gfortran para Fortran

--without-pvm

No compilar con librería PVM

--without-tk

No compilar con TCL,TK

--with-intelcompilers

Utilizar compiladores de Intel icc e ifortran

--with-local-xaw

Utilizar objetis Xaw3d widgets dados junto a Scilab

--without-pvm

No compilar con librería PVM

--without-tk

No compilar con TK

--with-gtk2, --with-gtk

Usa Gtk, para objeto como ventanas, provenientes de la plataforma GTK para desarrollar interfaces graficas de usuario.

`--without-ocaml`

No compilar con ocaml, lenguaje de programación de la familia de los lenguajes META (ML)

`--with-pvm-include=DIR`

Especifica el directorio donde se encuentran los archivos incluidos de PVM

`--with-pvm-library=DIR`

Especifica el directorio donde se encuentran los archivos de las librerías de PVM

`--with-tk-library=DIR`

Especifica el directorio donde se encuentran los archivos de las librerías de TK

`--with-tk-include=DIR`

Especifica el directorio donde se encuentran los archivos incluidos de TK

2.2. Descripción del Proceso de Instalación

Los paquetes que requieren instalación desde los archivos comprimidos fuente, y no requieren configuración por parte del usuario, por medio de menús de configuración, poseen el mismo proceso de instalación tanto para Ubuntu como para Fedora y Centos. Es recalcable que toda aplicación en tiempo real que se desee ejecutar, requiere que se ingrese al kernel parchado durante el proceso de instalación e ingresar como super-usuario. Otros tipos de paquetes que se encuentran en el

proceso de instalación son los paquetes de las dependencias previas, los cuales varían entre Ubuntu y Fedora, pero entre Fedora y Centos son esencialmente los mismos. Y el tipo restante de paquetes que encontramos durante el proceso de instalación son los paquetes cuya instalación se hace con los archivos fuente y que requieren de configuración del usuario, la configuración en el caso de Ubuntu, Fedora y Cantos es la misma, la única diferencia es el método de visualización del panel de configuración de las herramientas y el comando que se usa para llamarlos. En el caso de Ubuntu los panales de configuración se los llama mediante el comando: *`make menuconfig`*, en el caso de Fedora y Cantos el comando a utilizar es: *`make xconfig`*.

NOTA: Para referencias y descripción de cada paso de cada uno de los dos procesos de instalación refiérase a la sección de Apéndice A y Apéndice B

2.2.1. Instalación en Distribución Ubuntu

2.2.1.1. Instalación de Dependencias Previas

El proceso parte con la instalación de las dependencias necesarias para la compilación de las herramientas descritas en el gráfico, y los paquetes co-requisito para los mismos.

```
apt-get install subversion build-essential automake
checkinstall x11proto-xext-dev xlibs-static-dev
libxext-dev libxt-dev libncurses5-dev fakeroot
kernel-package swig python-dev libtool libboost-
program-options-dev libgsl0-dev libxmu-dev libxi-dev
```

```
bison doxygen g77 gfortran flex sablotron xaw3dg-dev
libpvm3 pvm-dev libgtkhtml2-dev libzvt-dev libvte-dev
tcl8.4 tk8.4 tcl8.5-dev tk8.5-dev libqt4-dev libqwt5-
qt4-dev qt3-dev-tools libdrm-dev xorg-dev cvs tcl8.4-
dev tk8.4-dev libglu1-mesa-dev
```

2.2.1.2. Primera Sección de Instalación

Como primera parte de la instalación de las herramientas señaladas en el diagrama de la sección de justificación, se encuentran MesaLib y EFLTK, sistema para crear aplicaciones interactivas de gráficos 3D y librería para crear aplicaciones de entorno gráfico, ambas paquetes son instalados ya que más adelante se procederá a instalar Xrtailab, herramienta de visualización, el cual requiere de estos dos paquete para poder ser compilado e instalado.

Instalación Mesa3D

```
cp  MesaLib-7.0.3.tar.gz /usr/local/src
cp rtai-3.6-cv.tar.bz2 /usr/src/
cp linux-2.6.24.7.tar.bz2 /usr/src/
cd /usr/local/src
tar xvf MesaLib-7.0.3.tar.gz
cd Mesa-7.0.3
make realclean
make linux-x86-static
make install
```

Instalación de EFLTK

```
cd /usr/local/src
svn co
https://ede.svn.sourceforge.net/svnroot/ede/trunk/efl
tk
svn co
https://ede.svn.sourceforge.net/svnroot/ede/trunk/ede
cd efltk
autoconf
./configure --disable-mysql --disable-unixODBC
./emake
./emake install
echo "/usr/local/lib" >> /etc/ld.so.conf
/sbin/ldconfig
```

Segunda Sección de Instalación

Luego de realizar los pasos del punto anterior se procede con el proceso de parchado del kernel. Ya que se requiere de un sistema de interacciones en tiempo real es necesario que el kernel sea alterado, colocarle módulos, e instalar un método para atención inmediata para interrupciones de procesador y hardware, por lo que es necesario

colocar una capa de software nuevo debajo del núcleo Linux con el control total de las interrupciones y las características principales del procesador, este software se encontrará en interacción directa con el hardware. Para poder llevar esto acabo es necesario colocar un kernel nuevo el cual se encuentre parchado para el funcionamiento con RTAI.

Parchado y Levantamiento de Kernel

```
cd /usr/src/
tar xjvf rtai-3.6-cv.tar.bz2
ln -s rtai-3.6-cv rtai
tar xjvf linux-2.6.24.7.tar.bz2
mv /usr/src/linux-2.6.24.7 /usr/src/linux-2.6.24.7-
rtai
ln -s linux-2.6.24.7-rtai linux
cd /usr/src/linux
patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-
linux-2.6.24-x86-2.0-07.patch
cp /boot/config-2.6.24-24-generic .config
make oldconfig
make menuconfig
```

Nótese que se está usando la instrucción `menuconfig` la cual solo funciona en Ubuntu, esta instrucción presenta una ventana de configuración, estructurada dentro de un terminal en base a caracteres

del código Ascii. En esta parte el usuario debe de adecuar los módulos tal como se indica en la sección de *Apéndices.*

```
make
make modules_install
make install
cp -aR /lib/firmware/2.6.24-24-generic
/lib/firmware/2.6.24.7-rtai
update-initramfs -c -k 2.6.24.7-rtai

update-grub
```

En este punto es necesario reiniciar la máquina e ingresar en el kernel con soporte para Rtai.

2.2.1.3. Tercera Sección de Instalación

El siguiente paso, la instalación de ComediLib, se la realiza para poder contar con una librería que permita manejar la tarjeta de adquisición de datos. La tarjeta de adquisición de datos será controlada por Comedi, dentro del kernel, pero el usuario tendrá acceso a ella por medio de ComediLib, y para configurar la tarjeta es necesario hacerlo por medio de ComediLib por lo que es necesario tenerlo instalado antes que se instale Comedi.

Instalación de Comedilib

```
cp comedilib-0.8.1.tar.gz /usr/local/src/
cp comedi-0.7.74.tar.gz /usr/local/src/
cd /usr/local/src
tar xzvf comedi-0.7.74.tar.gz
tar xzvf comedilib-0.8.1.tar.gz
mv comedi-0.7.74 comedi
mv comedilib-0.8.1 comedilib
cd comedilib
sh autogen.sh
./configure --sysconfdir=/etc
make
make install
make dev
```

2.2.1.4. Cuarta Sección de Instalación

Para poder instalar los siguientes paquetes, RTAI y Comedi, es necesario tomar en cuenta que para poder instalar cada paquete es necesario tener previamente instalado el otro lo cual, produce un lazo cíclico, por lo que no se podría instalar ninguno de los dos paquetes. Para corregir esto, se debe realizar el siguiente proceso, se debe instalar inicialmente Rtai sin soporte para Comedi, por lo que durante la instalación Rtai no buscara la instalación de Comedi, por lo que la instalación será exitosa, Luego se instala Comedi, el cual una vez que identifica que ya se encuentra instalado Rtai, se instalará

correctamente. Una vez instalado Comedi, se vuelve a instalar RTAI, esta vez con soporte para Comedi, y RtaiLab

```
cd /usr/src/rtai
```

```
make menuconfig
```

Nótese que se está usando la instrucción `menuconfig` la cual solo funciona en Ubuntu, esta instrucción presenta una ventana de configuración, estructurada dentro de un terminal en base a caracteres del código Ascii. En esta operación el usuario no debe alterar los valores por defecto.

```
make
make install
echo "PATH=$PATH:/usr/realtime/bin" >> /root/.bashrc
echo "export PATH" >> /root/.bashrc
```

En este punto es necesario reiniciar la maquina e ingresar en el kernel con soporte para Rtai.

Si desea comprobar la efectividad del proceso de instalación realizado hasta ahora, recomendamos realizar pruebas de latencia con los siguientes comandos.

Sección Comandos de Prueba de Instalación

```
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko

cd /usr/realtime/testsuite/kern/latency
./run

insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
insmod /usr/realtime/modules/rtai_lxrt.ko

cd /usr/src/rtai/testsuite/kern/latency/
insmod latency_rt.ko period=1000000
./display
rmmod latency_rt
```

En caso de realizar estas pruebas, para poder continuar con el proceso de instalación es necesario reiniciar la máquina e ingresar nuevamente en el kernel con soporte para Rtai.

Instalación de Comedi

```
cd /usr/local/src/comedi
sh autogen.sh
./configure --with-linuxdir=/usr/src/linux --with-
rtaidir=/usr/realtime --enable-kbuild --disable-
pcmcia
make
make install
depmod -a
make dev
cp include/linux/comedi.h include/linux/comedilib.h
/usr/include/
cp include/linux/comedi.h include/linux/comedilib.h
/usr/local/include/
ln -s /usr/include/comedi.h
/usr/include/linux/comedi.h
ln -s /usr/include/comedilib.h
/usr/include/linux/comedilib.h
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
insmod /usr/realtime/modules/rtai_sem.ko
insmod /usr/realtime/modules/rtai_mbx.ko
insmod /usr/realtime/modules/rtai_msg.ko
```

```
insmod /usr/realtime/modules/rtai_netrpc.ko
ThisNode="127.0.0.1"
insmod /usr/realtime/modules/rtai_shm.ko
insmod /usr/realtime/modules/rtai_signal.ko
insmod /usr/realtime/modules/rtai_tasklets.ko
modprobe comedi
modprobe kcomedilib
modprobe comedi_fc
modprobe 8255
modprobe ni_pcimio
comedi_config -v /dev/comedi0 ni_pcimio
```

En este momento se pueden insertar script de carga de módulos de Rtai en el listado de daemones que corren al arranque del sistema operativo. Los ejemplos de estos dos scripts y como cargarlos en los niveles de arranque se encuentran detallados en la sección de *Apéndices*

Luego de esto se requiere reiniciar la máquina e ingresar en el kernel con soporte para Rtai.

```
cd /usr/src/rtai
```

Nótese que se está usando la instrucción menuconfig la cual solo funciona en Ubuntu, esta instrucción presenta una ventana de configuración, estructurada dentro de un terminal en base a caracteres

del código Ascii. En esta operación el usuario debe seleccionar COMEDI support over LXRT y escribir `/usr` en Comedi installation directory. También debe seleccionar RTAI Lab, y escribir `/usr/local` en EFLTK installation directory

```
make menuconfig
make
make install
```

Se requiere reiniciar la máquina y se ingresa en el kernel con soporte para Rtai.

2.2.1.6. Quinta Sección de Instalación

Esta es la parte final de la instalación en la que se instala la herramienta con la que el usuario podrá crear los diagramas y los ejecutables de las aplicaciones para controlar señales eléctricas. Se realiza la instalación de Scilab y junto a él, se instala el módulo Scicos para crear los diagramas y luego se instala el parche para Scilab y Scicos para que dentro de los elementos que el usuario encontrará en Scicos, se encuentren los bloques de comunicación con la tarjeta y bloques de manejo de tiempo real de Rtai.

Instalación de Scilab

```
cp scilab-4.1.2-src.tar.gz /usr/local/
cd /usr/local/
tar xzvf scilab-4.1.2-src.tar.gz
cd /usr/local/scilab-4.1.2/
./configure --without-java --with-tcl-
library=/usr/lib --with-tal-
include=/usr/include/tcl8.4
make all
ln -s /usr/local/scilab-4.1.2/bin/scilab
/usr/local/bin/scilab
```

Instalación de Addons Rtai-Scilab

```
cd /usr/src/rtai/rtai-lab/scilab/macros
make install
make user
```

Desde este momento el usuario puede comenzar a trabajar creando aplicaciones de control de señales adquiridas por la tarjeta en tiempo real.

2.2.2. Instalación en Distribución Fedora y Centos.

2.2.2.1. Instalación de Dependencias Previas

El proceso parte con la instalación de las dependencias necesarias para la compilación de las herramientas descritas en el gráfico, y los paquetes co-requisito para los mismos.

```
yum install cvs subversion xorg-x11-proto-devel libXext-devel libXt-devel intltool python-devel compat-gcc-34-g77 boost gsl-devel libXmu-devel libXi-devel Xaw3d-devel pvm gtkhtml2-devel tcl-devel tk-devel qt-devel flex python

yum groupinstall "Development Tools"
```

2.2.2.2. Primera Sección de Instalación

Esta sección de instalación está constituida por herramientas que son instaladas desde su código fuente y que no requieren configuración por parte del usuario por medio de una panel de configuración, por lo tanto su configuración, compilación, e instalación son similares al proceso de realizado en la sección de instalación para Ubuntu. Para realizar esta parte de la instalación seguir los pasos indicados en la sección *2.2.1.2 Primera Sección de Instalación.*

2.2.2.3. Segunda Sección de Instalación

Esta sección de instalación tiene la misma finalidad que el proceso de instalación que se realiza en la sección *2.2.1.3 Segunda Sección de Instalación*. El cambio que puede ser visualizado por el usuario y que diferencia del proceso de instalación que se realiza en Ubuntu, es la substitución del comando `make menuconfig` por el comando `make xconfig`.

Parchado y Levantamiento de Kernel

```
cd /usr/src/
tar xjvf rtai-3.6-cv.tar.bz2
ln -s rtai-3.6-cv rtai
tar xjvf linux-2.6.24.7.tar.bz2
mv /usr/src/linux-2.6.24.7 /usr/src/linux-2.6.24.7-
rtai
ln -s linux-2.6.24.7-rtai linux
cd /usr/src/linux
patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-
linux-2.6.24-x86-2.0-07.patch
cp /boot/config-2.6.23.1-42.fc8 .config
make oldconfig
make xconfig
echo "Kernel 2.6.24.7-rtai Configurado con exito"
make -j2
```

```
make -j2 modules
make modules_install
cp arch/i386/boot/bzImage /boot/bzImage-2.6.24.7-rtai
mkinitrd -v /boot/initrd-2.6.24.7-rtai.img 2.6.24.7-
rtai
```

2.2.2.4. Tercera Sección de Instalación

Esta sección de instalación está constituida por herramientas que son instaladas desde su código fuente y que no requieren configuración por parte del usuario por medio de una panel de configuración, por lo tanto su configuración, compilación, e instalación son similares al proceso de realizado en la sección de instalación para Ubuntu. Para realizar esta parte de la instalación seguir los pasos indicados en la sección *2.2.1.4 Tercera Sección de Instalación.*

2.2.2.5. Cuarta Sección de Instalación

Esta sección de instalación es semejante a la sección de instalación en Ubuntu, *2.2.1.5 Cuarta Sección de Instalación*. La única diferencia notable en el momento de la configuración es el reemplazo de comando *make menuconfig* por el comando *make xconfig*, y las respectivas visualizaciones de cada uno. Todos los otros parámetros y pasos de configuración son iguales a los indicados en la sección de instalación en Ubuntu *2.2.1.5*.

2.2.2.6. Quinta Sección de Instalación

Esta sección de instalación está constituida por herramientas que son instaladas desde su código fuente y que no requieren configuración por parte del usuario por medio de una panel de configuración, por lo tanto su configuración, compilación, e instalación son similares al proceso de realizado en la sección de instalación para Ubuntu. Para realizar esta parte de la instalación seguir los pasos indicados en la sección *2.2.1.6.*

CAPÍTULO 3

ANÁLISIS DEL DESEMPEÑO DE LA INTERFAZ PARA APLICACIONES EN TIEMPO REAL (RTAI) PARA EL MANEJO DE SEÑALES ELÉCTRICAS EN LAS DISTRIBUCIONES LINUX: UBUNTU, FEDORA Y CENTOS.

3.1 Identificación de criterios para evaluación de desempeño de RTAI en las Distribuciones Linux: Ubuntu, Fedora y Centos.

En esta sección se detallan los criterios usados para realizar la comparación de desempeño de las herramientas dependiendo del sistema operativo en el cual se encuentran instaladas.

Criterios Lógicos

Para evaluar el desempeño de las herramientas durante las pruebas se ha identificado como criterios lógicos de análisis, a las herramientas de medición de velocidad de respuesta y procesamiento que son incluidas durante la instalación del paquete de RTAI.

Estas medidas están presentes indirectamente durante la ejecución de una aplicación de tiempo real, esto es así ya que el usuario no tiene la necesidad de identificar los datos cuantitativos del procesamiento de la máquina durante su interacción con la aplicación de tiempo real.

A continuación se conceptualizan las dos herramientas principales incluidas en RTAI.

Latencia

La latencia ante una interrupción hardware es el tiempo desde que se produce la interrupción hasta que se ejecuta la primera instrucción de la rutina de tratamiento. Puede haber retrasos debido al acceso al bus de datos.

La latencia ante una interrupción software es el tiempo desde que la señal es generada hasta que la primera instrucción de la tarea es ejecutada. Aquí el valor depende únicamente del acceso a los registros del procesador.

Jitter

El periodo del jitter se refiere a las variaciones en el tiempo que experimenta una tarea cuando se ejecuta de manera repetitiva, por programación o por petición de evento de interrupción.

La Figura 3.1 muestra la representación de la latencia y del jitter de un sistema frente a un evento.

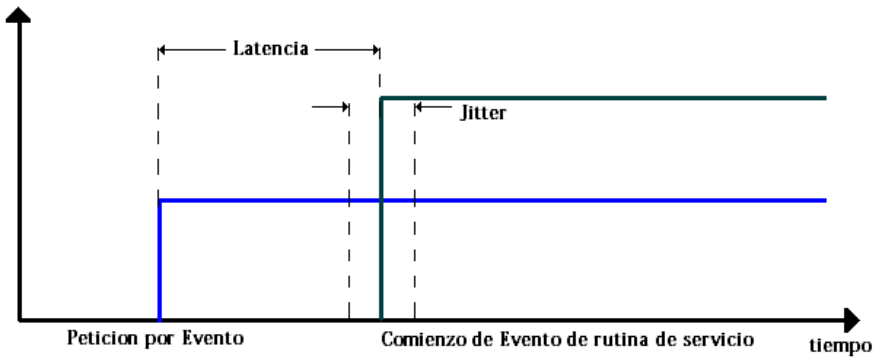


Figura 3.1 Representación de Latencia y de Jitter.

Criterios Gráficos

Los criterios gráficos de evaluación del desempeño de las herramientas son cualidades con las que el usuario interactúa directamente durante el uso de una aplicación de tiempo real, ya que comprenden la definición en la gráfica de señales

y la capacidad máxima de procesamiento gráfico que presenta cada sistema operativo durante el uso de las herramientas instaladas.

Criterios de Desempeño

Se ha definido como criterios de desempeño al tiempo en que le toma responder y estabilizarse a todo el sistema en conjunto, es decir la planta de control de nivel de fluidos y el computador que posee la tarjeta de adquisición de datos. Se ha realizado pruebas con cada uno de los sistemas operativos, utilizando una misma aplicación de tiempo real para cada prueba con un mismo tiempo de muestreo. La aplicación consiste en un esquema de control automático del nivel de fluido con un lazo de realimentación, con los valores de filtros y PID previamente determinados para un punto de operación alrededor de los 20 centímetros de altura del fluido.

3.2 Comparación de Desempeño de RTAI entre Sistema Operativos: Ubuntu, Fedora y Centos.

Criterios Lógicos

Todas las lecturas presentadas a continuación son en términos de tiempo, en el orden de los nanosegundos. Algunas de las medidas presentadas en las gráficas tienen magnitudes negativas debido a que el hardware de la máquina utilizada para la experimentación, ya es eficiente en términos de velocidad y la adaptación como sistema adaptivo que produce RTAI, origina que el cálculo matemático de valores como la latencia generen resultados negativos, es decir teóricamente el sistema

excede el rango de eficiencia para el cual debe funcionar con la alteración realizada por RTAI.

Para poder realizar el análisis de los datos lógicos que proveen las herramientas de RTAI, es necesario indicar que para el mismo se han utilizado criterios, como: latencias máximas, el valor más alto de latencia que se ha presentado dentro de un lapso de tiempo; latencias mínimas, los valores más bajos dentro de un lapso de tiempo; y latencias promedio, el promedio de todas las lecturas de latencia obtenidas en un intervalo de tiempo. Todas estas medidas son aplicables para procesamientos en espacio de kernel y en espacio de usuario.

Por lo tanto, se puede cuantificar desempeño lógico de forma general, es decir las velocidades de respuesta del sistema frente a varias peticiones dentro de un intervalo de tiempo, con un cálculo promedio de dichas medidas, como también se puede considerar el comportamiento crítico del sistema, considerando los máximos valores posibles de rendimiento que toma un sistema en un período de tiempo.

Los siguientes datos fueron obtenidos de los sistemas operativos instalados en un mismo computador, sin alteraciones en su hardware (bajo actividades comunes del sistema) sin la ejecución de una aplicación de tiempo real, ya que de este modo se simula la velocidad de respuesta del sistema bajo la ejecución de una única aplicación de tiempo real que realice las misma peticiones que realizan las herramientas utilizadas para obtener las siguientes medidas.

Cabe indicar que un sistema solo puede ejecutar una aplicación de tiempo real de control de hardware a la vez, ya que estas aplicaciones son manejadas con máxima prioridad, y el intento de ejecución de dos aplicaciones de este tipo no es viable por

que causaría un conflicto de prioridades. Por lo tanto, la realización de las siguientes mediadas no es recomendable realizarse simultáneamente durante la ejecución de una tarea de tiempo real, ya que puede generar resultados no concluyentes, o incluso es posible que las herramientas de medición no funcionen y no se puedan obtener medidas.

La Figura 3.2 muestra las lecturas de latencias máximas de los sistemas operativos, utilizando la herramienta de latencia de Kernel, las cuales han sido medidas en intervalos de 1 segundos. El análisis de puntos críticos máximos del sistema de control es el método más concluyente, ya que nos indica las lecturas de máximo retardo que pueden tener tareas de tiempo real que poseen momento establecido de ejecución, por lo tanto este factor muestra una estimado del máximo tiempo de tardanza que puede sufrir una llamada.

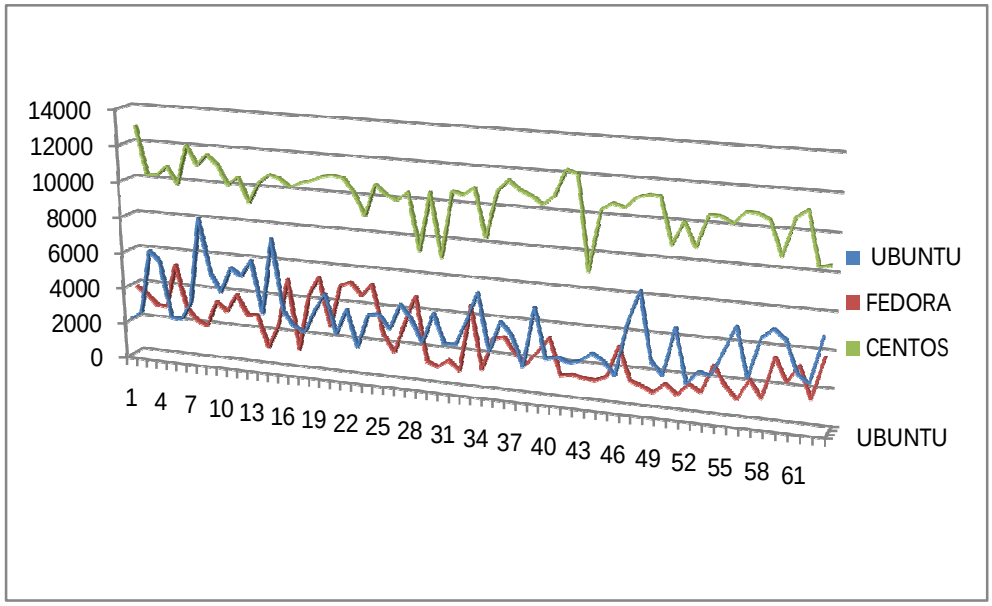


Figura 3.2 Latencias Máximas medidas con Herramienta Latencia del Kernel.

Por medio de los datos de la Figura 3.2 se concluye que el sistema operativo Centos presenta la mayor latencia de este tipo, y que Ubuntu y Fedora poseen latencia del mismo orden. Lo que significa que los retardos más grandes que presenta Centos, son más extensos que los que presentan Fedora y Ubuntu. Centos con lecturas alrededor de 11 microsegundos, Fedora y Ubuntu con lecturas del orden 3 microsegundos.

La Figura 3.3 muestra las lecturas de latencia máxima obtenidas con la herramienta latencia de usuario y expresa los mismos criterios que los datos mostrados en la anterior gráfica, respecto a las operaciones de usuario. Se observa un comportamiento semejante al obtenido con la herramienta para lecturas de latencias de kernel.

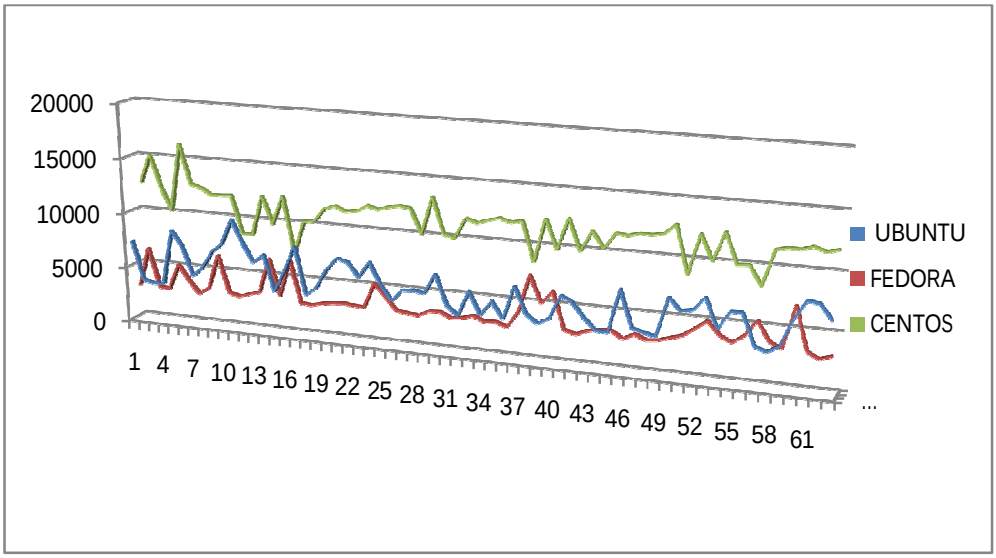


Figura 3.3 Latencias Máximas medidas con Herramienta Latencia del Usuario.

En los datos mostrados en la Figura 3.3, Centos presenta nuevamente los valores de latencia superiores a Fedora y a Ubuntu.

La Figura 3.4 muestra las lecturas de latencias mínimas de los sistemas operativos, utilizando la herramienta de latencia de Kernel, las cuales han sido medidas en intervalos de 1 segundos. Estos datos muestran cual de los sistemas operativos es capaz de alcanzar los mínimos valores de latencia dentro de los ciclos de análisis.

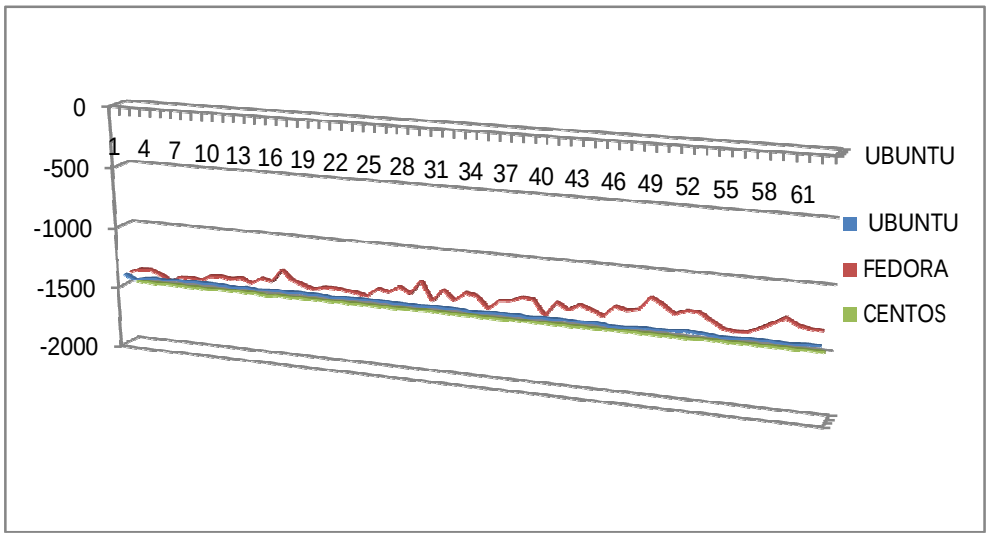


Figura 3.4 Latencias Mínimas medidas con Herramienta Latencia del Kernel.

Por medio de los datos de la Figura 3.4 se concluye que el sistema operativo Centos, dentro de un bloque de lecturas realizadas cada segundo es capaz de presentar latencias inferiores. Luego de este, Ubuntu es el segundo sistema operativo capaz de alcanzar los valores más bajos de latencia. Es decir probabilísticamente, los retardos más cortos que presentan Centos y Ubuntu son más pequeños que los mejores retardos que presenta Fedora.

La Figura 3.5 muestra las lecturas de latencia mínima obtenidas con la herramienta latencia de usuario y expresa los mismos criterios que los datos mostrados en la anterior gráfica, pero respecto a las operaciones de usuario. Se observa que Centos presenta nuevamente, valores mínimos entre los tres sistemas operativos, Ubuntu es el siguiente inmediato, mientras Fedora presenta latencias mínimas relativamente más altas que los otros sistemas operativos.

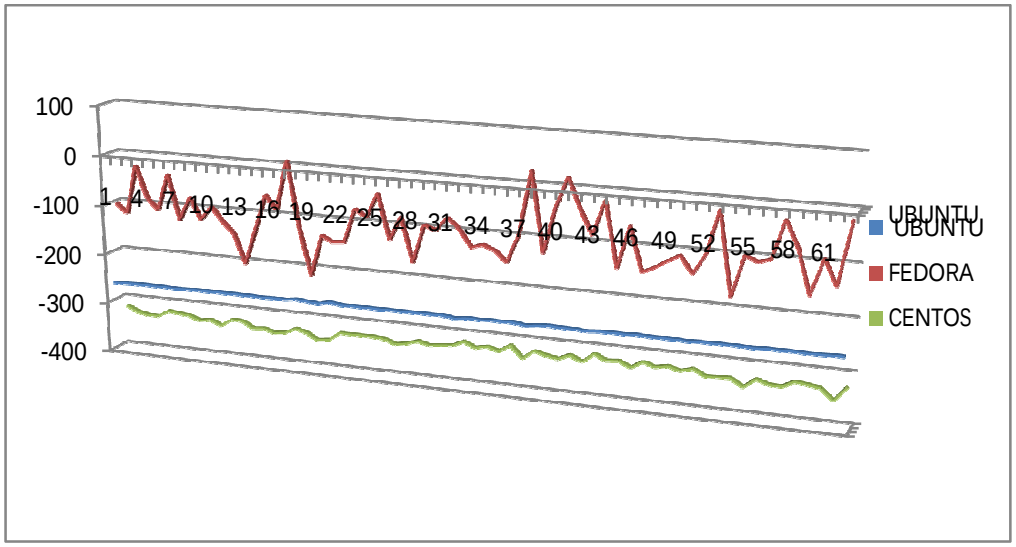


Figura 3.5 Latencias Mínimas medidas con Herramienta Latencia del Usuario.

Los datos mostrados en la Figura 3.5, muestran que Ubuntu presenta una latencia mínima casi constante mientras Fedora tienes latencias mínimas más altas que Centos y Ubuntu, y muestra grandes fluctuaciones, lo que significa que en términos cantidades de tiempo muy pequeñas, es menos probable conseguir la respuesta más rápida con Fedora que con Ubuntu o con Centos. Estas diferencias de velocidades de respuesta son en el orden de fracciones de microsegundos.

La gráfica de la Figura 3.6 muestra las lecturas de latencias promedio de los sistemas operativos, utilizando la herramienta de latencia de Kernel, las cuales han sido medidas en intervalos de 1 segundos. Esto nos muestra el promedio de todas lecturas de latencia realizadas durante un segundo, incluyendo las lecturas más altas y más bajas.

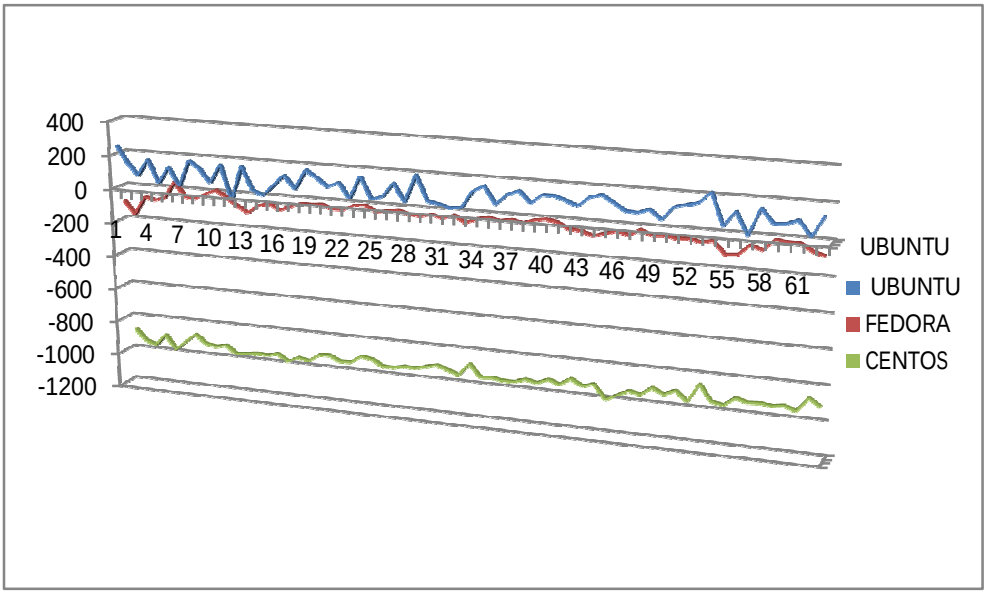


Figura 3.6 Latencias Promedio medidas con Herramienta Latencia del Kernel.

Valores cercanos a cero significan que las latencias máximas y mínimas están equitativamente distribuidas. Las curvas que se proyectan mayoritariamente positivas o negativas, muestran latencias distribuidas de forma no equitativa.

Los datos que se pueden apreciar en la Figura 3.6 indican que Ubuntu y Fedora tienen una distribución equitativa de latencias mínimas y máximas, mientras Centos muestra tener latencias generalmente más bajas durante cada periodo de

muestreo, en relación a Ubuntu y a Fedora, pese a que Centos presenta los datos de latencia máxima más altas, lo que es un indicador de una relativa inestabilidad de los tiempos de latencia en relación a los otros dos sistemas operativos.

La Figura 3.7 muestra las lecturas de latencia promedio obtenidas con la herramienta latencia de usuario y expresa los mismos criterios que los datos mostrados en la anterior gráfica, pero respecto a las operaciones de usuario. En la gráfica se observa los sistemas operativos Ubuntu y Fedora poseen valores de latencia promedio relativamente superiores a los de Centos, pese a que los valores de las latencias promedio de usuario de los tres sistemas operativos son bajos en relación a los valores de latencia de usuario máxima. Se identifica que Ubuntu tiene los valores más altos de latencia promedio de usuario. Fedora presenta valores muy cercanos a los de Ubuntu.

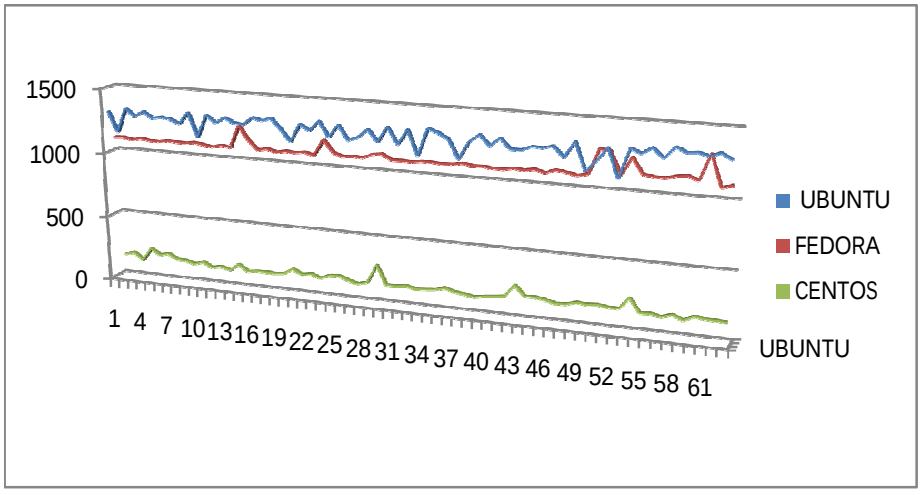


Figura 3.7 Latencias Promedio medidas con Herramienta Latencia del Usuario.

Esto encuentra su justificación en la cantidad de la capacidad de procesamiento que es invertida para el soporte gráfico de cada uno de los sistemas operativos, por lo tanto en la carga gráfica que maneja cada sistema operativo. Dado que Ubuntu está orientado a brindar un ambiente gráfico más robusto que los Centos y Fedora, este puede presentar los mayores valores de latencia. Igual explicación para el caso de Fedora, que prioriza más la calidad del entorno gráfico que en el sistema operativo Centos.

En otras mediciones como las de jitter, los valores son constantes durante todo el ciclo de mediciones, y junto a estos también se obtienen valores constantes de latencias máximas y mínimas obtenidas por la herramienta de medición de jitter.

A continuación se muestra la Tabla II con los valores obtenidos por la herramienta preempt para el cálculo del jitter en el espacio de usuario.

Usuario	**Ubuntu**	**Fedora**	**Centos**
Jitter Rápido	10375	15956	14791
Jitter Lento	12110	16344	13696
Latencia Máxima	9840	10648	12566
Latencia Mínima	-724	-591	-862

Tabla II. Valores Obtenidos para la medición de Jitter en el espacio de usuario.

A partir de los datos mostrados en la Tabla II se puede constatar que Ubuntu posee las latencias máximas y mínimas más eficientes de entre los tres sistemas operativos y que sus periodos de jitter son también los más cortos de entre los tres sistemas operativos. Las medidas mostradas están en el orden de nanosegundos.

Cabe recalcar que mientras los sistemas operativos Ubuntu y Fedora presentaron valores constantes durante de la prueba, el sistema operativo Centos presentó cambios súbitos de valor, durante un periodo de tiempo antes de alcanzar estabilización en los valores.

Criterios Gráficos

Los datos mostrados en esta sección han sido determinado mediante pruebas experimentales, calificando la calidad gráfica que presenta la herramienta de visualización de aplicaciones de tiempo real XRtaiLab. Las aplicaciones de tiempo real que se han utilizado como prueba para evaluar el desempeño de XRtaiLab fueron elaboradas en base a un esquemático de generación de onda seno, y convertidas a aplicación de tiempo real siguiendo los pasos detallados en el Anexo A.

A continuación la figura 3.8 muestra el esquemático utilizado para la generación de onda seno. En cada prueba se alteró la frecuencia de muestro del esquemático. Se detalla el proceso de creación de este esquemático en el Anexo A.

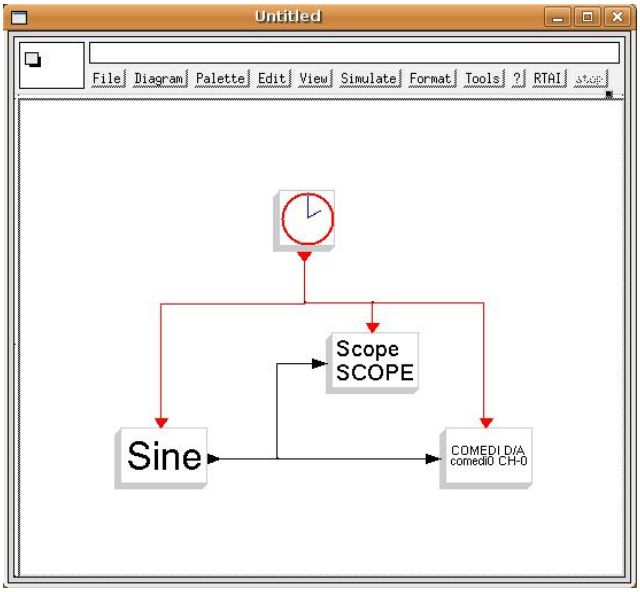


Figura 3.8. Esquemático de Scicos para generación de Onda Seno.

En esta sección se muestra los cuadros de comparación cualitativa del desempeño gráfico de cada sistema operativo durante el uso de las herramientas.

La Tabla III muestra las definiciones de los experimentos, en la cual se detallan los valores de período de muestreo aplicado al esquemático y la cantidad de divisiones por segundo que se utiliza para visualizar la señal en el osciloscopio virtual.

La definición de la gráfica se altera de acuerdo con la cantidad de divisiones por segundo que se utilice en el osciloscopio virtual, por lo que la visualización con dos cantidades diferentes de divisiones por segundo de una misma aplicación es considerada como dos experimentos diferentes.

Experimento	Período de Muestro	Div/Seg
1	0,1	0,1
2	0,1	1
3	0,01	0,1
4	0,01	1
5	0,001	0,1
6	0,001	1
7	0,0001	0,1
8	0,0001	1
9	0,0005	0,1
10	0,0005	1
11	0,00001	0,1
12	0,00001	1
13	0,00005	0,1
14	0,00005	1

Tabla III. Tabla de definición de Experimentos.

En la Tabla IV se encuentra el resumen de comportamiento gráfico de cada sistema frente a cada experimento, en la cual se califica la calidad de visualización que presentó cada experimento.

Experimento	Ubuntu	Fedora	Centos
1	Bueno	Distorcionada	Segmentación
2	Bueno	Distorcionada	Graficación Lenta
3	Excelente	Excelente	Graficación Lenta
4	Excelente	Excelente	Graficación Lenta
5	Excelente	Muy bueno	Muy bueno
6	Excelente	Muy bueno	Muy bueno
7	Excelente	Muy bueno	Bueno con problemas de maximizacion
8	Excelente	Inhibición	Inhibición de Computador
9	Muy bueno	Bueno	Inhibición de Computador
10	Muy bueno	Bueno	Inhibición de Computador
11	Sin visualizacion	Inhibición de Computador	Inhibición de Computador
12	Sin visualizacion	Inhibición de Computador	Inhibición de Computador
13	Se cierra XRtaiLab	Bueno	Inhibición de Computador
14	Se cierra XRtaiLab	Se cierra XRtaiLab	Inhibición de Computador

Tabla IV. Tabla de Evaluación de Resultados por Sistema Operativo.

En la Tabla V se define cuantitativamente las clases de desempeño gráfico de un sistema operativo frente a un experimento con una aplicación de tiempo real, en relación a la Tabla IV.

Calidad de Desempeño Gráfico	**Valor**
Excelente	6
Muy bueno	5
Bueno y buena con problemas de maximizacion de ventana	4
Graficación Lenta o Distorcionada	3
Con retrazos o segmentacion	2
Sin Visualización	1
Inhibición o se Cierra Xrtailab	0

Tabla V. Tabla de Definición Cuantitativa de Desempeño.

La Tabla VI muestra los valores de desempeño que presenta cada sistema operativo frente a cada uno de los experimentos utilizando la Tabla V para catalogar el desempeño en términos cuantificables.

Experimento	Ubuntu	Fedora	Centos
1	4	3	2
2	4	3	3
3	6	6	3
4	6	6	3
5	6	5	5
6	6	5	5
7	6	5	4
8	6	0	0
9	5	4	0
10	5	4	0
11	1	0	0
12	1	0	0
13	0	4	0
14	0	0	0

Tabla VI. Tabla de Valoración Cuantitativa de Sistemas Operativos.

La Figura 3.9 muestra la representación gráfica de la tabla de desempeño de los sistemas operativos mostrada en la Tabla VI.

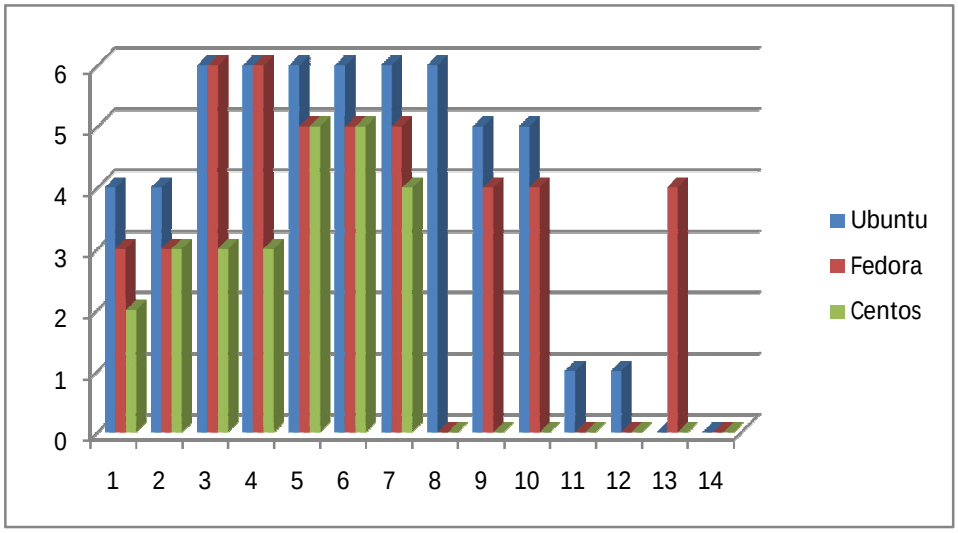


Figura 3.9 Desempeño Gráfico de los Sistemas Operativos.

En la gráfica de la Figura 3.9 se visualiza que Ubuntu ha tenido un desempeño superior en casi todas las pruebas de rendimiento gráfico, el segundo sistema operativo más eficiente en términos gráficos es Fedora.

Criterios de Desempeño

La Figura 3.10 muestra los tiempos de estabilización de la planta ante cambios de nivel, de una altura inicial a una altura final, dependiendo del sistema operativo el cual maneja a la planta. Los datos utilizados para construir la Figura 3.10 fueron obtenidos utilizando un esquemático con un lazo cerrado de control, elaborado en base a los datos de la planta de control de fluidos diagramada en la Figura 3.11. El esquemático anteriormente mencionado fue utilizado para crear una aplicación de tiempo real para el control de dicha planta. La misma aplicación fue utilizada para todas las pruebas, sin alterar ningún parámetro.

Las variaciones en los tiempos de respuesta de la planta no son producidas por cambios en los datos para construir el esquemático de control de la misma. Las variaciones en los tiempos de estabilización son consideradas dentro de este trabajo como muestras de alteraciones mínimas ocurridas por efectos del procesamiento y la forma de manejo de operaciones que posee cada sistema operativo. Ubuntu ha presentado tiempos relativamente más cortos de estabilización que los sistemas operativos Fedora y Centos. Centos en estas pruebas es el segundo sistema operativo más eficiente. Toda variación de tiempo de estabilización es mínima como para significar una alteración importante dentro del sistema de control de la planta.

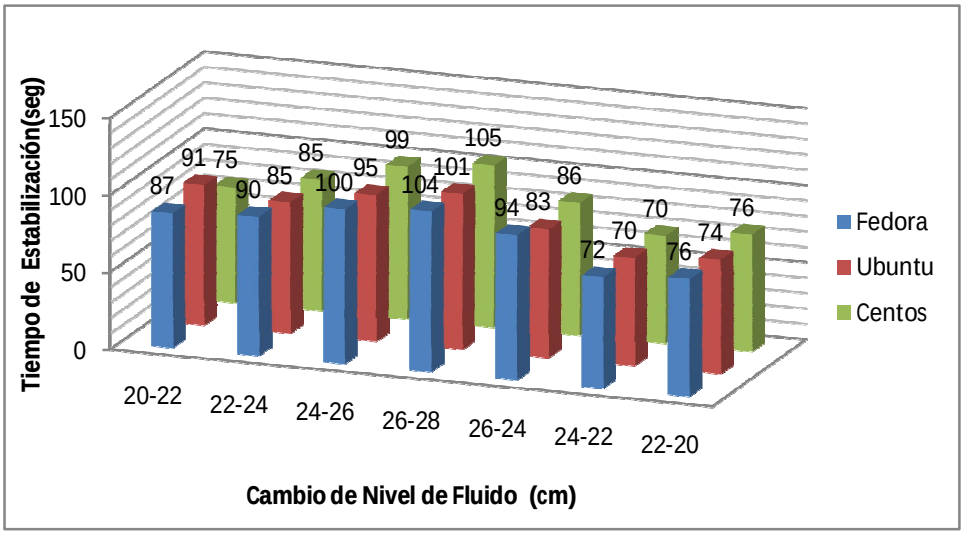


Figura 3.10 Desempeño Práctico del la Planta con cada Sistema Operativo.

La Figura 3.11 muestra el esquema de la planta de control de fluidos, y los componentes que constituyen al sistema.

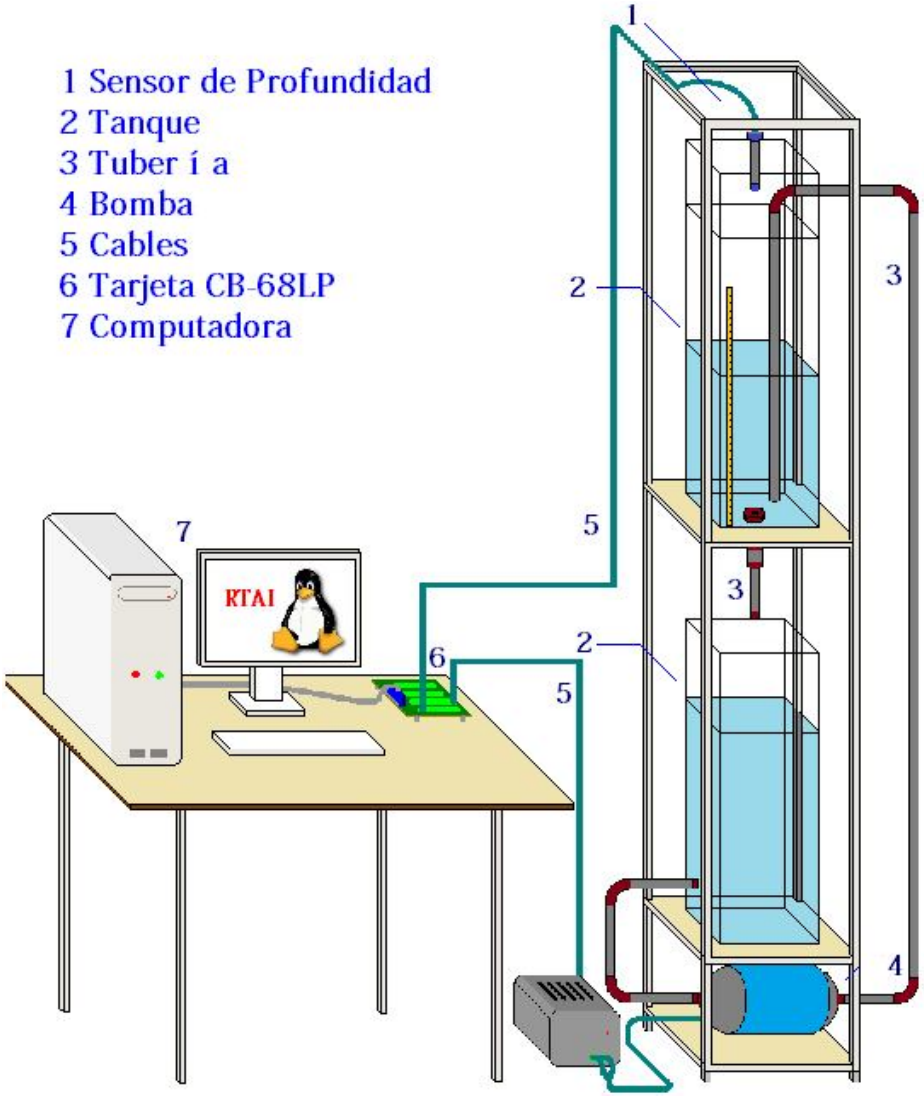


Figura 3.11 Esquema de diseño físico de la planta de control de nivel de fluido.

3.3 Análisis de Herramienta Scilab-Scicos

3.3.1 Descripción de Herramientas Scilab y Scicos

Scilab fue desarrollado para aplicaciones de control de sistemas y procesamiento de señales. Es un programa de código libre y distribuido sin costo. Scilab contiene: un interpretador, librerías de funciones propias de Scilab y librerías con funciones de Fortran y C.

Uno de los atributos básicos de la sintaxis Scilab es su capacidad de manejo de matrices: con manipulaciones básicas como concatenación, extracción o transposición que son llevadas a cabo inmediatamente tan bien como las operaciones básicas como suma y multiplicación. Scilab también maneja objetos más complejos que matrices numéricas. Por ejemplo, para aplicaciones de control se requiere manipular matrices racionales o polinomiales. Esto es hecho en Scilab manipulando listas y escribiéndolas, lo que permite una representación simbólica natural de objetos matemáticos complicados como funciones de transferencia.

Scilab provee funciones para el análisis de sistemas no lineales. La integración de modelos dinámicos explícitos a implícitos puede ser llevada a cabo numéricamente. La barra de herramientas Scicos permite una definición gráfica y simulación de sistemas híbridos. Existen facilidades de optimización numérica para la optimización no lineal, optimización cuadrática y lineal.

Scilab tiene una programación ambiental abierta en donde la creación de funciones y librerías de funciones accesibles para el usuario. Las funciones son reconocidas como objetos de datos en Scilab y pueden ser manipulados o creados como otros objetos. Por ejemplo, las funciones pueden ser definidas en Scilab y pasadas como una entrada o salida de argumentos de otras funciones. Además, Scilab acepta caracteres que permiten la creación de funciones en línea.

Finalmente, Scilab se puede comunicar fácilmente con Fortan o programas de C. Esto permite utilizar paquetes estandarizados y librerías en el ambiente de Scilab.

Scilab brinda los siguientes atributos como ambiente de trabajo:

- Tipos de datos alterables con una fácil sintaxis.
- Grupos de funciones para varios tipos de cálculos.
- Capacidad de colocar nuevas funciones en términos de programación.

Scilab provee de constantes pre definidas como pi, el número de Euler, la raíz cuadrada de -1. Permite trabajos con escalares y con matrices, crear, extraer rangos, operaciones con matrices. También permite definición de funciones por el usuario, trabajos con integrales y soluciones de ecuaciones no lineales. Scilab posee un módulo semejante a Simulink el cual permite la simulación y la programación grafica de ciertos sistemas, tales como sistemas lineales, eléctricos, y sistemas de control de forma que el usuario utiliza los bloques incluidos

dentro de Scicos, los cuales están agrupados por el tipo de función que cumplen, el conjunto de estos bloques se los conoce como paletas.

La Figura 3.12 muestra las paletas incluidas por defecto dentro de Scicos

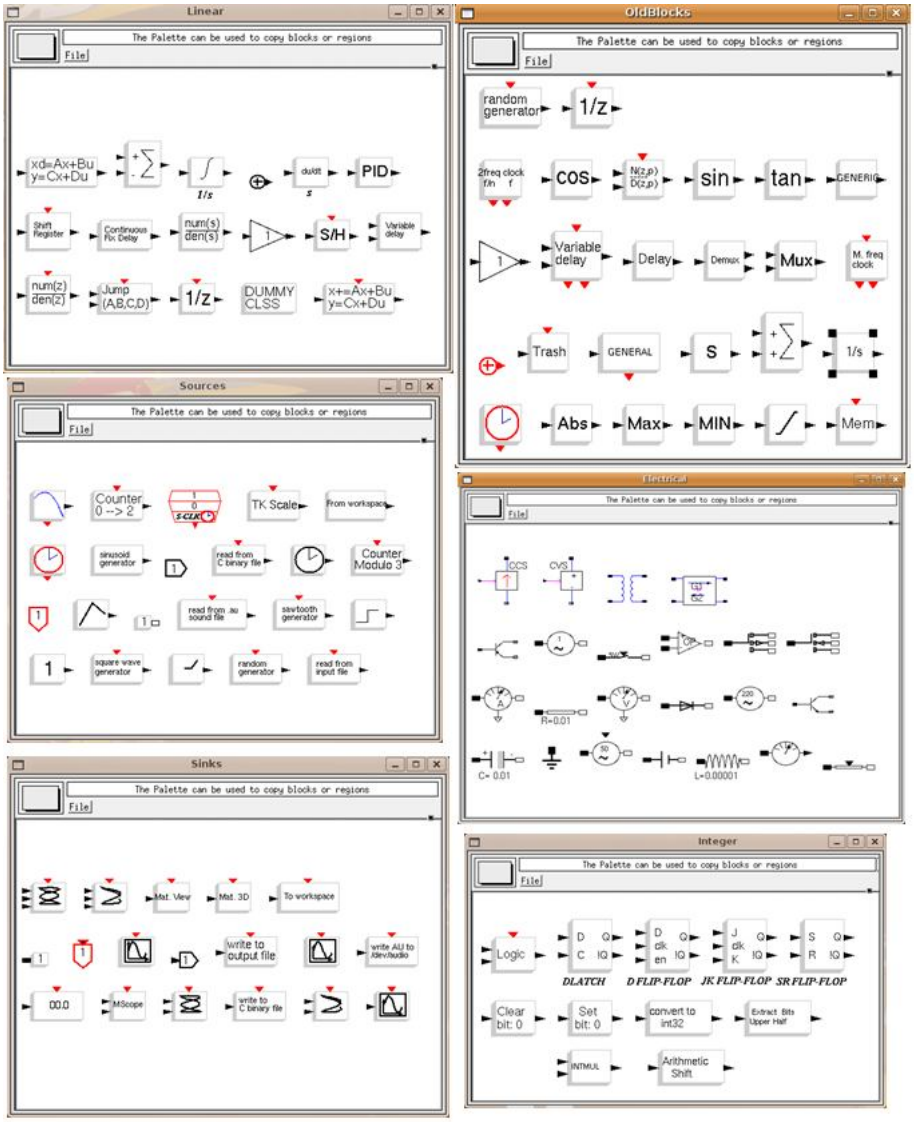


Figura 3.12 Paletas Importantes Incluidas en Scicos.

En la Figura 3.13 se muestra la paleta que se ha sumado durante el proceso de instalación de RTAI. En esta paleta encontramos las fuentes, los visualizadores y los elementos para realizar la lectura y escritura de datos y señales en el hardware, es decir la tarjeta de adquisición de datos.

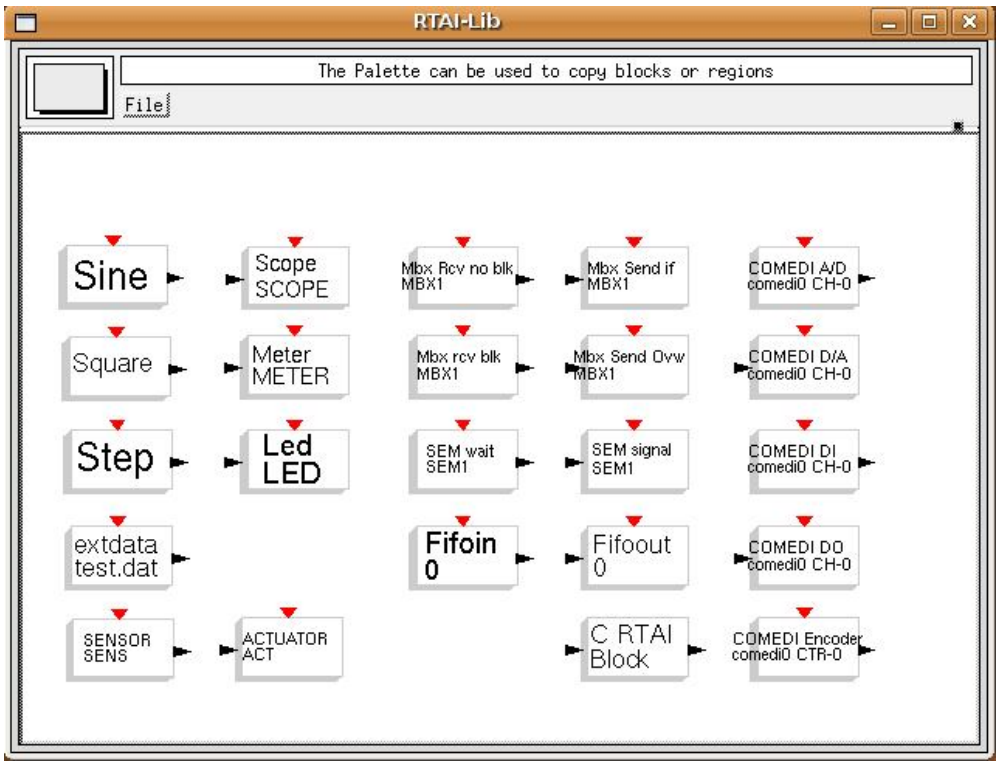


Figura 3.13. Paleta de la Librería RTAI.

3.3.2 Herramientas Cargables en Scilab.

Scilab también permite la carga de herramientas que no son instaladas por defecto, como herramientas para aplicaciones especializadas que estén publicados en el foro de Scilab. Un ejemplo de esto, es que Scilab admite la instalación de un nuevo bloque de paletas y elementos para Scicos dirigidos exclusivamente a la estructuración de simulaciones de sistemas de comunicaciones, como es el caso de la herramienta Modnum.

En la Figura 3.14 se muestran las paletas y los elementos más importantes que se cargan dentro de Scicos después de la instalación de la herramienta Modnum. Para poder realizar la instalación de herramientas como Modnum, es necesario seguir los pasos dentro del script INSTALL que se encuentra dentro del paquete de instalación de este tipo de herramientas. Para más detalle del proceso de instalación de esta herramienta refiérase al Anexo A.

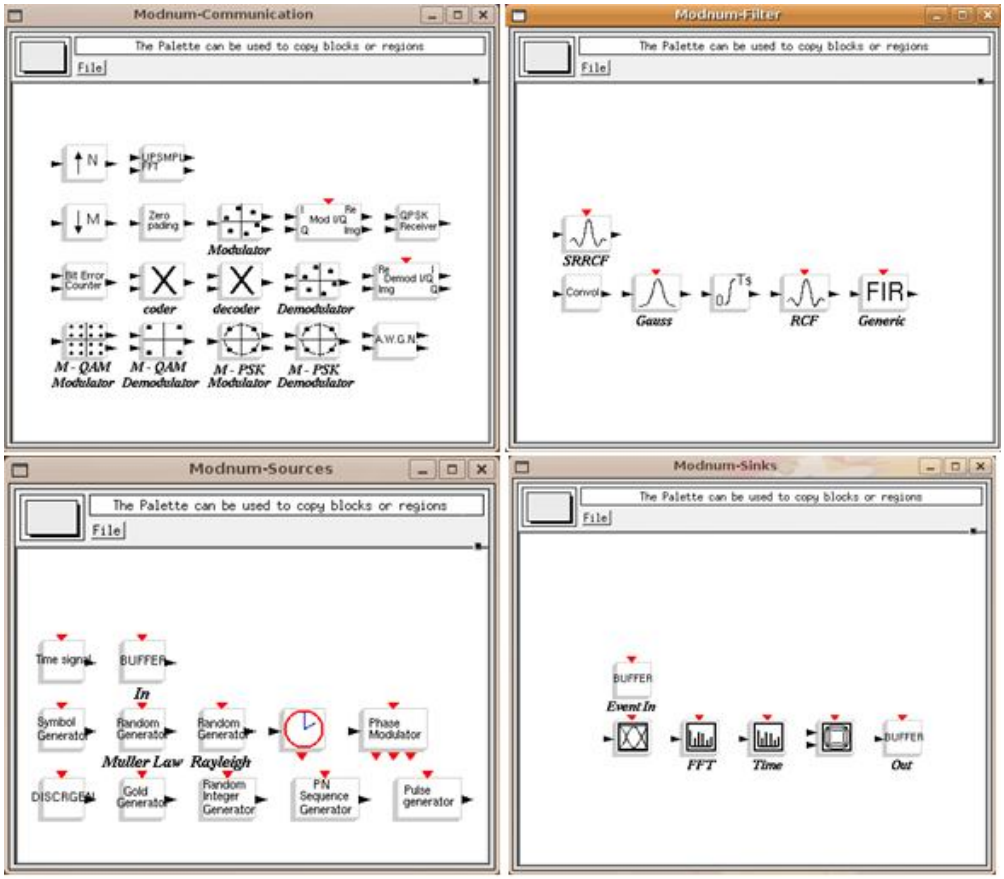


Figura 3.14. Paletas de la Librería Modnum.

Es dentro de la aplicación Scicos en la cual se realiza los esquemáticos para luego elaborar las aplicaciones ejecutables que son las que interactuaran y se ejecutaran en tiempo real. Los pasos necesarios para la elaboración de los esquemáticos y la compilación de las aplicaciones de tiempo real están detallados en el Anexo A.

3.4 Comparación y Monitoreo de señales eléctricas en tiempo real a través de los osciloscopios virtuales.

En esta sección se analizará el osciloscopio virtual provisto por la herramienta XRtaiLab. Este osciloscopio es estrictamente local, osciloscopio tales como de herramientas remotas como JRtaiLab, no están orientados para trabajo de visualización dentro de la máquina servidora, sino para realizar lectura desde otro computador conectado en red y será analizado en el Capítulo 4.

Localmente es posible trabajar con dos herramientas de visualización de señales XRtaiLab y QRtaiLab, ambas poseen interfaces diferentes y por lo tanto osciloscopios diferentes.

En la Figura 3.15 se muestra la interfaz que presenta la herramienta XRtaiLab y su osciloscopio virtual.

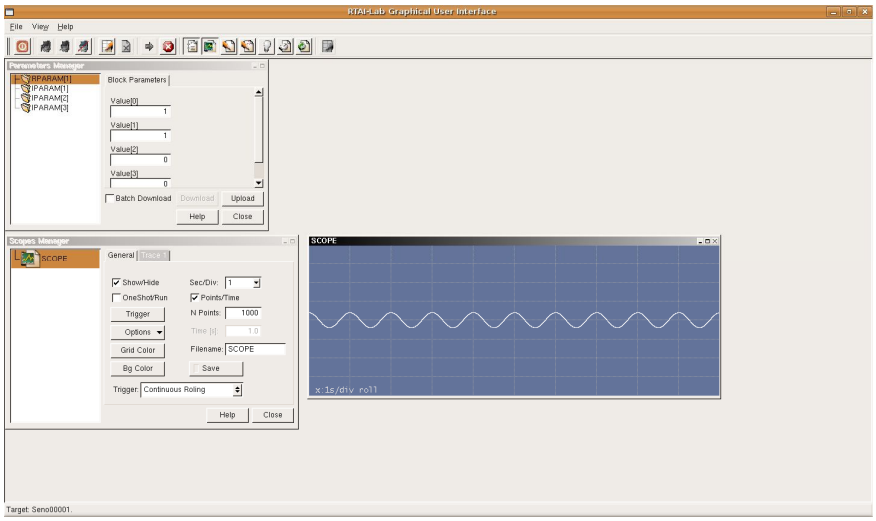


Figura 3.15 Herramienta XRtaiLab.

Se debe mencionar que dado que ambas herramientas son instaladas en su forma fuente y que han sido creadas en lenguajes de programación gráficas específicas en independientes del sistema operativo, por lo tanto ninguna de las herramientas cambia su interfaz al ser instalada en un sistema operativo diferente, es decir esencialmente ambas herramientas se las visualizar de igual forma en los sistemas operativos Ubuntu, Fedora y Centos.

La herramienta XRtaiLab posee un proceso de instalación que está virtualmente incluido en el proceso de instalación de la herramienta Rtai. La herramienta QRtaiLab posee un proceso extra de instalación, y su instalación también significa alteraciones en algunos de los pasos de la instalación de todas las herramientas del sistema, por lo que pese a que es posible contar con ambas herramientas instaladas en el mismo computador servidor, es recomendable solo tener una de las dos herramientas instaladas. Durante las pruebas realizadas para este trabajo se ha utilizado la herramienta XRtaiLab, herramienta propia de RTAI.

La Figura 3.16 muestra la interface de la herramienta QRtaiLab y su osciloscopio virtual.

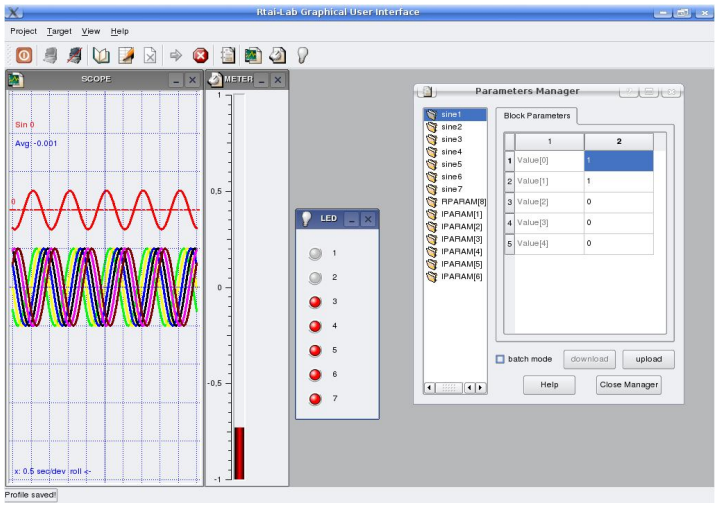


Figura 3.16 Herramienta QRtaiLab.

Se puede observar que las señales en QRtaiLab poseen color, la herramienta XRtaiLab también permite colocar color a las señales para la distinción de las mismas.

La figura 3.17 muestra un ejemplo de cómo se visualizan un conjunto de señales seno, en el osciloscopio de XRtaiLab, sin alterar el valor por defecto de XRtaiLab de color de las señales

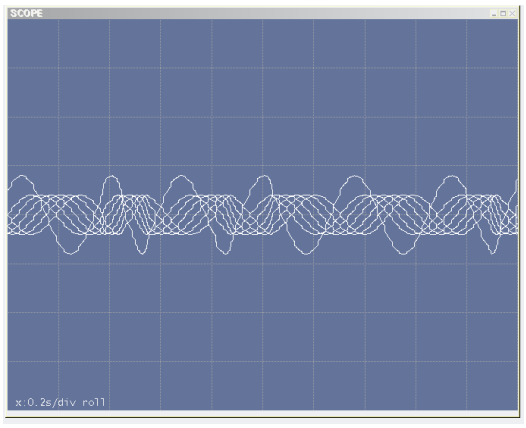


Figura 3.17 Osciloscopio virtual de la Herramienta XRtaiLab.

A continuación se muestra un ejemplo de la visualización de las mismas señales mostradas en la gráfica anterior con la herramienta XRtaiLab. La figura 3.18 muestra las mismas señales mostradas en la gráfica anterior, utilizando el osciloscopio virtual de QRtaiLab.

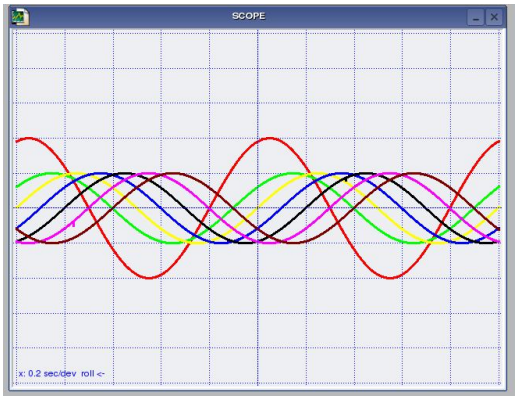


Figura 3.18 Osciloscopio virtual de la Herramienta QRtaiLab.

Realizando una comparación entre ambos osciloscopios presentados por las herramientas, el osciloscopio de QRtaiLab posee mejor definición y un mejor manejo de cantidad de datos por gráfica lo cual produce que el muestreo de las señales tenga una mejor calidad y que la velocidad de la gráfica se aproxime más a la velocidad del muestreo físico de la señal.

3.5 Conclusiones de los resultados obtenidos

En base a los resultados obtenidos en las pruebas realizadas, considerando los tres criterios. Se puede concluir que aunque las diferencias entre el desempeño de los tres sistemas operativos, durante las pruebas funcionales óptimas, no son muy amplias, el sistema operativo Ubuntu presenta, superioridad en la mayoría de los criterios utilizados para las comparación, cualidades que son mutuamente sustentable, como el vinculo existente entre las velocidades de estabilización del sistema total y la capacidad de respuesta gráfica. Durante la mayor parte de las pruebas de desempeño gráfico Ubuntu presenta el mejor desempeño, seguido de Fedora y por último Centos.

Como observación principal al desempeño del sistema operativo Ubuntu, es capaz de alcanzar procesos con frecuencias de muestreo más altas que los sistemas operativos Centos y Fedora.

Los sistemas operativos Ubuntu y Fedora presentaron valores de latencia máxima más bajos que los presentados por el sistema Centos, lo que significa que sus mayores retrasos son más cortos que los mayores retrasos que puede sufrir Centos, pese a que todos los retrasos en cuestión son del orden de los microsegundos, por lo tanto no podrían ser percibidos por la mayoría de los sistemas a controlar, lo que sustenta el hecho de que los tres sistemas operativos manejan adecuadamente las operaciones de tiempo real, dentro de un cierto límite de frecuencias de trabajo.

Los sistemas operativos Ubuntu y Centos presentaron los valores más bajos de

latencia mínima, en relación a los valores presentados por Fedora, lo que significa que ambos sistemas operativos son capaces de alcanzar eventualmente los retrasos más cortos. En estos términos Ubuntu es el sistema operativo que califica en ambos aspectos lógicos.

También se recalca el hecho de que, a pesar de que los tres sistemas operativos presentan las cualidades de desempeño que les permite calificar como óptimos para el manejo de aplicaciones de tiempo real, el sistema operativo Centos, dentro de los marcos de su comportamiento, presentaba latencias muy distribuidas lo que podría constituir inestabilidad bajo ciertos regímenes de operación.

Estos resultados se sustentan en el hecho de que originalmente RTAI fue diseñado para ambientes provenientes de Debian. Ubuntu es un sistema operativo basado en Debian, el cual ha heredado características como mecanismos de instalación repositorios y herramientas. Respecto a Fedora y Centos, entra en consideración la capacidad de respuesta gráfica que es capaz de manejar un sistema de forma natural. Fedora está orientado a tener un mejor ambiente gráfico ya que ha sido elaborado para poder interactuar de forma gráfica con el usuario. Centos posee un desempeño diferente ya que está diseñado originalmente para manejo de servicios y operaciones por consola, por lo que está orientado a manejar una actividad gráfica relativamente pequeña en relación a sistemas operativos como Ubuntu, sin que esto signifique es no sea capaz de manejar gráficas por completo.

Objetivamente las capacidades de desempeño de los sistemas operativos no

difieren en gran medida y el orden de desempeño medido durante las pruebas realizadas, coloca a Ubuntu como el sistema operativo con mejor desempeño general, seguido de Fedora. El sistema operativo que presento el desempeño menos optimo durante las pruebas es el sistema Centos.

CAPITULO 4

VISUALIZACIÓN REMOTA DE SEÑALES ELÉCTRICAS EN TIEMPO REAL.

4.1. Definición de las Herramientas para Visualización Remota.

RTAI-XML

RTAI-XML es un componente de tipo servidor del proyecto RTAI, implementa un servicio basado en diseño y desarrollo del control de aplicaciones en tiempo real.

Su funcionamiento se basa en un puerto de red de un servidor en espera de llamadas entrantes donde el proceso de tiempo real o la tarjeta, están en ejecución o están listos para estarlo.

Este proyecto nació con el objetivo de satisfacer las necesidades de un grupo de universitarios, principalmente enfocado en lograr una plataforma flexible para el aprendizaje de diseño y control de sistemas, que permitan a los estudiantes probar sus programas de forma remota, a través de Internet.

Un cliente o cualquier programa genérico, puede comunicarse con el servidor a través de TCP / IP, utilizando un protocolo estándar basado en XML, y por lo tanto interactuar con el destino, con la finalidad de vigilar el estado del proceso de tiempo real, o simplemente para ver las señales generadas por el sistema, además de tener la facilidad de buscar y cambiar todos los parámetros en tiempo real que son modificables en el ejecutable que está corriendo en el servidor y que ha sido creado en Scicos.

En otras palabras, RTAI-XML proporciona una forma sencilla de interacción remota de aplicaciones de control, añadiendo flexibilidad al proyecto RTAI, sin perder las características de una aplicación abierta y estándar.

ARQUITECTURA

RTAI-XML proporciona una ruta para el control remoto de aplicaciones en tiempo real, basadas en la capa de abstracción XML. El objetivo de RTAI-XML es lograr una estructura como se muestra en la Figura 4.1, donde el proceso de control en tiempo real (servidor) y los procedimientos relacionados con la interacción del usuario (Cliente) se ejecutan de manera independiente, por lo general en dos CPUs diferentes, comunicándose a través de la red (Internet).

Figura 4.1. Arquitectura de RTAI-XML

ESTRUCTURA

La Estructura de RTAI-XML sigue la arquitectura general. Básicamente se enfoca en dos componentes principales que son:

- RTAI-XML Lado Interno: Servidor- Tarjeta de Comunicacion
- RTAI-XML Lado Externo: Servidor - Comunicación SRT de Cliente

Como se puede observar en la Figura 4.2, el núcleo del sistema es el servidor de RTAI-XML que funciona como un puente entre el dominio RTAI y la Red en tiempo real TCP/IP, es decir crea instancias de un procedimiento SRT de la red Internet a la red TRH RT_NET. Por el lado de Internet, RTAI XML-RPC implementa un servidor basado en XML, dentro del dominio de RTAI. Por el lado del cliente RTAI-XML debe ser capaz de crear una instancia de procedimientos en los datos del servidor y el intercambio organizado de tipos datos estándar.

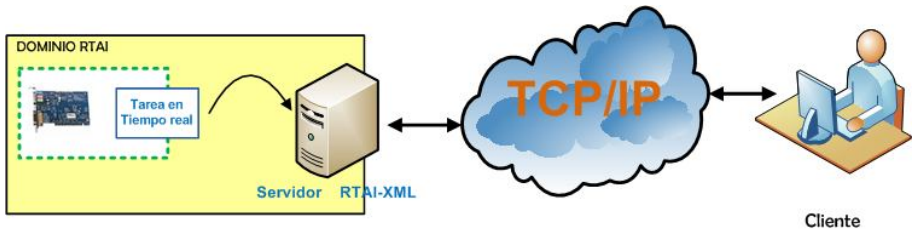


Figura 4.2. Estructura de RTAI-XML

RTAI-XML en términos modernos es un proveedor de servicios Web.

RPC –PROTOCOL

El RPC (Remote Procedure Call) Llamada a Procedimiento Remoto, es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos, es decir trabaja con un alto nivel de abstracción, ignorando los mecanismos de comunicación subyacentes y las características de su implementación. RPC es muy utilizado dentro del modelo cliente-servidor. Siendo el cliente el que inicia el proceso solicitando al servidor que ejecute cierto procedimiento o función y enviando de vuelta el resultado de dicha operación al cliente.

La idea con RPC es hacer ver a una llamada remota como si fuera local, por esto la invocación debe ser transparente para el que la utiliza. La transparencia en RPC se logra agregando un Stub o proxy, que son procedimientos locales tanto al cliente como al servidor. Ver Figura 4.3.

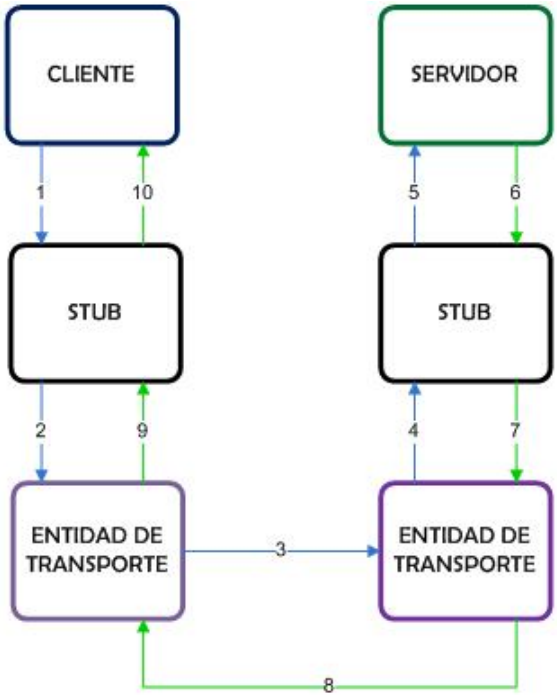


Figura 4.3. Modelo RPC Cliente-Servidor

A continuación se detallan los pasos que realiza un Cliente para invocar un procedimiento remoto en un Servidor:

- El Cliente llama un procedimiento local llamado el Client Stub.
- El Client Stub empaqueta los parámetros, construye un mensaje y lo envía a la entidad de transporte.
- La entidad de transporte local lo envía a la entidad de transporte donde se encuentra el Server Stub
- La entidad de transporte remota envía el mensaje al Server Stub.

- El Server Stub desempaqueta los parámetros, identifica el procedimiento y lo ejecuta.
- El Server Stub recibe el resultado del servidor local
- El Server Stub empaqueta la respuesta, construye un mensaje y lo envía a la biblioteca de ejecución.
- La entidad de transporte remota envía de nuevo a la entidad de transporte original.
- La entidad de transporte local envía el mensaje al Client Stub.
- Client Stub desempaqueta el resultado y lo retorna de la misma forma que un procedimiento local.

Hay distintos tipos de RPC, muchos de ellos estandarizados. Hoy en día se utiliza el XML como lenguaje para definir el IDL y el HTTP como protocolo de red, dando lugar a lo que se conoce como servicios web. Ejemplos de éstos pueden ser SOAP o XML-RPC.

XML-RPC

XML-RPC es un protocolo para la realización de llamadas a procedimiento remoto a través de TCP/IP, utiliza XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. La figura 4.4 nos ilustra de forma gráfica lo mencionado anteriormente.

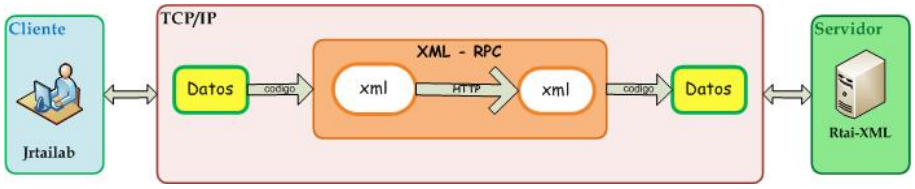


Figura 4.4. Comunicacion XML-RPC

Es un protocolo muy simple ya que solo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.

La Figura 4.5 nos ilustra la interacción de objetos que se genera en la solicitud de host a host en una llamada de procedimiento remoto.

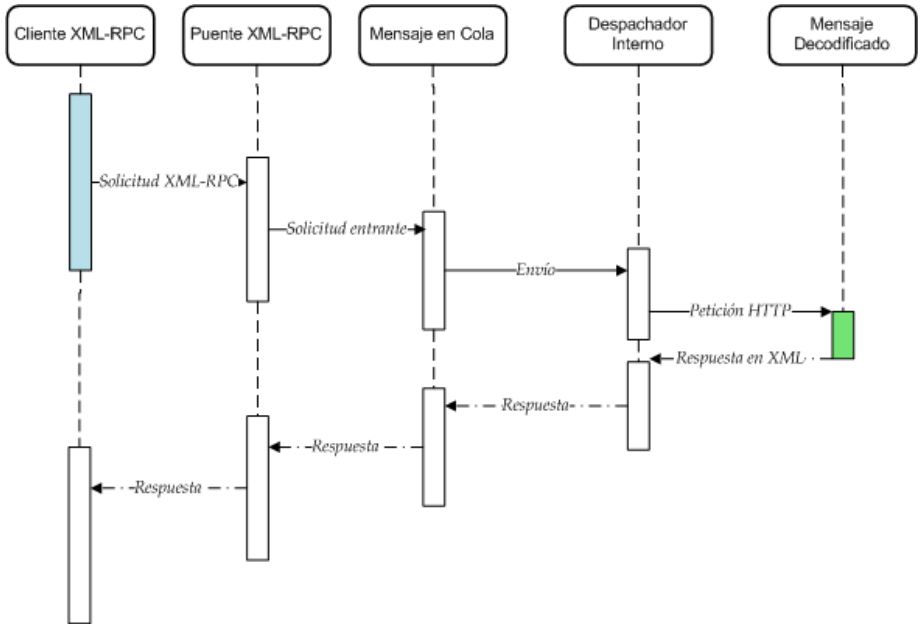


Figura 4.5 Diagrama de Interacción de Objetos de Llamada Remota al Servidor

La Figura 4.6 muestra con mayor detalle el proceso de construcción de una solicitud XML-RPC que consiste en el envío de una petición por parte de Cliente a un servidor, el cual es modelado inicialmente como un objeto de petición XML-RPC. El servidor responde de manera que es modelado mediante un objeto de respuesta XML-RPC. Los parámetros son añadidos a la petición antes de que estos sean enviados al servidor usando los respectivos métodos de objetos que participan en este procedimiento. Por lo que XML-RPC puede ser utilizado para transportar objetos o estructuras tanto como entrada y como parámetros de salida. El método *Submit* de la Petición XML-RPC es utilizado para presentar la solicitud al servidor

remoto, este método también retorna un objeto de respuesta XML-RPC del servidor remoto.

La Figura 4.6 también ilustra el proceso de decodificación de una respuesta, muchos métodos y propiedades se quitan de los diagramas de clases para mayor claridad. Cuando un objeto de respuesta de XML-RPC se retorna desde el método enviar donde la propiedad "*Estado"*, es el Objeto de respuesta XML-RPC que establece un valor que indica como la respuesta ha sido recibida y decodificada.

Si la llamada a procedimiento remoto se ha completado con éxito, los parámetros que devuelve la llamada se almacenan en un objeto denominado Parámetros XML-RPC. Utilizando el método *"Ítem"* del objeto Parámetros XML-RPC, el objeto Valor XML-RPC pueden ser decodificados para obtener la información de la respuesta.

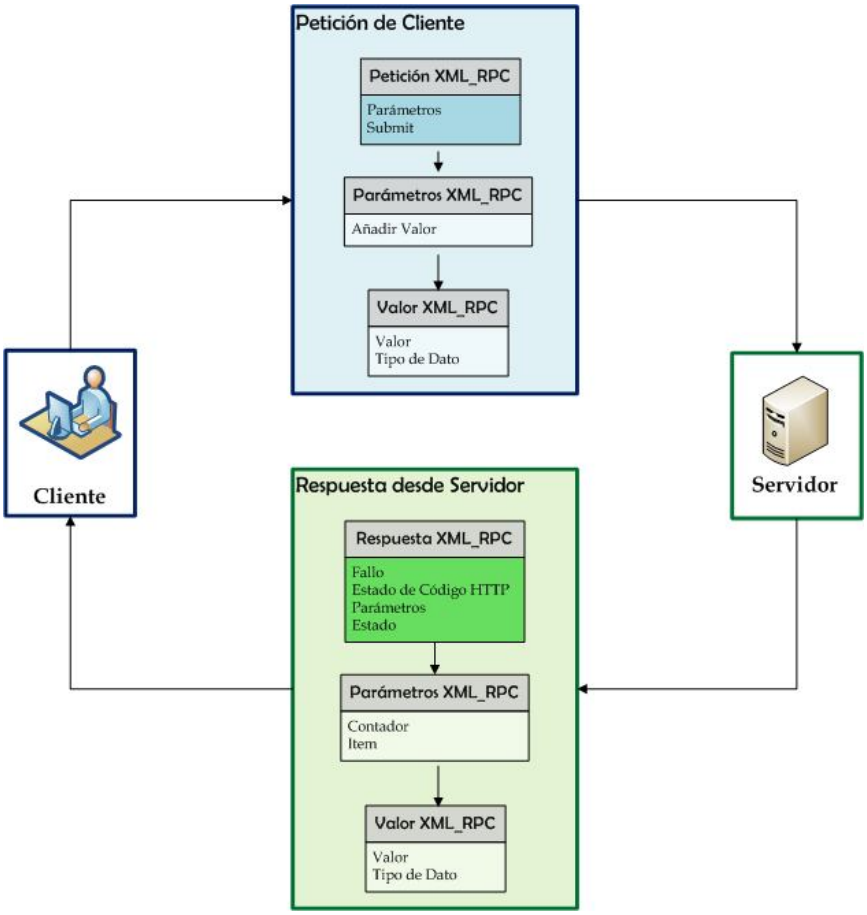


Figura 4.6. Procedimiento de RPC entre Cliente y Servidor

Aplicación Remota para Usuario

JRTAILab

JRTAILab es una pequeña aplicación escrita en lenguaje de programación java, también es conocida como applet; implementa un cliente genérico para ser atendido a través de internet mediante un servidor web que en este caso es RTAI-XML.

Este proyecto es basado en la interfaz gráfica XRtaiLab, y hasta ahora sólo implementa algunas de sus funcionalidades básicas, que permiten al cliente el control remoto de procesos situado en lado del servidor. Ver Figura 4.7

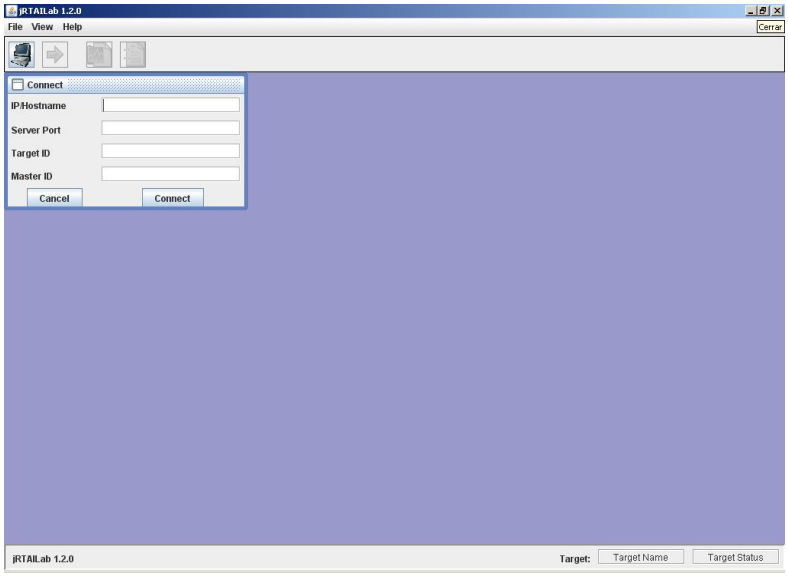


Figura 4.7. Applet de Java- Jrtailab

4.2. Configuración y Levantamiento de un sistema básico para Visualización Remota.

4.2.1 Instalación de RTAI-XML

Cabe señalar que RTAI-XML a partir de la versión 0.9 que es la utilizada en nuestro proyecto, incluye un script de instalación.

Los pasos de instalación de RTAI-XML son simples y fáciles de seguir, a continuación se detalla el proceso de instalación de esta herramienta.

1. Descargar el código fuente de la página

http://artist.dsi.unifi.it/rtaixml

Se recomienda que el lugar donde se va a realizar instalación sea el directorio /opt ya que este directorio es usado generalmente para aplicaciones adicionales que son instaladas manualmente por el usuario. De preferencia realizar los pasos como super-usuario.

2. Descomprimir el archivo el un directorio /opt

cd /opt

tar xvzf rtaixml-0.9.tar.gz

3. Ingresamos en el directorio /rtaixml y ejecutamos el siguiente comando:

cd rtaixml-0.9/

make all

El comando anterior compila todo el código fuente. Este proceso de compilación ha sido probado con la versión de gcc 4.2, pero también debería trabajar con las versiones más reciente sin ningún problema.

4. Ingresamos como administrador, y ejecutamos:

make install

Finalmente se instalará RTAI-XML. Esta herramienta debería funcionar correctamente en la mayoría de sistemas y si se desea personalizar la instalación hay que modificar las primeras líneas del Makefile que contiene información sobre la instalación. Este script es fácil de entender y modificar.

El proceso de instalación crea por defecto el directorio /rtaixml en /usr/realtime/ donde ubica todos los archivos necesarios además de algunos enlaces simbólicos que se los encuentra en /usr/realtime/bin; en este directorio se encuentra rtmanager que es el administrador de las tareas en tiempo real, y rtaixml que es el servidor de tiempo real.

Configuración y Utilización de Rtai-Xml y Herramientas de Visualización.

Existen tres pasos que nos permitirán hacer uso de las herramientas para la visualización remota de una tarea en tiempo real.

1. Configurar el ejecutable objetivo en el script de ejecución como se muestra en la siguiente sección.

2. Iniciar rtmanager, este ejecutable es el encargado de comunicarse a través de XML-RPC y de ejecutar los servidores necesarios RTAI-XML para su gestión, este proceso es completamente tranparente para el usuario.

3. Ejecutar el cliente, es decir JRtaiLab y conectar con el nombre de destino o ejecutable destino, como se especifica en el script de ejecución.

Configuración del Script de destino

Con el fin de registrar el ejecutable objetivo en el dominio RTAI-XML es necesario crear un script único para cada ejecutable de tiempo real. Este archivo debe ser guardado en el directorio /usr/realtime/rtaixml/script con el mismo nombre del ejecutable de tiempo real, es decir, si desea conectarse a su ejecutable objetivo a través de XML-RPC con el nombre MYTARGET, MYTARGET debe ser el nombre del script, el cual debe tener máximo 8 caracteres. La Figura 4.8 nos muestra un script modelo.

```
#HOWTO in brief
#     >> The filename MUST be composed at maximum by 6 chars.
#Uncomment the following line if before executing this target you
# need to start another one defined by the variable depend.
#(This can be used recursively).
#Name of the  target.

#depend=

#[optional] Introduce a delay before target execution. Useful to execute depending targets.
#delay=0.0

#Name of the target executable
target=

#Absolute path
dir=

#[optional] Real time priority
priority=8

#Final time
time=10

#[optional] Starting option. If true the target starts in wait.
#wait=true

#[optional] Verbose output
verbose=true

#[optional] Log file
#log=true

#[optional] Name of the real time Scope
rts=RTS

#[optional] Name of the real time Digital (Led)
rte=RTE
```

Figura 4.8. Script de Configuración de Dominio RTAI-XML

Los parámetros que se muestran en el script son los siguientes:

1. depend: Esta opción le permite iniciar otro destino antes de éste, lo que quiere decir que es posible ejecutar varias tareas de forma recursiva. Este parámetro tiene que fijarse en el nombre de los scripts relacionados a la tarea que desea iniciar.

2. delay: Introduce un retraso después de la orden ejecución en el destino.

3. target: Es el nombre del ejecutable en tiempo real que realiza la tarea.

4. dir: Es la ruta absoluta donde se encuentra el ejecutable de destino.

5. priority: Prioridad en tiempo real del destino en el dominio de RTAI.

6. time: Es el tiempo máximo que permitirá el administrador rtmanager a la tarea ejecutarse.

7. wait: Si es verdadero quiere decir que la tarea comienza en estado de espera.

8. Verbose: Detalla la salida de la tarjeta en la terminal.

9. RTS: Nombre del scope en tiempo real. Debe establecer esta cadena diferente de los otros nombres de destino en tiempo real.

10. rte: Nombre del Led digital en tiempo real.

Conexión con ejecutable de destino

Para iniciar el proceso de visualización remota como primer paso se debe abrir el .jar, el cual nos muestra la ventada de conexión y visualización.

Ubicamos en File-->Connect y se abrirá una pequeña ventana como se muestra en la figura 4.9

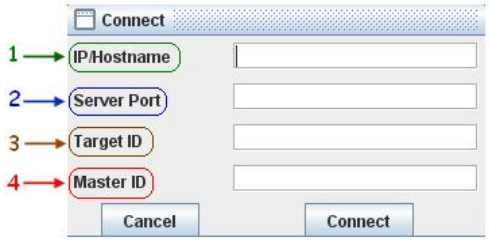


Figura 4.9. Opciones para Conexión en Llamada Remota en Jrtailab

1.- IP: La dirección IP del Servidor, por defecto se señala localhost; es necesario verificar la IP del servidor con la ayuda del comando ifconfig.

2.- Server Port: Es el puerto que se utilizará para enviar/recibir datos. Por defecto es el 29500

3.-Tarjet ID: Es el nombre del Script creado en la Sección anterior para conectarse al destino.

4.-Master ID: Este parámetro debe tener siempre la palabra Master.

4.3 ANÁLISIS TÉCNICO

4.3.1. Identificación de Necesidades.

En las carreras técnicas, la práctica de experimentos con plantas reales es indispensable para consolidar los conceptos teóricos. Sin embargo,

debido a diferentes factores, los estudiantes no tienen acceso libre a los laboratorios. Ventajosamente, las nuevas tecnologías orientadas al trabajo en red, pueden ser utilizadas para mejorar la accesibilidad del estudiante a los experimentos. Este capítulo realiza el análisis técnico del desempeño, manejo y funcionalidad de una de las herramientas utilizada dentro del ámbito de la Ingeniería de Control con la intención de que estas prácticas puedan extenderse a otras áreas del conocimiento.

La herramienta de visualización remota JRtailab posee aspectos que fueron estudiados y puestos a prueba durante la realización de este proyecto, con la finalidad de mostrar las ventajas y limitaciones del uso de esta herramienta para control remoto particularmente de plantas industriales.

La metodología empleada en este análisis técnico fue basada en tres criterios que detallamos a continuación:

·Rendimiento y Visualización de las Señales Transmitidas de Forma remota.

·Medición de Velocidad de respuesta de la Herramienta en relación a la planta.

·Procesamiento gráfico de la Herramienta y medición del tiempo de respuesta del Sistema ante cambios en los valores de los parámetros de la planta.

4.3.2 Evaluación del Uso de la herramienta en los Sistemas Ubuntu, Fedora y Centos.

Una vez identificados los criterios de análisis se procedió a someter a pruebas técnicas Jrtailab con el fin de medir el desempeño y los aspectos cuantitativos y cualitativos tanto lógicos como gráficos que ésta herramienta presenta.

En este caso, se hizo uso de Scicos para el diseño y obtención de los ejecutables, Linux RTAI para la implementación de la aplicación en tiempo real, COMEDI para los *drivers* de la tarjeta de adquisición, así como software específico (Rtai-XML + Jrtailab) para conseguir una arquitectura cliente / servidor que posibilite el acceso vía web al sistema a controlar.

4.3.3 Procedimiento experimental para las pruebas de la Herramienta

Se realizaron varios experimentos en los tres sistemas operativos con diferentes valores de muestreo, con los que se obtuvieron resultados sobre el rendimiento de la herramienta en el lado del cliente y del servidor como también el rendimiento de la planta con la que se ha trabajado en este proyecto.

Los primeros ensayos fueron efectuados con un ejemplo básico cuyo esquema realizado fue creado en Scicos haciendo uso de la paleta de

bloques Rtai-Lib para la construcción del diagrama de bloques como se muestra en la Figura 4.10; se ejecutó las pruebas a diferentes tiempos de muestreo.

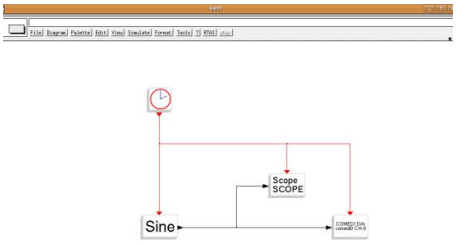


Figura 4.10 Diseño sencillo de onda sinusoidal

Las siguientes pruebas se realizaron con el mismo ejecutable de tiempo real a tres diferentes tiempos de muestreo como se muestra en la Tabla VII con el modelo de la planta de nivel de fluido con lazo de retroalimentación con los valores de filtros y PID previamente determinados para un punto de operación alrededor de los 20 centímetros de altura del fluido; se utilizó un computador que posee la tarjeta de adquisición de datos que calificaremos como servidor RTAI-XML conectado a la planta y el cliente que es otro host en este caso una laptop con acceso a internet y sistema operativo Windows 7 que tenía la aplicación Jrtailab. Observar Figura 4.11

Experimento	Frecuencia de Muestreo (Hercios)
1	100
2	1000
3	10000

Tabla VII. Frecuencias de Muestreo de pruebas realizadas

Figura 4.11 PLanta de Tanque de Agua

En la Figura 4.12 puede observarse el entorno de usuario para el control del nivel de agua del sistema. Además, de forma paralela a esta aplicación se utiliza una cámara IP y micrófonos/altavoces dispositivos con los que cuenta el Laboratorio de Control para que el usuario remoto pueda percibir información del laboratorio.

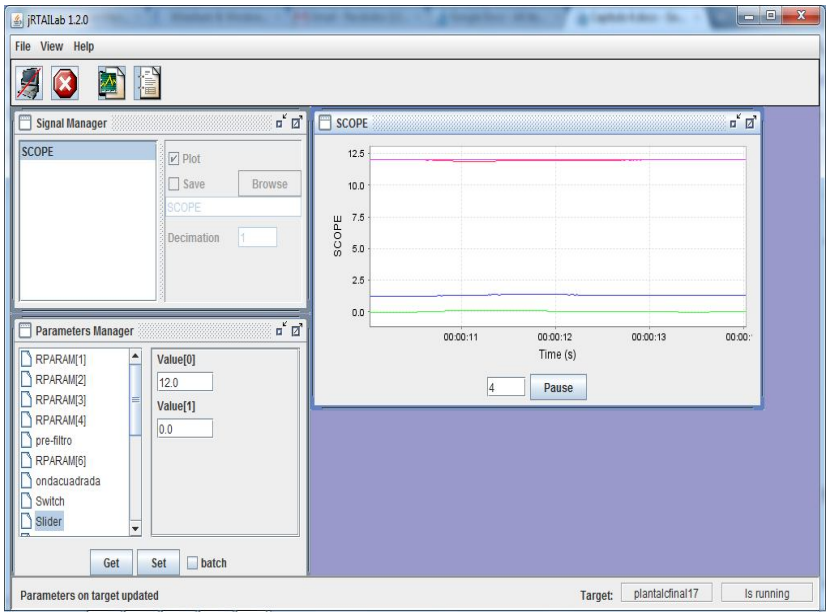


Figura 4.12 Control de Planta de Agua con Jrtailab

4.3.4 Resultados de Pruebas Realizadas

Resultados de Experimentación obtenidos del diseño básico de adquisición en tiempo real de una señal Sinusoidal.

Se realiza un análisis de la respuesta de la aplicación en las primeras pruebas para la generación de la señal a distintos tiempos de muestreo

cuya ejecución es controlada de forma remota a través de la aplicación Jrtailab.

La Tabla VIII nos indica los distintos tiempos de muestreo en los que se realizó la generación de la señal sinusoidal durante las primeras pruebas.

Experimento	**Periodos de Muestro**
1	0,1
2	0,01
3	0,001
4	0,0005
5	0,0007
6	0,0009
7	0,0001

Tabla VIII. Tiempos de Muestreo empleados en diseño Básico

La definición de las clases de desempeño gráfico de aplicación JRtaiLab en la ejecución del experimento básico de tiempo real la muestra la Tabla IX a continuación.

Calidad de Desempeño Gráfico	**Valor**
Excelente	6
Muy bueno	5
Buena pero con retraso al inicio de visualización	4
Graficación Lenta o Distorsionada	3
Con retrasos o segmentación	2
Mala Visualización	1
Inhibición de la aplicación	0

Tabla IX. Niveles de calidad de desempeño gráfico.

Una vez que se ha establecido la forma que se va a realizar el análisis de cada prueba realizada en el primer experimento, la tabla X muestra el resultado del comportamiento de Jrtailab frente a cada una de las

pruebas cuyo comportamiento en la visualización de la señal es calificado en base a la asignación numérica de la Tabla IX.

Experimento	Nivel de calidad de Visualización	Información
1	0	Es necesario forzar la finalización de la aplicación desde el administrador de tareas.
2	5	Ver Figura 4.13 que permite observar que la señal transmitida es igual que la señal recibida
3	5	Presentó un mínimo retraso en la visualización de la señal al iniciar el Scope en la aplicación Jrtailab.
4	3	La visualización de la por ciertos periodos de tiempo es distorsionada como se muestra en la Figura 4.14
5	3	La señal se visualiza mejor que en experimento 4
6	2	La señal en ciertos periodos de tiempo se observa cortada
7	1	La señal no se observa en la forma correcta que fue transmitida. Ver figura 4.15

Tabla X. Resultados de pruebas

Como se puede apreciar en la Figura 4.13 la señal transmitida a un periodo de muestreo de 0.01 segundos y visualizada en el Servidor a través de Xrtailab, es igual a la recibida y visualizada a través de Jrtailab.

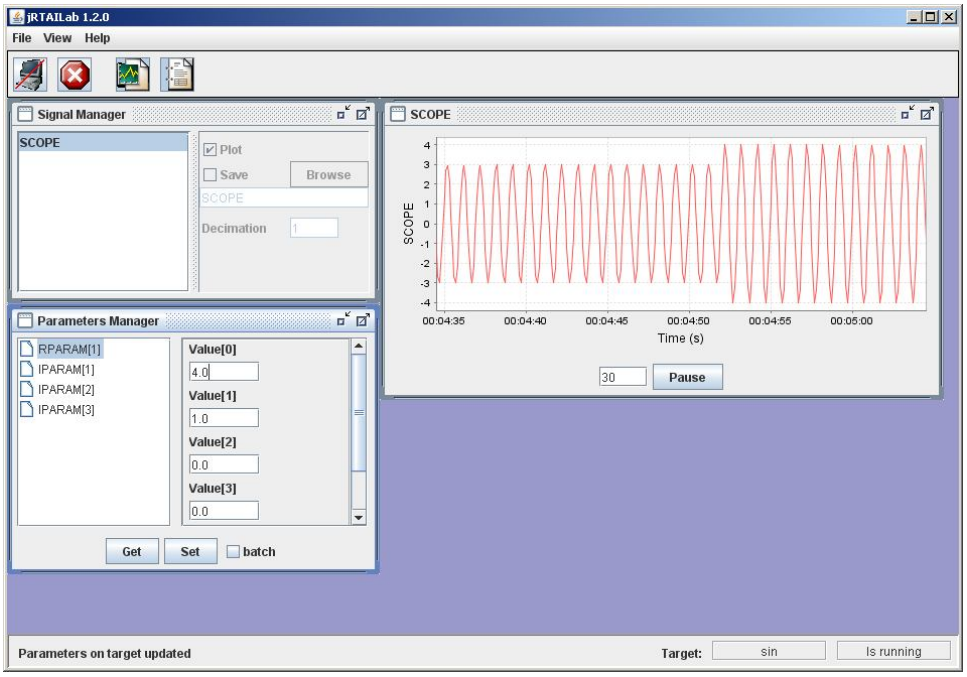


Figura 4.13 Comparación de Señales en Osciloscopio a) Xrtailab b) Jrtailab

La Figura 4.14 se muestra la señal visualizada por el software Jrtailab a un tiempo de muestreo de 0,0005 segundos, como se puede observar la señal se nota entre cortada en ciertos periodos de tiempo.

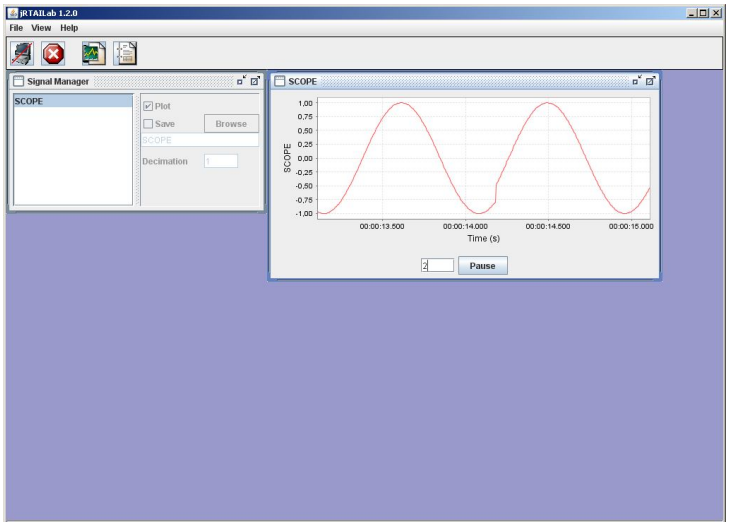


Figura 4.14 Señal distorsionada recibida en Jrtailab

La figura 4.15 muestra la señal a un tiempo de muestreo de 0,0001 segundos, lo que permite notar que la señal no se aprecia bien, es muy diferente a la señal que es transmitida desde el servidor RTAI-XML.

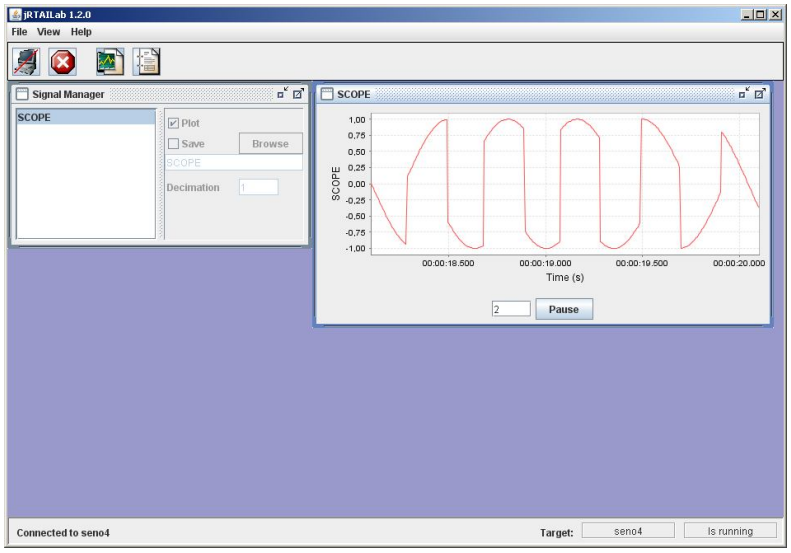


Figura 4.15 Señal transmitida a un periodo de muestreo de 0,0001 seg

Conclusiones:

Los resultados de esta experimentación en su mayoría no fueron exitosos ya que el comportamiento gráfico durante la visualización de señal en las diferentes pruebas realizadas se presentó diferente a la señal original transmitida; por lo que se puede concluir que la mejor frecuencia de muestreo para este experimento fue de 1000 Hz en la generación de la señal, es decir 0.001 segundos de tiempo de muestreo; lo que nos permite señalar que a mayor frecuencia de muestreo la señal presenta errores sistemáticos como distorsión, pérdida de datos durante la adquisición por la aplicación JRtaiLab , por otro lado no es posible observar la señal a frecuencias de muestreo bajos ya que la aplicación JRtaiLab se inhibe , lo que obliga al usuario a cerrar la aplicación de forma forzada.

Un análisis de las posibles causas de dependencia entre la degradación de la señal y la respuesta de control ante retardos y el esfuerzo de control, y visualización para algunos periodos de muestreo es el método de llamadas empleado en los planificadores para cada sistema.

El contador y otras interrupciones del hardware pueden ser también la causa del cambio en el estado de la tarea que está encolada, ya que un problema en las prioridades de las tareas en tiempo real puede ser la principal causa de que los sistemas se inhiban.

Es importante notar que la velocidad de transmisión, el tipo de conexión de red que puede ser a través del Wireless o Ethernet, y la aplicación JRtaiLab en el cliente no son las causas de los problemas presentados ya que todos estos aspectos han sido

estandarizados y probados también en el sistema Ubuntu donde no se presentó ningún problema similar.

Resultados de Experimentación obtenidos del diseño de Planta de Tanque de Agua con Lazo de Retroalimentación.

A continuación se muestra la Tabla XI con la cual se define cuantitativamente las clases de desempeño gráfico con la que se calificará el comportamiento gráfico de un sistema operativo frente al experimento de la Planta de Control de Nivel de Fluido.

Calidad de Desempeño Gráfico	**Valor**
Excelente	6
Muy bueno	5
Bueno y buena con problemas de maximización de ventana	4
Graficación Lenta o Distorsionada	3
Con retrasos o segmentación	2
Sin Visualización	1
Inhibición o finalización de aplicación en el servidor	0

Tabla XI. Vinculación cuantitativa respect a la calidad del desempeño

Una vez establecidos los criterios de calificación en la Tabla XII exponemos los resultados obtenidos durante el primer experimento.

Tiempo de muestreo 0.01 (segundos)	Tiempo de respuesta Servidor (segundos)	Tiempo de respuesta de Visualización Cliente	Tiempo de estabilización en Planta altura 10 cm (segundos)	Tiempo de estabilización en Planta (segundos)	Tiempo de respuesta del cambio (segundos)	Visualización (nivel definición)
UBUNTU	0	9	165	84	9	6
FEDORA	3	8	165	--	--	5
CENTOS	0	10	154	--	--	5

Tabla XII. Experimento 1 con frecuencia de muestreo 100 Hz

Conclusiones

Durante la evaluación de la Herramienta en el sistema operativo en tiempo real Ubuntu los resultados obtenidos fueron mejores que en los Sistemas Fedora y Centos; RTAI-XML en el servidor como Jrtailab en el cliente respondieron con eficiencia a los tres valores de frecuencia de muestreo que se sometió el diseño de la planta el sistema Ubuntu, además como se puede observar en la Tabla XII se

logró controlar con respuestas de procesamiento inmediatas la planta desde el cliente a través de la aplicación Jrtailab.

Como resultado de las pruebas en Fedora, la planta logra estabilizarse a una altura inicial de 10 cm valor que inicialmente en el diseño realizado en Scicos fue modificado, pero no es posible realizar cambios de valor de altura mediante el cliente Jrtailab; al intentar actualizar el valor el motor de la planta instantáneamente se apaga.

Es importante mencionar que es posible también el acceso remoto desde el cliente cuando el proceso en tiempo real ya ha sido ejecutado desde el servidor inicialmente, el servidor RTAI-XML da prioridad a la tarea que se está ejecutando en el Sistema Operativo y si hay una petición enviada al servidor desde el cliente el servidor da prioridad al cliente y si en caso de que desde el servidor se haga una petición para actualizar un valor da prioridad al servidor y así sucesivamente.

La Tabla XIII muestra el resultado del rendimiento del Servidor, Cliente y la Planta de Sistema de Fluido Simple realizada en los tres sistemas con una frecuencia de muestreo de 1 KHz.

Tiempo de muestreo 0.001	Tiempo de respuesta Servidor (segundos)	Tiempo de respuesta Cliente (segundos)	Tiempo de estabilización inicial en Planta (segundos)	Tiempo de estabilización en Planta (segundos)	Tiempo de respuesta del cambio (segundos)	Visualización
UBUNTU	1	7	150	70	3 aprox	6
FEDORA	0	11	147	--	--	5
CENTOS	6	16	194	--	--	5

Tabla XIII. Experimento 2 con frecuencia de muestreo 1000 Hz

Conclusiones:

Como resultado de las pruebas en el Sistema Operativo Centos el comportamiento del servidor, la planta y la aplicación que está siendo utilizada por el usuario (cliente) es el mismo que en la Distribución Fedora, no se logra realizar la modificación en los parámetros del diseño de la planta particularmente el nivel del agua, solo es posible visualizarla con los valores que inicialmente fue ejecutada.

Las distribución Fedora y Centos cuando trabajan a una frecuencia de muestreo de 1k Hz se comportan de la misma manera que al trabajar a una frecuencia de 100 Hz, la mínima diferencia se presenta en los tiempos de estabilización inicial de la planta, ya que tampoco es posible realizar cambios en los valores de los parámetros.

La Tabla XIV muestra el resultado del rendimiento del Servidor, Cliente y la Planta de Sistema de Fluido Simple realizada en los Tres sistemas con una frecuencia de muestreo de 10 KHz.

Tiempo de muestreo 0.0001	Tiempo de respuesta Servidor (segundos)	Tiempo de respuesta Cliente (segundos)	Tiempo de estabilización inicial en Planta (segundos)	Tiempo de estabilización en Planta (segundos)	Tiempo de respuesta del cambio (segundos)	Visualización
UBUNTU	0	20	145	87	3	6
FEDORA	0	10	148	--	--	5
CENTOS	0	7	155	--	--	5

Tabla XIV. Experimento 3 con frecuencia de muestreo 10 KHz

Conclusiones

La distribución Fedora y Centos cuando trabaja a una frecuencia de muestreo de 10k Hz se comporta de la misma manera que al trabajar a una frecuencia de 100 Hz, 1 k Hz, la diferencia se presenta en los tiempos de estabilización de la planta, por lo que tampoco es posible realizar cambios en los valores de los parámetros. Además es importante recalcar que a altas frecuencia el computador, tanto del servidor como el del cliente, quedan inoperables al momento de querer realizar un cambio de valores, lo que no ocurre con las otras frecuencias, por lo que la única solución es volver a reiniciar cada computador.

Los problemas anteriormente mencionados en los sistemas Fedora y Centos pueden ser el resultado de la mala asignación de memoria. Los sistemas embebidos y sistemas de tiempo real suelen tener acceso a dispositivos periféricos. Los registros en el que estos dispositivos colocan sus datos tienen que ser asignados en algún lugar en el espacio de dirección de la tarea de controlador de dispositivo correspondiente. La llamada al sistema POSIX que hace esta asignación es mmap(). Normalmente, esta asignación es una actividad de configuración, y por lo tanto no tiene que ser hecho en tiempo real. Muchas veces esta asignación de memoria y la transmisión de la dirección de un bloque de datos es propenso a errores en estos Sistemas o también en sistemas con multi-procesador, lo que impide al sistema operativo ejecutar las tareas en tiempo real de forma correcta.

CONCLUSIONES Y RECOMENDACIONES

Durante el desarrollo de este proyecto:

1. Se concluye que dado que la herramienta RTAI trabaja a nivel de kernel, un error en el proceso de una tarea podría dejar inoperable el computador en el que se trabaja.
2. Se recomienda verificar dentro de los archivos fuente de RTAI, que exista el correspondiente parche para la versión del kernel que se va a levantar.
3. Se recomienda verificar que exista el soporte de la tarjeta de adquisición de datos a utilizar dentro de la librería de Comedi
4. Se concluye que las herramientas instaladas son capaces de minimizar los retardos y ajustar las prioridades de las tareas.
5. Se recomienda realizar todo el proceso de instalación sin alteración en su orden ya que en caso de ser alterado la instalación no podrá realizar de forma correcta, y por lo tanto las herramientas no cumplirán sus funciones correctamente.
6. Se recomienda que durante el proceso de modificación del kernel se configure la menor cantidad posible de módulos de carga por defecto, de esta forma se agiliza el proceso de carga del sistema.
7. Se concluye que es viable la instalación de las herramientas en los sistemas operativos Ubuntu, Fedora y Centos.
8. Se recomienda no realizar alteraciones en las versiones de las herramientas o de las dependencias, ya que la alteración de la versión

de uno de los paquetes puede producir la necesidad de alterar las versiones de todos los paquetes y herramientas.

9. Se concluye que Ubuntu y Fedora son más cortos que los máximos retardos que puede presentar Centos; también se obtuvo medidas con las que se demuestra que los retardos más cortos que se pueden conseguir, son los retardos en los sistemas operativos de Ubuntu y de Centos. La convergencia de Ubuntu en ambos aspectos, lo define como el sistema que estadísticamente puede presentar retrasos cortos en la atención de una llamada o interrupción que Fedora y Centos.

10. Se concluye que Ubuntu es capaz de trabajar con frecuencias de muestreo superiores a las frecuencias de muestreo máximas con las que se Fedora y Centos pueden trabajar las cuales son alrededor de 10 Kh. Ubuntu muestra un buen desempeño gráfico en casi todas las frecuencias de trabajo hasta alcanzar periodos de muestreo alrededor de los 0.0005 segundos, mientras que los sistemas operativos Fedora y Centos se inhiben al trabajar con periodos de este orden. Fedora muestra eficiencias de trabajo muy cercanas a las de Ubuntu en gran parte de las pruebas, Centos respondió de forma relativamente eficiente solo en el 50 por ciento de las pruebas, en las que el período mínimo de muestreo alcanzado es del orden de los 0.0001.

11. Se recomienda no trabajar con frecuencias de muestreo muy bajas, pese a que la planta a controlar sea lenta, y por lo tanto no se requieran muchos datos, para que la aplicación de control puede hacer un estimado más práctico del comportamiento de la planta.

12. Se concluye que el desempeño más eficiente y óptimo para control de plantas e interacción con el usuario es el presentado por Ubuntu, lo que lo hace el sistema operativo apropiado para realizar operaciones de control en un alto rango de períodos de muestreo de señales. El sistema operativo Fedora presentó eficiencias semejantes a las de Ubuntu en gran parte de las pruebas, lo que hace el según sistema operativo más recomendable. El sistema operativo Centos presentó un desempeño apropiado para algunas frecuencias de muestro relativamente bajas.

13. Teniendo en cuenta que ciertos tipos de plantas poseen lenta respuesta, como es el caso de la planta utilizada para las pruebas, se concluye que el desempeño de cada uno de los sistemas operativos es el suficiente para poder realizar el control de plantas lentas, debido a que éstas no requieren frecuencias de muestreo muy altas y pueden sobrepasar frecuencias de muestreo con las que sistemas operativos como Fedora y Centos son eficientes. Tomando en cuenta estas mismas consideraciones también se concluye que para plantas de respuesta rápida, que requieran sistemas de control con frecuencias de muestreo relativamente altas se debe utilizar el sistema Ubuntu ya que es el sistema que logró funcionar eficientemente con estas frecuencias.

14. Se ha evaluado también el comportamiento de la Aplicación Jrtailab mediante la utilización de varios periodos de muestreo que permitieron medir el rendimiento de esta herramienta. Los experimentos y análisis muestran que Jrtailab nos permite realizar el diseño del control de diferentes plantas de una manera sencilla usando Scicos y poderlo

manejar y visualizar las señales a distancia a través de una interfaz web y así interactuar con un dominio del proceso en tiempo real gestionado por RTAI-XML, donde uno o varios procesos de tiempo real están siendo ejecutados en una o diferentes computadoras y comunicarse a través de una LAN/WAN.

15. Los diferentes resultados que el desempeño de Jrtailab proporciona un balance preestablecido entre precisión y tiempo de respuesta con respecto a un determinado tiempo de muestreo, y trató de mantener el tiempo de respuesta acotado mediante la reducción de la precisión computacional ante una frecuencia de muestreo alta; comportamientos que influirían en el caso de un proceso industrial ya que un retraso mínimo hablando en el orden de los segundos puede representar en el caso de una industria una pérdida económica grande ante la detención de un proceso, y más aún que la planta deje de funcionar.

16. En las pruebas efectuadas con la generación de una onda sinusoidal y con el servidor Rtai-XML instalado en el Sistema RTAI-Ubuntu, a un tiempo de muestreo de 0.01 segundos la señal visualizada en el osciloscopio de Jratilab fue la misma que en el osciloscopio virtual Xrtailab que se encuentra en el servidor RTAI-Ubuntu; a un periodo de muestreo de 0.0001 la señal se observó con mucha distorsión a pesar que el número de muestras tomadas por segundo fueron mucho mayores que 0.01 la señal en el lado del servidor se observa muy clara y sin distorsión, se debe considerar que este problema puede estar ligado a la velocidad de transmisión, y a la métrica que existe entre el cliente y

el servidor hablando en términos de redes, ya que la perdidas de datos en todo este proceso algunas veces es inevitable. Respecto a las pruebas realizadas con la planta, utilizando tres tiempos de muestreo diferentes, se ha podido visualizar eficientemente las gráficas de las señales en Jrtailab, estas señales son transmitidas desde los tres sistemas: Ubuntu, Fedora y Centos de manera individual, pero solo el Sistema Ubuntu permite realizar cambios en los parámetros del esquemático de la planta, ya que al tratar de realizar esta acción en los Sistemas Fedora y Centos; en el caso del experimento utilizado para realizar estas pruebas, cambios controlados en el nivel del agua del tanque; el motor de planta se apaga y no responde ante ningún cambio es decir queda inoperable y los sistemas del cliente y del servidor se inhiben.

17. Jrtailab tiene mucha semejanza a Xrtailab y además es el primer proyecto desarrollado en ser cliente genérico del control de procesos en tiempo real de forma remota. Existen otras dos interfaces de control basadas en Jrtailab como ARTIST y AFM. ARTIST es también una plataforma para control de pruebas remotas de esquemas de procesos en tiempo real, fue desarrollado en una versión modificada de RTAI-XML y soporta consultas de base de datos.

    AFM, Fuerza Atómica Microscópica es una técnica de medición en la que el rendimiento se encuentra principalmente influido por la eficiencia de un circuito de retroalimentación. Los principales usuarios de AFM son los biólogos y médicos.

18. Se recomienda la continuidad de las investigaciones en este proyecto para sentar las bases de migración a sistemas de código abierto en el área de control de sistemas industriales. Las herramientas detalladas en este proyector pueden ser empleadas tanto en el campo laboral como en el campo educativo y de investigaciones. Proyectos como transmisión de lecturas de señales eléctricas del cuerpo humano, control remoto de maquinaria, monitoreo remoto de lecturas como presión y temperatura, pueden ser desarrollados mediante la continua investigación de este tipo de herramientas.

BIBLIOGRAFIA

[1] WIKIPEDIA, *Tiempo Real,*http://es.wikipedia.org/wiki/Tiempo_real. Última actualización: 18 Agosto 2010.

[2] LÓPEZ ZAMARRÓN D., *Análisis de Sistemas Operativos de Tiempo Real Libres,* Universidad Politécnica de Madrid, http://gayuba1.datsi.fi.upm.es/~dlopez/cache/doc/sotr.pdf, 2004.

[3] YARMOUK UNIVERSITY, *Requerimientos de Tareas de Tiempo Real*, http://faculty.yu.edu.jo/halzoubi/DownloadHandler.ashx?pg=24fc7f00-0bf6-43ed-8bb0-8faac491ae13&section=4dc59155-70a0-4123-b10d-4ef4f53839bf&file=L4.pdf.,2009.

[4] NATIONAL INSTRUMENTS, *PCI-6023E/6024E/6025E User Manual* , http://www.ni.com/pdf/manuals/322072a.pdf, October 1998

[5] LÓPEZ ZAMARRÓN D., *Diseño de un Rótulo Luminoso con fines Docentes,* Trabajo Fin de Carrera, Segovia, Universidad Politécnica de Madrid, Junio del 2005.

[6] BOVET, D. J. - CESATI M., *Understanding The Linux Kernel* (22-25), First Ed., San Diego: O'Reilly, 2000.

[7] WIKIPEDIA, *Núcleo Linux*, http://es.wikipedia.org/wiki/Núcleo_Linux. Última actualización: 20 de Diciembre del 2010

[8] GRUPO DE DESARROLLO DE MESA 3D, *The Mesa 3D Graphics Library*, http://www.mesa3d.org/, Octubre 2010

[9] EDE TEAM, *EQUINOX DESKTOP ENVIRONMENT*, http://equinox-project.org/, 2010

[10] DAVID SCHLEEF - FRANK MORI HEES - IAN ABBOTT, *Comedi*, http://www.comedi.org/doc/, 2009

[11] DAVID SCHLEEF - FRANK MORI HEES - IAN ABBOTT,*Comedi- Gerarquía de Dispositivos*, http://www.comedi.org/doc/index.html#COMEDIDEVICES, 2009

[12] WIKIPEDIA, *Scilab*, http://es.wikipedia.org/wiki/Scilab. Última Actualización: 6 de Diciembre del 2010

[13] THOMAS NETTER - INRIA, *Scicos: Block diagram modeler/simulator*, http://www-rocq.inria.fr/scicos/, Última Actualización: 2009

[14] HOLGER NAHRSTAEDT - ANDREAS VIKLUND, *QRTAILAB - Installation RTAI*, http://qrtailab.sourceforge.net/rtai_installation.html, 2008-2009

[15] MAASSE J., *Scicos as an alternative for Simulink, Migrating from Simulink to Scicos with respect to real time programs,* Eindhoven, Technische Universiteit Eindhoven, Mayo del 2006

[16] WIKILEARNING, *Compilación del kernel paso a paso,* http://www.wikilearning.com/tutorial/compilacion_del_kernel_paso_a_paso-el_proceso_de_configuracion/876-3, Consultado en Octubre 2010.

[17] BUCHER R., MANNORI S., NETTER T., *RTAI-Lab tutorial: Scilab, Comedi, and real-time control*, http://www.dti.supsi.ch/~bucher/pdf/RTAI-Lab-tutorial.pdf, Marzo del 2009

[18] LÓPEZ ZAMARRÓN D., *Análisis de Sistemas Operativos de Tiempo Real Libres,* http://gayuba1.datsi.fi.upm.es/~dlopez/cache/doc/sotr.pdf, 2004.

[19] PÁGINA DEL PORTAL DE INGENIERÍA QUÍMICA, *Tutorial de Scilab,* http://www.ingenieriaquimica.org/system/files/TutorialDeScilab.doc, 2010

[20] HOLGER NAHRSTAEDT - ANDREAS VIKLUND, *QRTAILAB - Performance*, http://qrtailab.sourceforge.net/performance.html, 2008-2009

[21] GRUPO RTAI, *RTAI-XML*, http://www.rtaixml.net/, 2010

[22] BASSO M., VASSALI M, RTAI-*XML: A Web Services Approach to Real-Time Control Systems,* Universit_a degli Studi di Firenze, Julio del 2009.

[23] TEXTOS CIENTÍFICOS, *RPC Llamada a procedimiento remoto*, http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte/rpc, Consultado en Octubre 2010

APENDICE A

Manual de Instalación de RTAI en Ubuntu 8.04

1. INTRODUCCIÓN

En este manual de instalación se detalla de forma clara y sencilla el proceso de instalación para convertir un sistema operativo de uso general, en uno de tiempo real utilizando la herramienta RTAI; se seguirá un determinado grupo de instrucciones para reducir la complejidad que presenta el proceso de instalación de RTAI en una Distribución Linux; particularmente Ubuntu 8.04.

En este documento se describen los conceptos básicos de paquetes, conocimientos que le ayudarán a entender de mejor manera el procedimiento de instalación. Es de suma importancia que se realice con anterioridad al proceso de instalación, la revisión del soporte del software que se requiere para el control del hardware, en este caso haremos uso de la tarjeta de adquisición de datos de National Instruments PCI-6024E.

También se detalla la instalación del programa de análisis y diseño Scilab, el cual contiene la herramienta para elaboración de esquemáticos Scicos. Ambas herramientas pueden ser utilizadas para la elaboración de programas personalizados para el control y lectura de señales eléctricas, analógicas o digitales en el hardware soportado por la librerías, en este caso Comedi.

Además se agregará una pequeña aplicación del sistema instalado para el control automático de un sistema de fluido simple.

DEPENDENCIAS NECESARIAS

Es fundamental realizar la instalación de todos y cada uno de los paquetes requeridos en el proceso de Instalación. Estas librerías juegan un papel primordial tanto en compilación e instalación del nuevo Kernel, como también en los Sistemas de control de versiones, herramientas de desarrollo, herramientas gráficas y librerías para GNU.

A continuación se detallan los paquetes previos instalados en la Distribución Ubuntu para la Instalación de RTAI.

Tabla de Dependencias Previas

Cada paquete detallado a continuación debe ser instalado como superusuario, estando en la terminal desde cualquier carpeta del sistema.

General		
Software	**Comando**	**Dependencias**
cvs	apt-get install cvs	ninguna
svn	apt-get install subversion	libapr1 libaprutil1 libpq5 libsvn1
build-essential	apt-get install build-essential	build-essential dpkg-dev g++ g++-4.2 libc6-dev libstdc++6-4.2-dev libtimedate-perl linux-libc-dev patch
checkinstall	apt-get install checkinstall	ninguna
Kernel		
Software	**Comando**	**Dependencias**
libncurses5-dev	apt-get install libncurses5-dev	ninguna

kernel-package	apt-get install kernel-package	intltool-debian kernel-package po-debconf
fakeroot	apt-get install fakeroot	ninguna
Rtai		
Software	**Comando**	**Dependencias**
libxmu-dev	apt-get install libxmu-dev	libxmu-dev libxmu-headers
libxi-dev	apt-get install libxi-dev	ninguna
doxygen	apt-get install doxygen	ninguna
autoconf	apt-get install autoconf	ninguna
automake	apt-get install automake	autoconf automake autotools-dev m4
libtool	apt-get install libtool	ninguna
Comedi-Lib		
Software	**Comando**	**Dependencias**
bison	apt-get install bison	ninguna
flex	apt-get install flex	ninguna

python-dev	apt-get install python-dev	python2.5 python2.5-minimal
Comedi-Calibrate		
Software	**Comando**	**Dependencias**
libboost-program-options-dev	apt-get install libboost-program-options-dev	libboost-dev libboost-program-options-dev libboost-program-options1.34.1
libgsl0-dev	apt-get install libgsl0-dev	ninguna
Scilab 4.1.2		
Software	**Comando**	**Dependencias**
g77	apt-get install g77	cpp-3.4 g77 g77-3.4 gcc-3.4 gcc-3.4-base libg2c0 libg2c0-dev
gfortran	apt-get install gfortran	gfortran gfortran-4.2 libgfortran2 libgmp3c2 libmpfr1ldbl
sablotron	apt-get install sablotron	libsablot0
tcl8.4	apt-get install tcl8.4	libc6

tk8.4	apt-get install tk8.4	libc6 libx11-6
tcl8.5-dev	apt-get install tcl8.5-dev	ninguna
tk8.5-dev	apt-get install tk8.5-dev	ninguna
xaw3dg-dev	apt-get install xaw3dg-dev	libxpm-dev xaw3dg xaw3dg-dev
libpvm3	apt-get install libpvm3	ninguna
libgtkhtml2-dev	apt-get install libgtkhtml2-dev	libatk1.0-dev libcairo2-dev libexpat1-dev libfontconfig1-dev libgail-dev libglib2.0-dev libgtk2.0-dev libgtkhtml2-dev libpango1.0-dev libpixman-1-dev libpng12-dev libxcomposite-dev libxcursor-dev libxdamage-dev libxfixes-dev libxft-dev libxinerama-dev libxml2-dev libxrandr-dev libxrender-dev x11proto-composite-dev x11proto-damage-dev x11proto-fixes-dev x11proto-randr-dev x11proto-render-dev x11proto-xinerama-dev (Actualizar: libexpat1 libglib2.0-0 libxml2)
libzvt-dev	apt-get install libzvt-dev	gdk-imlib11 imlib-base libgif4 libglib1.2-dev libglib1.2ldbl

		libgtk1.2 libgtk1.2-common libgtk1.2-dev libzvt-dev libzvt2
libvte-dev	apt-get install libvte-dev	ninguna
Scilab 5.x (Adicionales a las anteriores)		
Software	**Comando**	**Dependencias**
swig	apt-get install swig	ninguna
pvm-dev	apt-get install pvm-dev	libreadline5-dev pvm
gettext	apt-get install gettext	ninguna
MesaLib		
Software	**Comando**	**Dependencias**
x11proto-xext-dev	apt-get install x11proto-xext-dev	ninguna
xlibs-static-dev	apt-get install xlibs-static-dev	libfontenc-dev libfreetype6-dev libxfont-dev x11proto-fonts-dev zlib1g-dev
libxext-dev	apt-get install libxext-dev	ninguna

libxext-dev	apt-get install libxt-dev	ninguna
libglu1-mesa-dev	apt-get install libglu1-mesa-dev	ninguna
QRtaiLab		
Software	**Comando**	**Dependencias**
libqt4-dev	apt-get install libqt4-dev	comerr-dev libaudio-dev libaudio2 libjpeg62-dev libkadm55 libkrb5-dev liblcms1-dev libmng-dev libmysqlclient15off libpq-dev libqt4-core libqt4-dev libqt4-gui libqt4-qt3support libqt4-sql libsqlite0-dev libssl-dev mysql-common (Actualizar: libkrb53 libssl0.9.8)
libqwt5-qt4-dev	apt-get install libqwt5-qt4-dev	libqwt5-qt4

- **INSTALACIÓN RTAI**

Una vez instaladas las dependencias, proceder a instalar RTAI en Ubuntu versión 8.04.

.1 Descarga de Instaladores comprimidos

-Descargar: Mesa3D de la siguiente dirección:

```
http://sourceforge.net/projects/mesa3d/files/MesaLib/7.0.3/MesaLib7.0.3.tar.gz/download
```

-Copiar el archivo MesaLib-7.0.3.tar.gz a la carpeta /usr/local/src.

```
cp  MesaLib-7.0.3.tar.gz /usr/local/src
```

-Descargar el archivo comprimido de RTAI desde la pagina web, especificando que no se realice validación del certificado del servidor (HTTPS; SSL,TLS).

```
wget --no-check-certificate https://www.rtai.org/RTAI/rtai-3.6-cv.tar.bz2
```

-Copiar rtai-3.6-cv.tar.bz2 en la carpeta /usr/src/.

```
cp rtai-3.6-cv.tar.bz2 /usr/src/
```

-Descargar el archivo comprimido del kernel de Linux seleccionado.

```
Wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.24.7.tar.bz2
```

Copiar el archivo linux-2.6.24.7.tar.bz2 en la carpeta /usr/src/.

```
cp linux-2.6.24.7.tar.bz2 /usr/src/
```

.2 Instalación Mesa3D

Mesa3D es un sistema Open Source de procesamiento interactivo de gráficos 3D implementado con soporte de OPENG,L la cual es una librería gráfica Open Source que abstrae un conjunto de instrucciones y brinda un conjunto de funciones y procedimientos para la creación de aplicaciones que permitan generar gráficos en 2 y en 3 dimensiones.

Descripción del Proceso de Instalación

-Cambiarse al directorio /usr/local/src.

```
cd /usr/local/src
```

-Descomprimir el archivo MesaLib-7.0.3.tar.gz.

```
tar xvf MesaLib-7.0.3.tar.gz
```

-Ingresar a la carpeta que se crea Mesa-7.0.3.

```
cd Mesa-7.0.3
```

-Hacer una limpieza de la carpeta para asegurar de que se produzca un buen levantamiento de configuración.

```
make realclean
```

-Configurar las variables de configuración para Linux con optimización x86.

```
make linux-x86-static
```

-Instalar el programa con las configuraciones indicadas en el paso anterior.

```
make install
```

.3 Instalación de EFLTK

EFLTK es una herramienta desarrollada en base de librerías de software libre y de código abierto, es una extensión de la librería FLTK. Esta biblioteca es una mejora de FLTK 2.0, con muchos cambios y modificaciones fundamentales como: soporte para XML, red, y base de datos. Para compilarlo, no es necesario el código fuente FLTK 2.0, eFLTK ya lo incluye.

La biblioteca EFLTK, soporta varios sistemas operativos: Windows/ Unix / Mac, entre otros, es de licencia LGPL (Licencia Pública General Reducida de GNU); proporciona moderna funcionalidad GUI, soporte en gráficos a través de OpenGL que es el primer ambiente para desarrollar aplicaciones portables, que muestren gráficos 2D y 3D, integra una emulación de GLUT (OpenGL Utility Toolkit).

Descripción del Proceso de Instalación

-Cambiar al directorio /usr/local/src

```
cd /usr/local/src
```

-Descargar: Por svn EFLTK con el siguiente comando:

```
svn co https://ede.svn.sourceforge.net/svnroot/ede/trunk/efltk
```

-Descargar: Por svn EDE con el siguiente comando:

```
svn co https://ede.svn.sourceforge.net/svnroot/ede/trunk/ede
```

-Ingresar a la carpeta efltk

```
cd efltk
```

-Levantar los archivos de configuración de acuerdo con la distribución y las configuraciones que tengamos hasta ahora.

```
Autoconf
```

-Realizar la configuración omitiendo el soporte para MySQL y para unixODBC, ambos administradores de bases de datos

```
./configure --disable-mysql --disable-unixODBC
```

-Construir la librería EFLTK

```
./emake
```

-Instalar la librería

```
./emake install
```

-Colocar una ruta extra al archivo de configuración de carga de la base de datos de las librerías, para que librerías también sean buscadas en la dirección /usr/local/lib.

```
echo "/usr/local/lib" >> /etc/ld.so.conf
```

-Hacer la carga de los módulos instalados por los procesos anteriores

```
/sbin/ldconfig
```

.4 Configuración del Kernel

-Cambiar al directorio /usr/src/

```
cd /usr/src/
```

-Descomprimir el archivo rtai-3.6-cv.tar.bz2

```
tar xjvf rtai-3.6-cv.tar.bz2
```

-Crear un vínculo para facilitar el acceso a la carpeta que produjo la anterior descompresión. Nombre de vínculo: rtai

```
ln -s rtai-3.6-cv rtai
```

-Descomprimir el Kernel descargado en el paso anterior

```
tar xjvf linux-2.6.24.7.tar.bz2
```

-Es recomendable cambiar el nombre de la carpeta que se creó con el paso anterior, para poder identificar fácilmente al kernel en el que vamos a trabajar.

```
mv /usr/src/linux-2.6.24.7 /usr/src/linux-2.6.24.7-rtai
```

-Crear un vínculo para un fácil acceso a la carpeta. El nombre del vínculo es Linux.

```
ln -s linux-2.6.24.7-rtai linux
```

-Ingresar a la carpeta del kernel

```
cd /usr/src/linux
```

-Parchar la versión del kernel descargado con el parche ubicado dentro de las carpetas de Rtai.

```
patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-linux-
2.6.24-x86-2.0-07.patch
```

-Copiar el archivo de configuración del kernel que ya está corriendo en nuestro sistema y que fue levantado automáticamente durante la instalación de Ubuntu.

```
cp /boot/config-2.6.24-24-generic .config
```

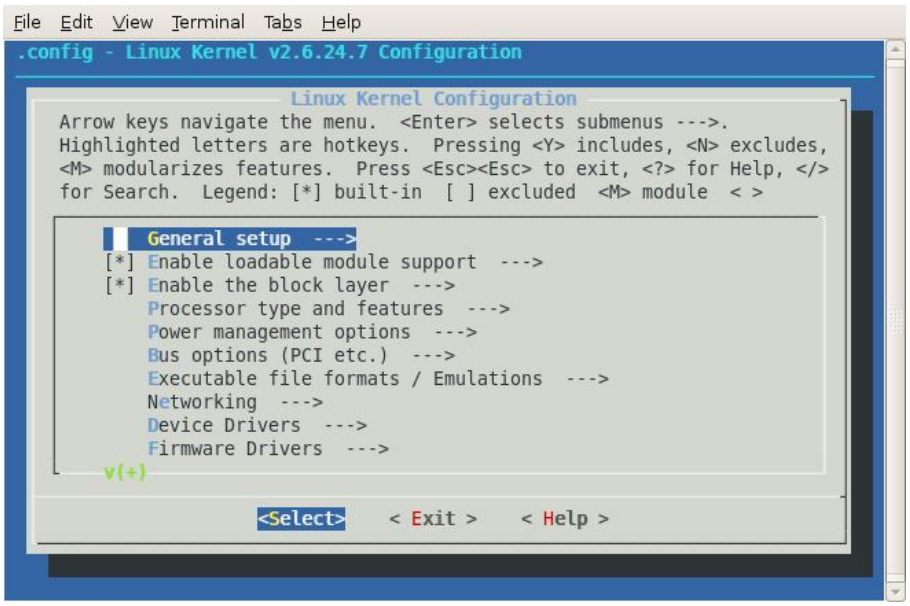
-Configurar el nuevo archivo de configuración del Kernel para adecuarlo con las instrucciones del archivo de configuración del Kernel levantado por Ubuntu

```
make oldconfig
```

-Levantar la configuración necesaria para adecuar al nuevo Kernel de Linux como Kernel de sistema en tiempo real

```
make menuconfig
```

Este comando mostrará en la terminal un menú como se indica en la siguiente figura que nos permitirá elegir los parámetros que ayuden a ajustar nuestro kernel para que realice las tareas en tiempo real.



Parámetros de Configuración del Kernel

Se debe verificar el tipo de procesador y los dispositivos que posee el computador en el cual se va a realizar la instalación para elegir los correctos parámetros, caso contrario la instalación no se realizará con éxito.

Los siguientes comandos le permitirán realizar las recomendaciones anteriormente descritas:

lspci : Lista los dispositivos pci del computador.

cat /proc/cpuinfo : Muestra la información del procesador.

General setup

-Local version - append to kernel release: -rtai

Enable loadable module support

-Module versioning support : deshabilitado

-Source checksum for all modules: habilitado

Processor type and features

-Subarchitecture Type: PC-compatible

-Proccesor family: Seleccionar el procesador que posee el computador

-Preemption Model: Preemptible Kernel (Low-Latency Desktop)

-Interrupt pipeline : habilitado

-Toshiba Laptop support: deshabilitado

-Dell Laptop support: deshabilitado

-High Memory Support: off

-64 bit Memory and IO resources (EXPERIMENTAL): deshabilitado para pc 32 bits

---

Power Management support: deshabilitado

Bus Options (PCI etc.)

-PCCard (PCMCIA/cardBus) support : deshabilitado

Networking

-Networking options

-The IPV6 protocol: deshabilitado

-The DCCP Protocol (Experimental): deshabilitado

-The TIPC Protocol (Experimental): deshabilitado

-Asynchronous Transfer Mode (ATM) (Experimental): deshabilitado

-The IPX Protocol: deshabilitado

-Appletalk protocol support: deshabilitado

-WAN router: deshabilitado

-Amateur Radio support: deshabilitado

-IrDA (infrared) subsystem support: deshabilitado

-Bluetooth subsystem support: deshabilitado

-RxRPC session sockets: deshabilitado

-Wireless: deshabilitar todo el contenido interno

-RF switch subsystem support: deshabilitado

-Plan 9 Resource Sharing Support (9P2000)(Experimental): deshabilitado.

Device Drivers

-Memory Technology Device (MTD) support: deshabilitado

-Parallel port support: deshabilitado

-Misc devices:deshabilitado

-I2O device support: deshabilitado

-Macintosh device drivers :deshabilitado

-Network device support

    -Token Ring driver support: deshabilitado

    -Wireless Lan: deshabilitar todo el contenido interno

    -USB Network Adapters: deshabilitar todo el contenido interno

    -FDDI: deshabilitado

-ISDN support: deshabilitado

-Telephony support: deshabilitado

-Input device support

    -Joystick interface: deshabilitar

    -Joysticks/Gamepads: deshabilitado

    -Tablets: deshabilitado

    -Touchscreens:deshabilitado

-I2C support:deshabilitar todo el contenido interno

-Multimedia devices

-Video for linux

Radio Adapters: deshabilitado

-DAB adapters:deshabilitado

-Graphics support

-/dev/agpart (AGP Support) : habilitado

-Direct Rendering Manager (XFree86 4.1.0 and higher ...: habilitado

-Lowlevel video output switch controls: deshabilitado

-Support for frame buffer devices: deshabilitado

-Backlight & LCD device support: deseleccionar todo el contenido interno

-Display device support: deseleccionar todo el contenido interno

-Console display driver support: deseleccionar todo el contenido interno

-Sound

-Sound card support: deshabilitado

-USB support

-USB Modem (CDC ACM) support: deshabilitado

-USB Printer support: deshabilitado

-Siemens ID USB Mouse Fingerprint sensor support: deshabilitado

-Apple Cinema Display support: deshabilitado

-USB Serial Converter support: deshabilitado

-USB DSL modem support: deshabilitado

-Virtualization: deshabilitado

File systems

-Second extended fs support: habilitado

-Ext3 journalling file system support: habilitado

-Ext3 extended attributes: habilitado

-Reiserfs support: deshabilitado

-CD-ROM/DVD Filesystems: seleccionar todo el contenido interno

- Luego de seleccionar los parámetros construimos los paquetes del Kernel nuevo con los siguientes comandos:

```
make
```

-Instalar los módulos levantados por el nuevo Kernel

```
make modules_install
```

-Instalar los paquetes levantados por el nuevo Kernel

```
make install
```

-Copiar el firmware del kernel original con el nombre del nuevo Kernel

```
cp -aR /lib/firmware/2.6.24-24-generic /lib/firmware/2.6.24.7-rtai
```

-Crear la imagen de arranque del nuevo kernel

```
update-initramfs -c -k 2.6.24.7-rtai
```

-Editar el archivo de configuración del gestor de arranque

```
gedit /boot/grub/menu.lst
```

-Colocar el nuevo kernel en las opciones de arranque, con su respectiva maquina e imagen

-Reiniciar el computador y seleccionar el nuevo kernel en el gestor de arranque.

.5 Instalación de Comedilib

Comedilib es una biblioteca en espacio de usuario que proporciona una interfaz amistosa con el desarrollador de dispositivos Comedi.

En la distribución Comedilib encontramos la documentación, configuración y calibración de los servicios de acceso público y programas de demostración. Esta librería viene incluida en las distribuciones Linux.

Descripción del proceso de instalación.

-Cambiar al directorio /usr/local/src

```
cd /usr/local/src
```

-Iniciar una sesión cvs para la descarga de comedi

```
cvs -d :pserver:anonymous@cvs.comedi.org:/cvs/comedi login
```

-Descargar comedi usando cvs

```
cvs -d :pserver:anonymous@cvs.comedi.org:/cvs/comedi co comedi
```

-Descargar comedilib usando cvs

```
cvs -d :pserver:anonymous@cvs.comedi.org:/cvs/comedi co comedilib
```

-Ingresar a la carpeta descargada de comedilib

```
cd comedilib
```

-Generar los archivos de levantamiento de configuración

```
sh autogen.sh
```

-Levantar el archivo de configuracion

```
./configure --sysconfdir=/etc
```

-Generar los archivos de instalación

```
make
```

-Realizar la instalación de los paquetes configurados en los pasos anteriores

```
make install
```

-Generar los módulos de la tarjeta de adquisición dentro de los dispositivos

```
make dev
```

.6 Instalación Rtai parte 1

En esta sección realizamos la instalación de RTAI, lo que nos permitirá ejecutar tareas en tiempo real.

-Ir a la carpeta /usr/src/rtai/

```
cd /usr/src/rtai
```

-Verificar los valores del menú de Rtai

-Ejecutar el menú de la consola de configuración de Rtai

```
make menuconfig
```

-Generar los archivos configurados en el paso anterior

```
make
```

-Realizar la instalación de la configuración actual de Rtai

```
make install
```

-Incluir la ruta /usr/realtime/bin en la variable global PATH

```
echo "PATH=$PATH:/usr/realtime/bin" >> /root/.bashrc
```

-Incluir la exportación del paso anterior en el script de arranque

```
echo "export PATH" >> /root/.bashrc
```

-Reiniciar el computador y seleccionar el kernel de RTAI

Prueba del Progreso de la instalación de RTAI

-Levantar los modulos de la rtai

```
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
```

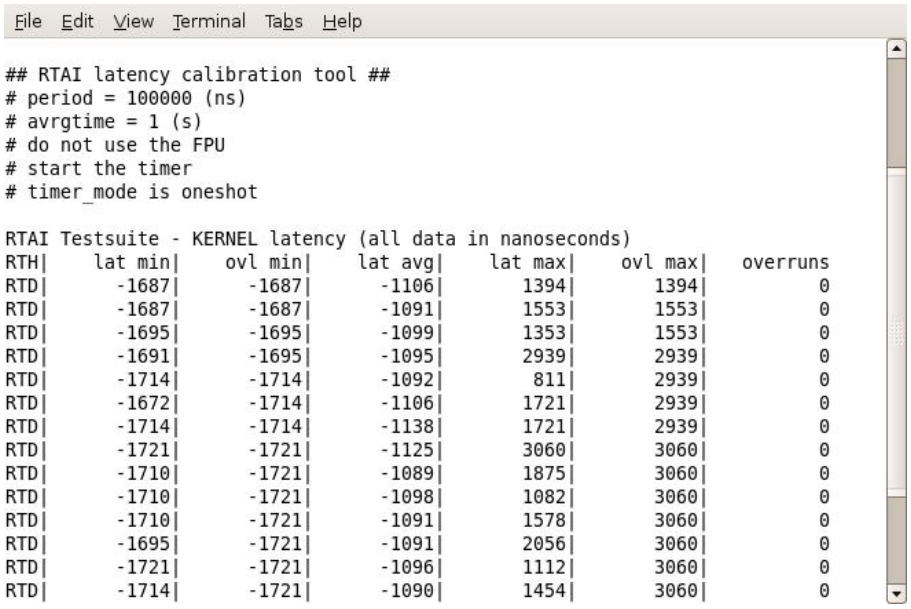
- Ir a la carpeta /usr/realtime/testsuite/kern/latency

```
cd /usr/realtime/testsuite/kern/latency
```

-Ejecutar el programa medidor de latencia

```
./run
```

Luego de escribir la línea anterior debemos obtener algo similar a como se muestra en la siguiente figura.



-Levantar los modulos para rtai y la tarjeta

```
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
insmod /usr/realtime/modules/rtai_lxrt.ko
```

-Ir a la carpeta /usr/src/rtai/testsuite/kern/latency/

```
cd /usr/src/rtai/testsuite/kern/latency/
```

-Llamar al modulo de latencia

```
insmod latency_rt.ko period=1000000
```

-Mostrar las medidas de latencia

```
./display
```

-Remover el modulo de latencia

```
rmmod latency_rt
```

.7 Instalación de Comedi

Comedi es un proyecto de software libre; es una colección de drivers. Estos drivers están implementados como una combinación de los siguientes elementos:

- Un módulo del kernel de Linux que proporciona las funciones más comunes.
- Módulos individuales para las funciones de bajo nivel para cada dispositivo.

Estos módulos son la combinación de tres elementos de software complementarios: un elemento genérico, independiente del dispositivo de la API; una colección de módulos del kernel Linux que implementan esta API para una amplia gama de tarjetas, y un usuario de Linux para la colección de espacio con una interfaz de programación orientada a desarrolladores para configurar y usar las tarjetas.

Proporciona las herramientas y bibliotecas para un montón de tarjetas de adquisición de datos (DAQ). En este caso particular la instalación es realizada utilizando la Tarjeta de Adquisición NI PCI-6024E.

Descripción del Proceso de Instalación

-Ir a la carpeta /usr/local/src/comedi

```
cd /usr/local/src/comedi
```

-Ejecutar el script de levantamiento de archivos de configuracion

```
sh autogen.sh
```

-Configurar los paquetes de instalación, indicando el directorio de Linux, el directorio de Rtai y desabilitando la opcion de tarjeta de adquisición de datos tipo pcmcia.

```
./configure --with-linuxdir=/usr/src/linux --with-
rtaidir=/usr/realtime --enable-kbuild --disable-pcmcia
```

-Generar los paquetes de instalación

```
make
```

-Instalar el los driver comedi

```
make install
```

-Comprobamos los módulos levantados

```
depmod -a
```

-Crear los módulos de la tarjeta de adquisición de datos dentro de los dispositivos

```
make dev
```

-Copiar las librerías comedi y comedilib al grupo de librerías por defecto del sistema

```
cp include/linux/comedi.h include/linux/comedilib.h /usr/include/
```

-Copiar las librerías comedi y comedilib al grupo de librerías por defecto del sistema

```
cp include/linux/comedi.h include/linux/comedilib.h
/usr/local/include/
```

-Crear un vínculo con la librería comedi

```
ln -s /usr/include/comedi.h /usr/include/linux/comedi.h
```

-Crear un vínculo con la librería comedilib

```
ln -s /usr/include/comedilib.h /usr/include/linux/comedilib.h
```

-Inicializar los módulos de Rtai

```
insmod /usr/realtime/modules/rtai_hal.ko

insmod /usr/realtime/modules/rtai_up.ko

insmod /usr/realtime/modules/rtai_fifos.ko

insmod /usr/realtime/modules/rtai_sem.ko

insmod /usr/realtime/modules/rtai_mbx.ko

insmod /usr/realtime/modules/rtai_msg.ko

insmod/usr/realtime/modules/rtai_netrpc.ko
ThisNode="127.0.0.1"

insmod /usr/realtime/modules/rtai_shm.ko

insmod /usr/realtime/modules/rtai_signal.ko

insmod /usr/realtime/modules/rtai_tasklets.ko

modprobe comedi

modprobe kcomedilib

modprobe comedi_fc
```

```
modprobe 8255

modprobe ni_pcimio

comedi_config -v /dev/comedi0 ni_pcimio
```

-Copiar el script de inicialización de módulos de la tarjeta de adquisición de datos en la carpeta de scripts de arranque

```
cp lmbrtai /etc/init.d/
```

-Configurar como script de arranque al script copiado en el paso anterior

```
update-rc.d lmbrtai defaults
```

.8 Instalación de Rtai parte 2

-Copiar el script de inicialización de módulos de rtai en la carpeta de scripts de arranque

```
cp lmbrtai2 /etc/init.d/
```

-Configurar como script de arranque al script copiado en el paso anterior

```
update-rc.d lmbrtai2 defaults
```

-Ir a la carpeta /usr/src/rtai

```
cd /usr/src/rtai
```

-Verificar los valores del menú de Rtai

-Ejecutar el menú de la consola de configuración de Rtai

```
make menuconfig
```

-Generar los archivos configurados en el paso anterior

```
make
```

-Realizar la instalación de la configuración actual de Rtai

```
make install
```

-Reiniciar el computador y seleccionar el kernel de RTAI

.9 Instalación de Scilab

-Descargar el archivo comprimido de Scilab-4.1.2 de la siguiente dirección:

```
wget         http://www.scilab.org/download/4.1.2/scilab-4.1.2-
src.tar.gz
```

-Copiar el archivo comprimido de Scilab en la carpeta /usr/local/

```
cp scilab-4.1.2-src.tar.gz /usr/local/
```

-Ir a la carpeta /usr/local/

```
cd /usr/local/
```

-Descomprimir el archivo scilab-4.1.2-src.tar.gz

```
tar xzvf scilab-4.1.2-src.tar.gz
```

-Ir a la carpeta /usr/local/scilab-4.1.2/

```
cd /usr/local/scilab-4.1.2/
```

-Configurar los archivos de instalación excluyendo el uso java, y especificando la librería tcl y la librería tcl8.8

```
./configure --without-java --with-tcl-library=/usr/lib --with-tcl-include=/usr/include/tcl8.4
```

-Construir los archives configurados en el paso anterior

```
make all
```

-NO REALIZAR MAKE INSTALL

-Crear un vínculo de scilab en la carpeta de programas ejecutables

```
ln -s /usr/local/scilab-4.1.2/bin/scilab /usr/local/bin/scilab
```

.10 Instalación Addons Rtai-Scilab

-Ir a la carpeta /usr/src/rtai/rtai-lab/scilab/macros

```
cd /usr/src/rtai/rtai-lab/scilab/macros
```

-Instalar los paquetes Addons Rtai-Scilab

```
make install
```

-Instalar los privilegios de uso de los Addons Rtai-Scilab para cada usuario

```
make user
```

Nota: si desea que otro usuario tenga estos privilegios de uso, deberá ingresar a la sesión de usuario deseado, ir a la carpeta /usr/src/rtai/rtai-lab/scilab/macros/ y hacer el paso make user.

.11 Instalación de QRtaiLab

QRtailab es una interfaz de usuario para RTAI. QRtaiLab es un puerto QT basado xrtailab. Se puede utilizar como un osciloscopio virtual y permite el control de aplicaciones para interactuar con el ejcutable en tiempo real.

Antes de realizar su instalación es necesario que verifique si QT para C++ está instalada en una versión >=4.3.0, al igual que la biblioteca qwt en su versión >=5.0.2.

A continuación se puntualizará el proceso de instalación de esta aplicación.

-Ir a la carpeta /opt/

```
cd /opt/
```

-Descargar el archivo comprimido de Scilab-4.1.2 de la siguiente dirección:

```
http://downloads.sourceforge.net/qrtailab/QRtaiLab-
0.1.9.tar.gz
```

-Descomprimir el archivo QRtaiLab-0.1.9.tar.gz

```
tar xvzf QRtaiLab-0.1.9.tar.gz
```

-Ir a la carpeta /opt/qrtailab

```
cd /opt/qrtailab
```

-En el archivo de configuración qrtailab.conf realizamos el siguiente cambio

Reemplazar '/usr/include/qwt/' por '/usr/include/qwt-qt4/' y también reemplazar '-lqwt' con '-lqwt-qt4'

-Prepara los archivos de configuración

```
qmake-qt4
```

-Preparar los archivos de instalación

```
Make
```

-Realizar la instalación de los paquetes previamente configurados

```
make install
```

4. EJEMPLO UTILIZANDO SCICOS

4.1. Ejemplo Básico usando Rtai.

INTRODUCCION

A través de este tutorial crearemos un sencillo ejemplo utilizando Scicos como herramienta para la simulación de diagramas de bloques y Xrtailab que es una software de control y visualización de señales producidas por RTAI.

1. Abrimos una Terminal de comandos, nos dirigimos a Aplicaciones-> Accesorios Terminal (Figura 1)

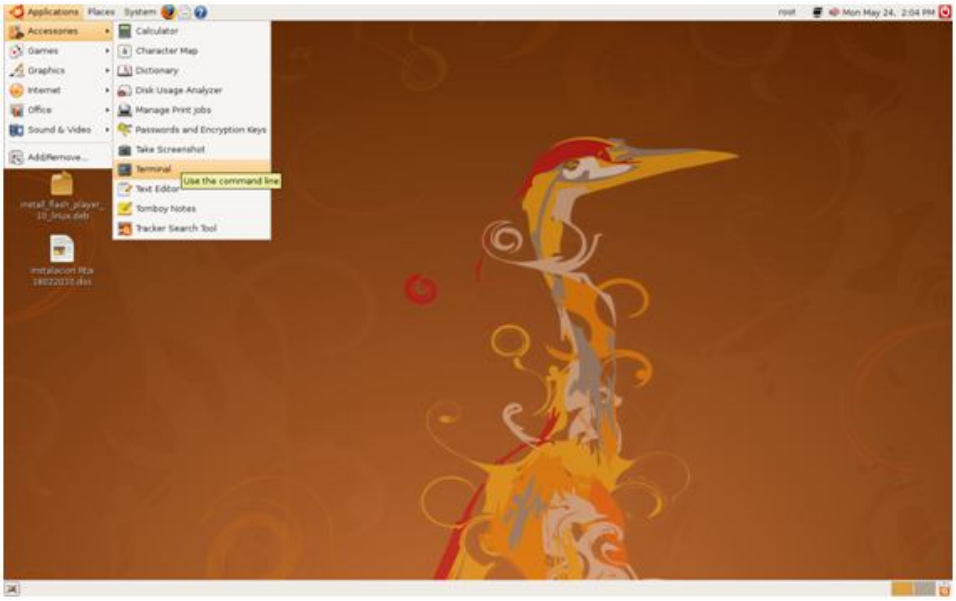


Figura 1. Terminal de Comandos

2. En la terminal escribimos scilab para iniciar el programa.

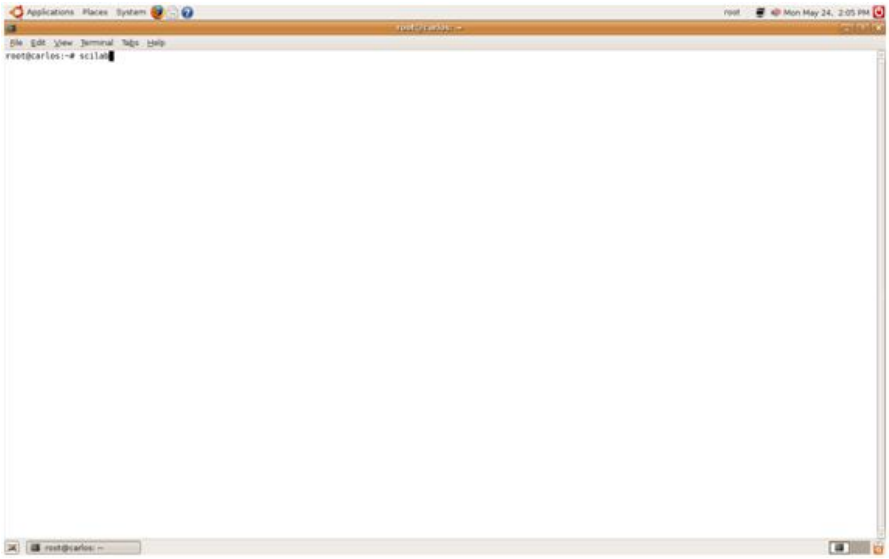


Figura 2.

3. Una vez iniciado Scilab, procedemos a abrir Scicos escribiendo la palabra *scicos* en la terminal de scilab. Es importante verificar que la librería RTAI está cargada correctamente observando el mensaje: ***Scicos-RTAI ready***, que se muestra en la consola de scilab al momento de iniciarla. Figura 3

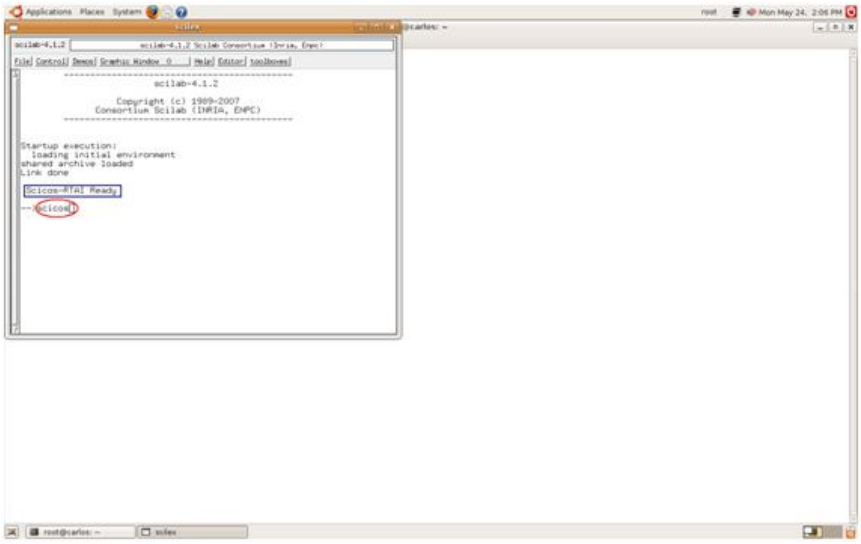


Figura 3. Scilab-Scicos

A continuación se muestra en la parte derecha de la figura el editor gráfico Scicos. Figura 4.

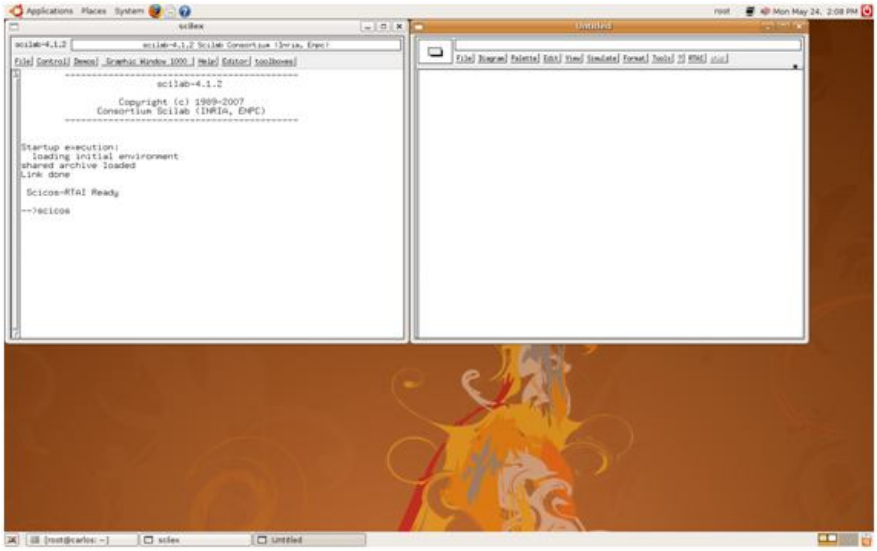


Figura 4.

4. Como siguiente paso con el cursor nos dirigimos al Menú PaletteàPalettes que se encuentra en la barra de herramientas (Figura 5.) para luego elegir la librería RTAI-Lab donde se encuentran algunos bloques a utilizar en nuestro diagrama de ejemplo. Ver Figura 6.

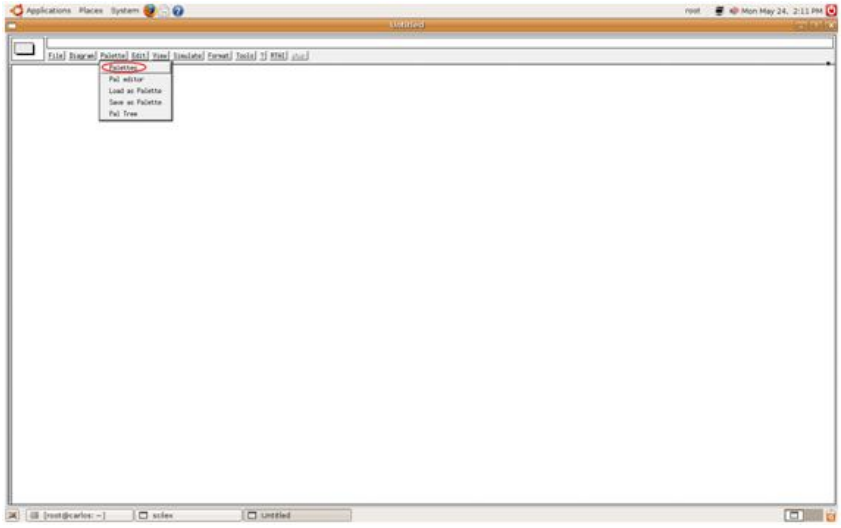


Figura 5.

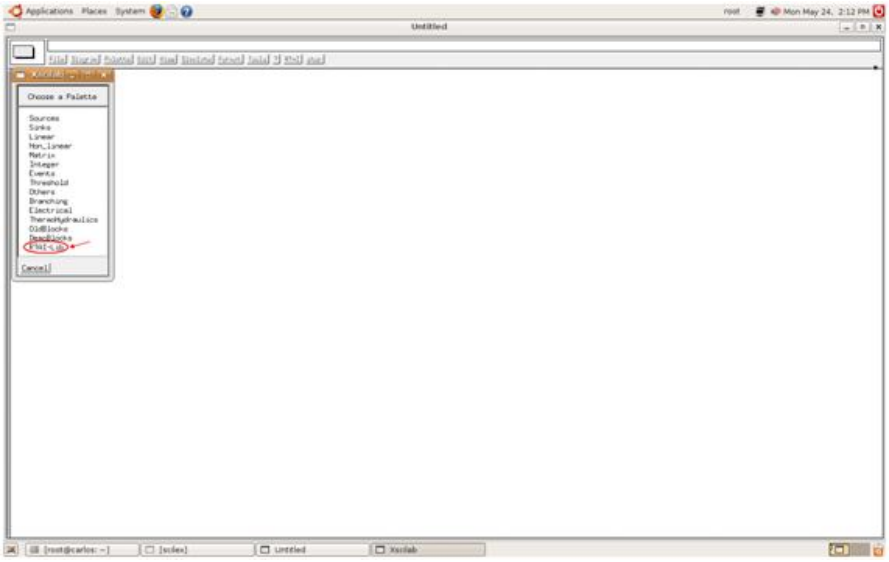


Figura 6.

· Elementos de RTAI-Lib

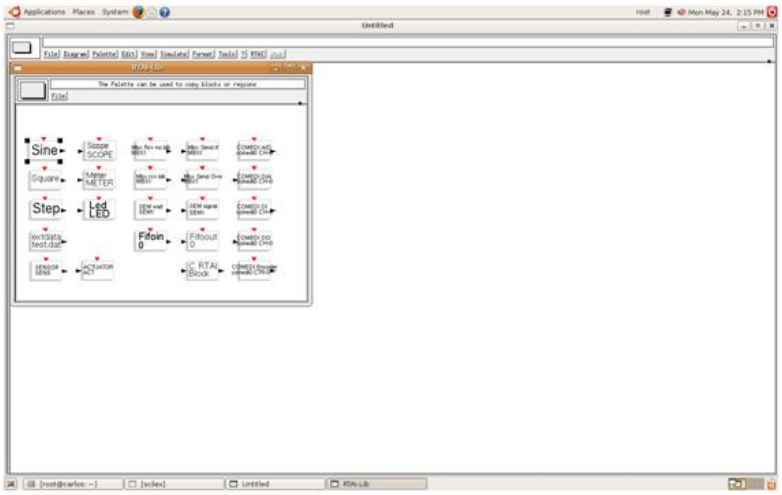


Figura 7

5. Con el mouse escogemos el bloque 1 y 2 mostrado en la figura 8 y los ubicamos en el diagrama que aún está en blanco, seguidamente volvemos al menú *Palettes* para elegir otro bloque que se encuentra en otra librería que es ***Sources***, y escogemos el bloque mostrado en la figura 9

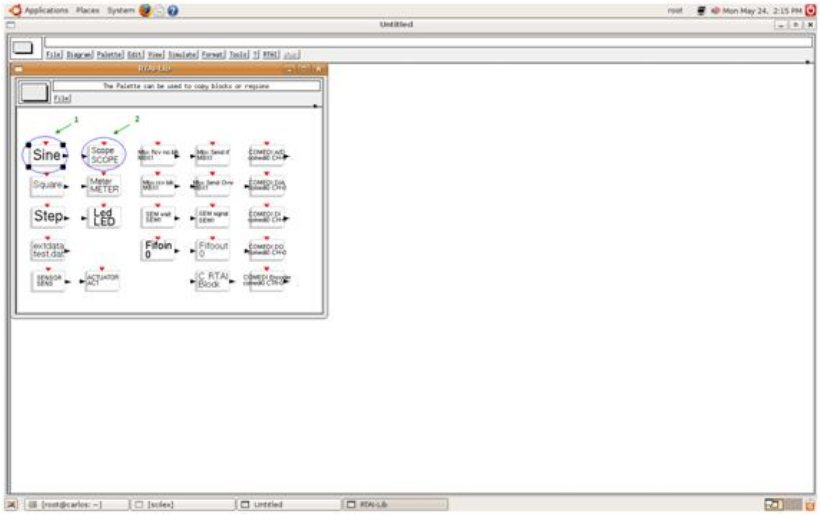


Figura 8

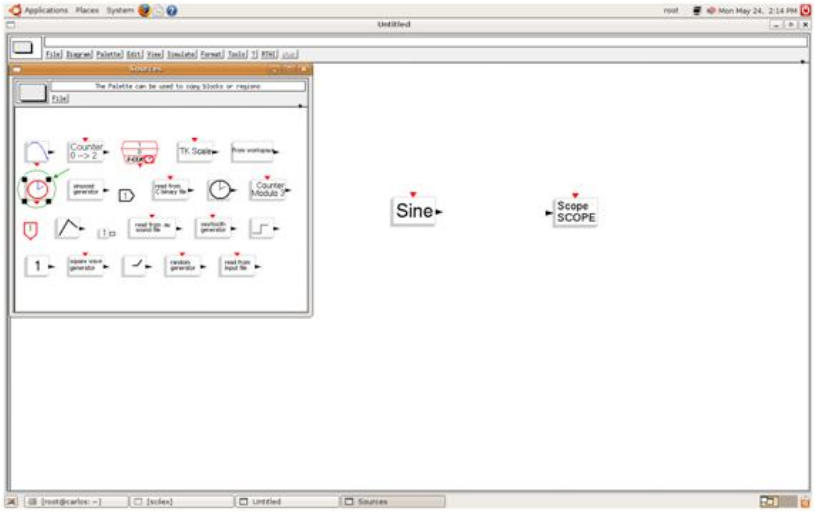


Figura 9

6. Una vez elegido todos elementos de nuestro primer diagrama, procedemos a enlazar a los bloques manteniendo presionado el clic derecho desde un bloque hacia el otro. Figura 10

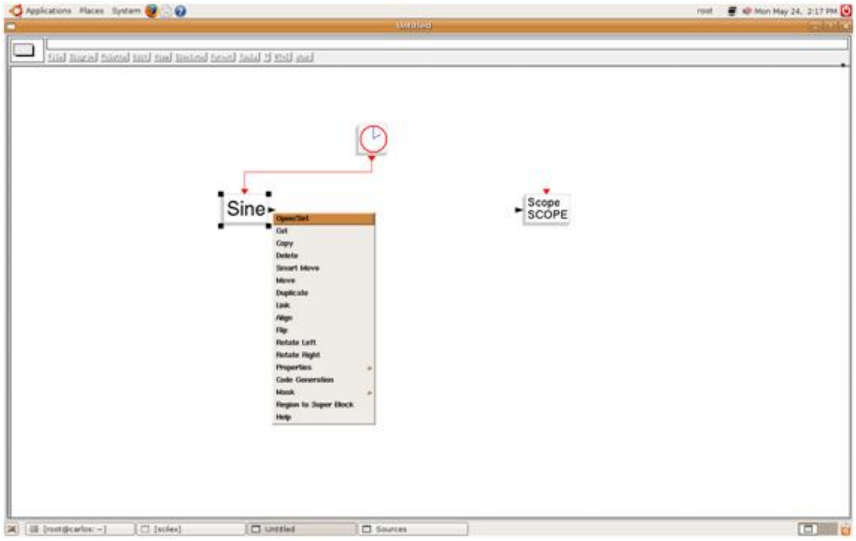


Figura 10.

A continuación se muestra el diagrama terminado. Figura 11

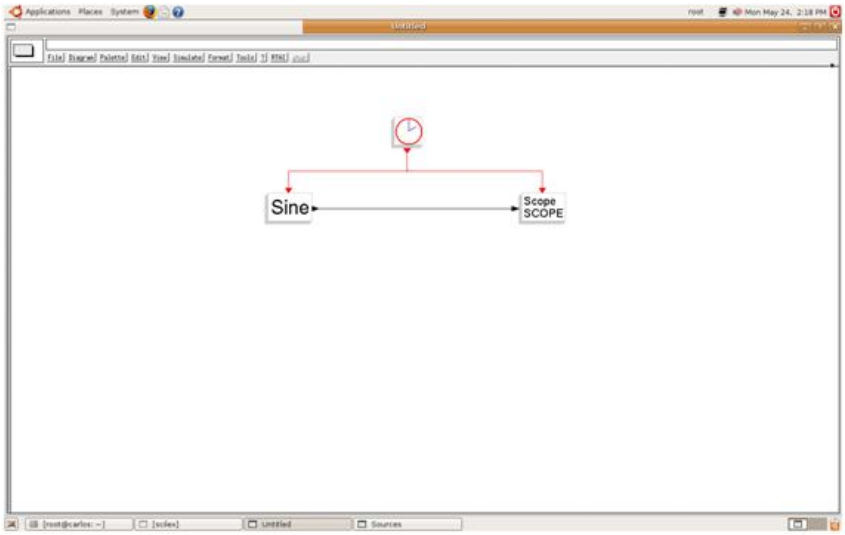


Figura 11.

7. Una vez creado el diagrama de bloques es necesario hacer Súper-bloque de la manera como se indica en la Figura 12.

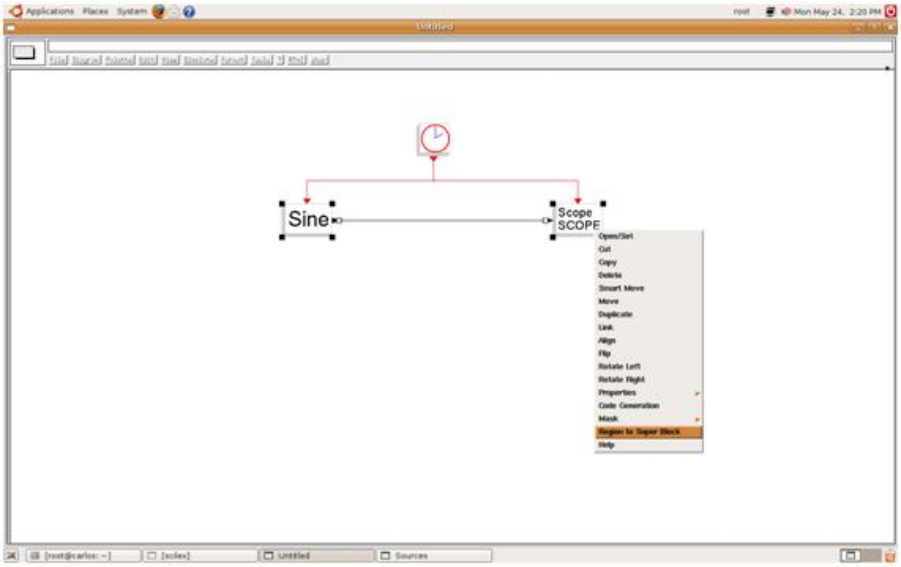


Figura 12.

·Diagrama con Súper-bloque. Figura 13

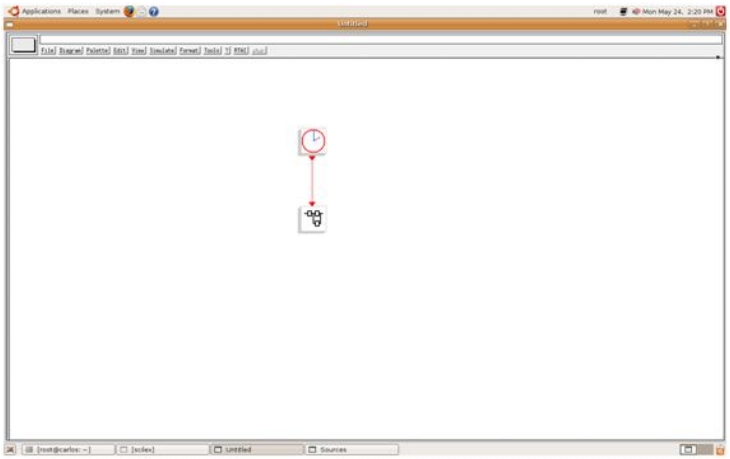


Figura 13

8. Una vez creado el Súper-bloque , lo seleccionamos, y seguidamente elegimos en la barra de herramientas principal al menú RTAI, aquí encontramos dos opciones: **"*Set Target* "** y ***"RTAI CodeGen"***, seleccionamos cada una de ellas respectivamente.

· Opcion RTAIàSet Target. Dejamos los valores por defecto para este ejemplo y damos clic en ***'OK'. Figura 14.***

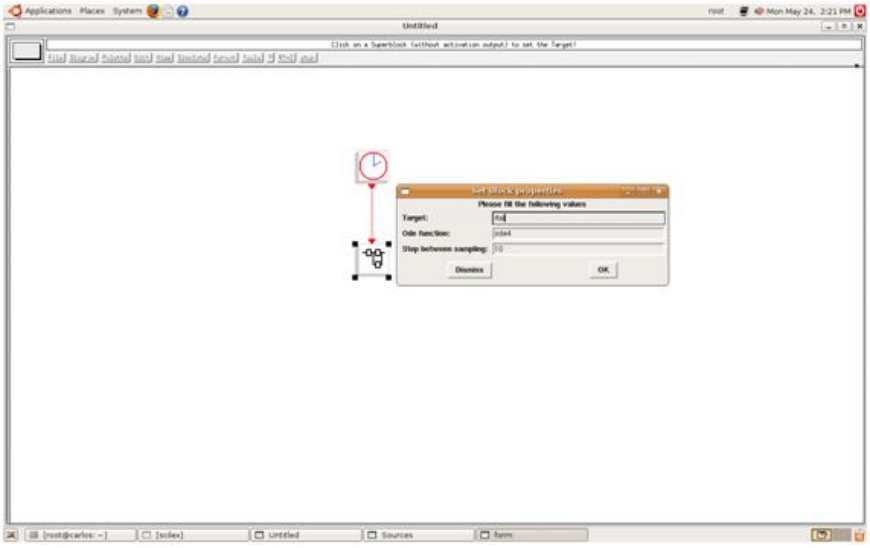


Figura 14.

·Opcion RTAIà RTAI CodeGen , aquí se escribe las propiedades del súper-bloque para porteriormente generar el ejecutable en tiempo real como se muestra en la Figura 15.

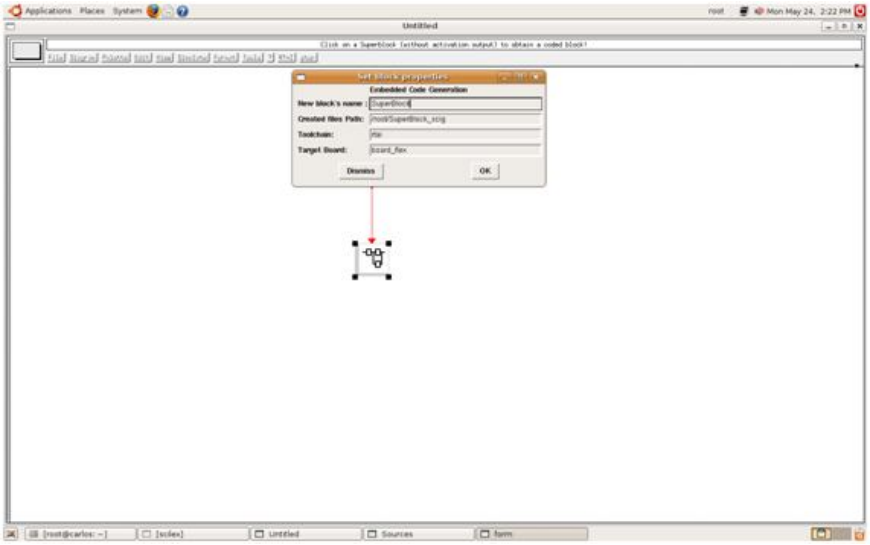


Figura 15

9. Guardamos el archivo con cualquier nombre, en este caso hemos asignado el nombre de ***Untitled*** (Figura 16)

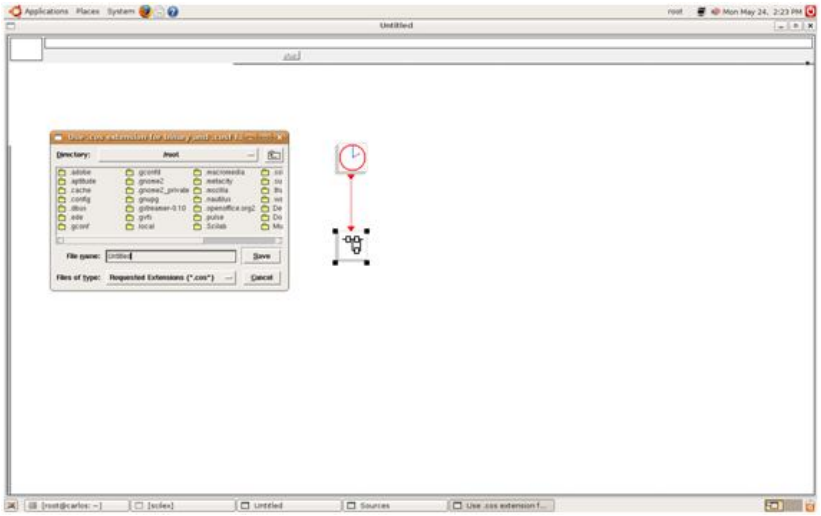


Figura 16.

10. A continuación debemos nos ubicamos en el directorio donde se encuentra el ejecutable anteriormente generado, y los ejecutamos de la siguiente manera. En la terminal debe salir un mensaje de '***Target Settings'*** como en la Figura 17.

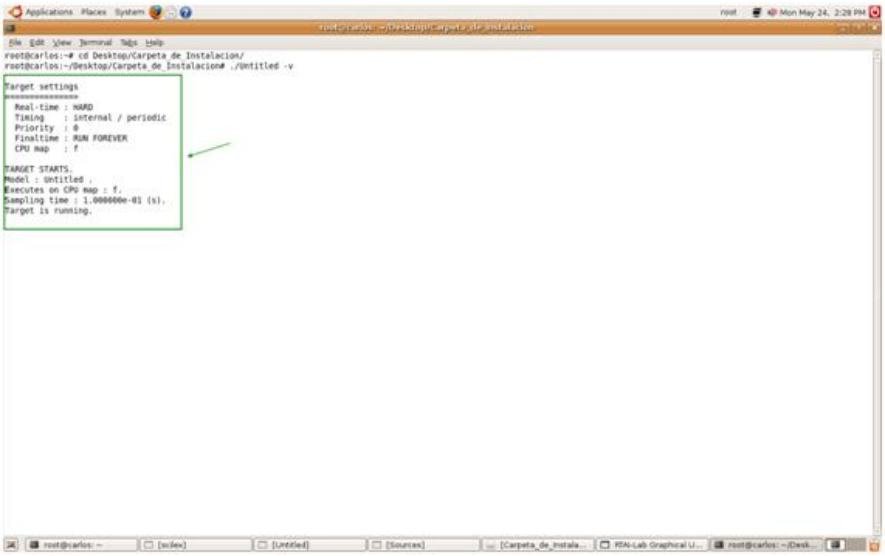


Figura 17.

11. Abrimos otra terminal y en ella escribimos xrtailab (Figura 18) para abrir el software que permitirá ver la señal generada, damos enter y el resultado será el de la Figura 19.

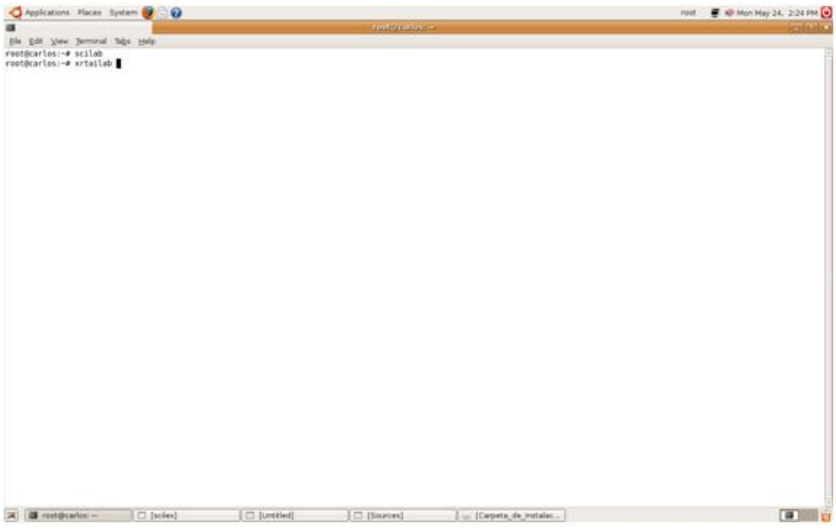


Figura 18

12. Una vez iniciada la aplicación damos click en el Menu FileàConnect (Figura 19) para iniciar la simulacion en tiempo real.

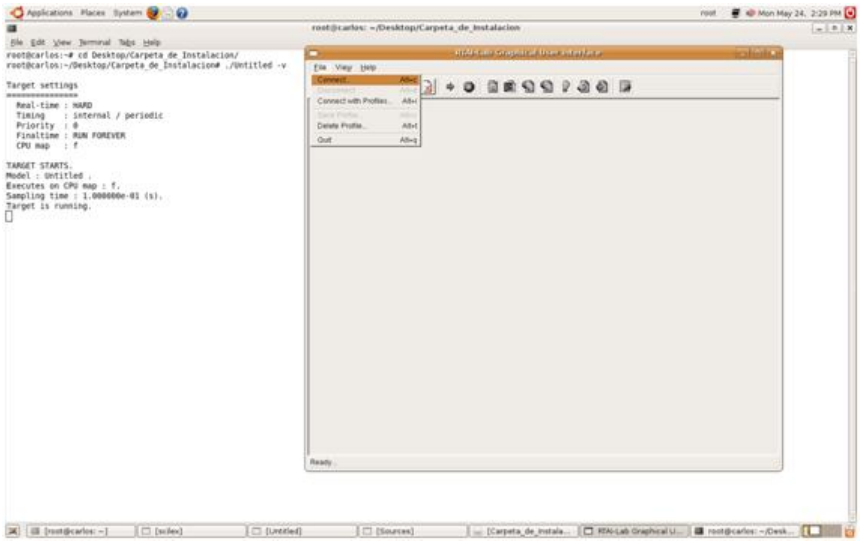


Figura 19.

· Verificamos los datos en la ventana ***"Connect to Target";*** en este ejemplo dejamos todos los valores por defecto, pero es indispensable volver a escribir la dirección IP a pesar que ya venga escrita por defecto (Figura 20)

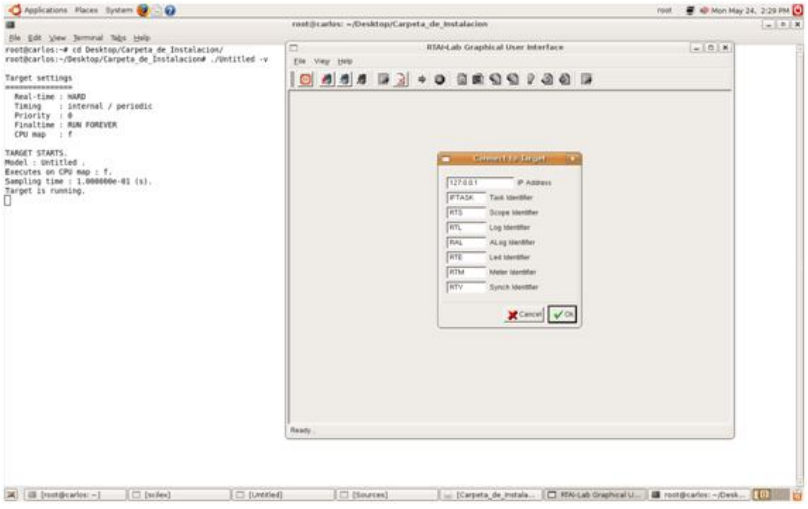


Figura 20

13. Damos clic en “***OK” .***El resultado es el siguiente. Figura 21.

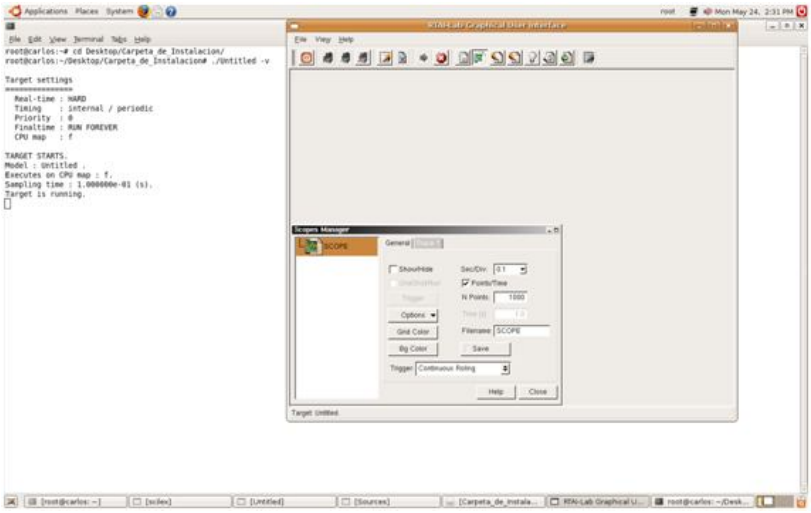


Figura 21.

· Si no puede ver aún la señal, es porque la opción del checkbox ‘***ShowHide*** ‘ no está aún habilitada. Figura 22.

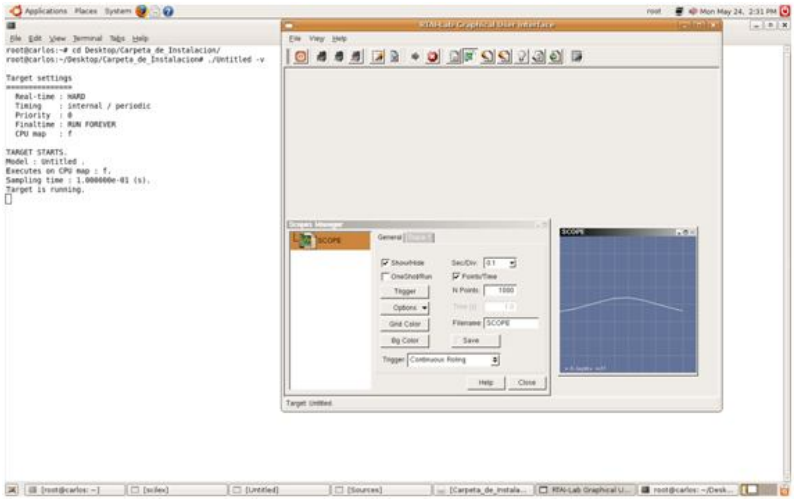


Figura 22.

14. Finalmente para detener la simulación damos clic en el círculo rojo con X en la parte superior y damos por terminado el ejemplo. Es importante verificar en la terminal que se ha detenido el proceso, lo mismo ocurre en la Terminal Scilab. Ver Figura 23 y 24.

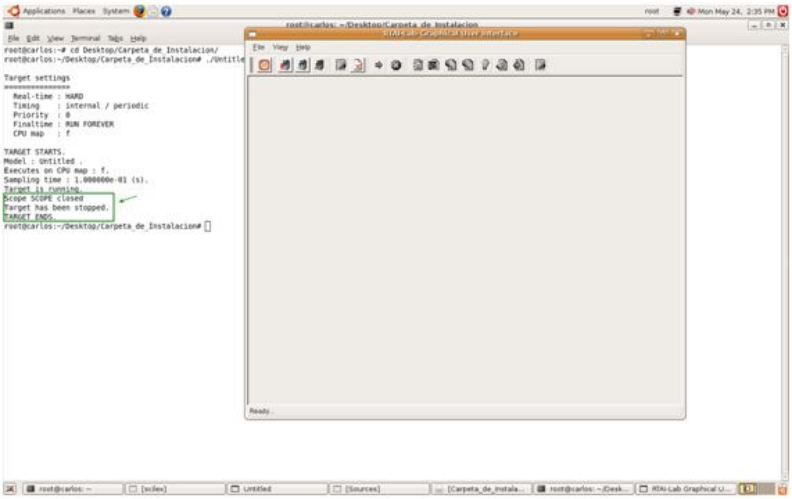


Figura 23.

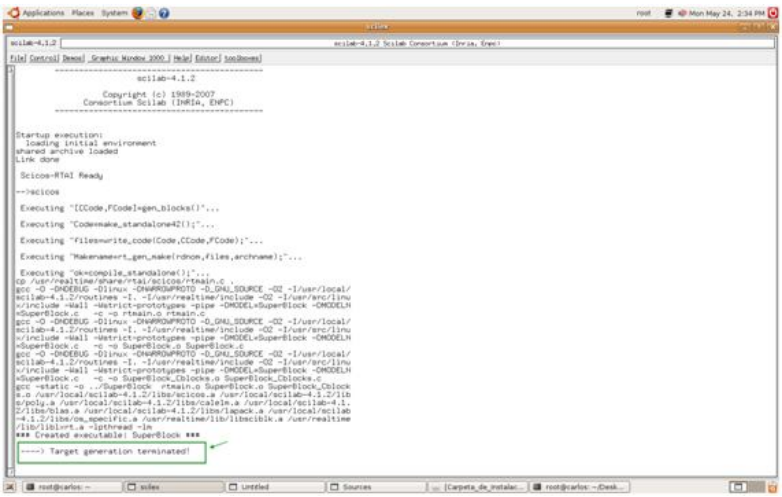


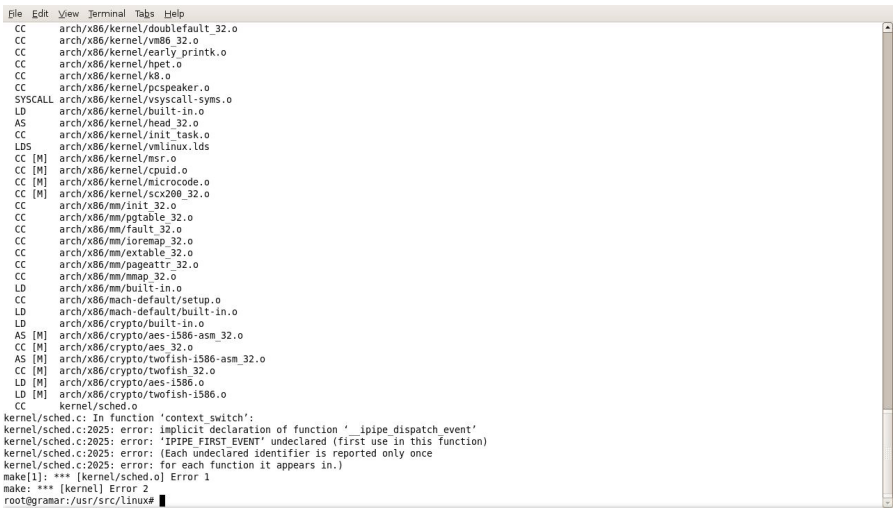
Figura 24

TROUBLESHOOTING

Kernel

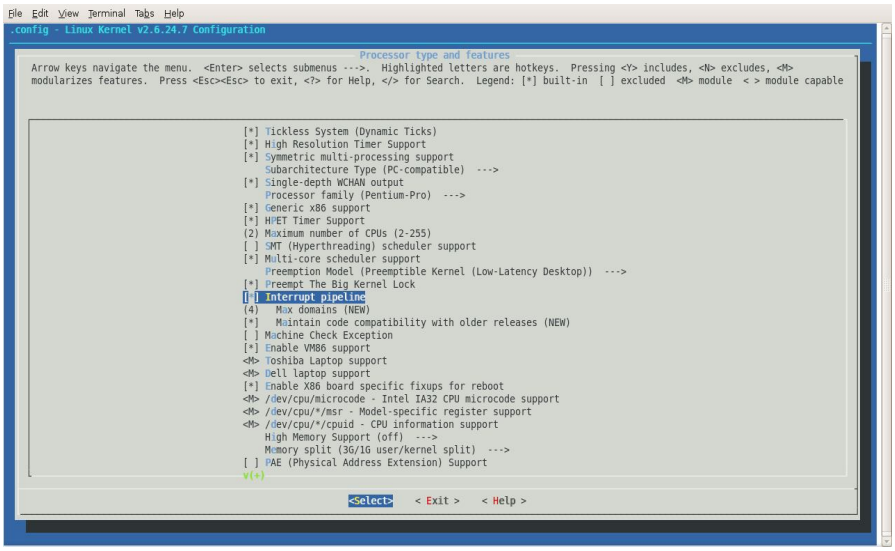
Error de compilación

Al ejecutar el comando make luego de haber realizado la configuración del nuevo kernel puede presentrarse el siguiente error:



Solución:

Volver a ingresar al menu de configuracion de parámetros y señalar o verificar que la siguiente opcion "*Interrupt Pipeline"* está habilitada.



SCILAB-SCICOS

Dispositivo no encontrado

Error:

Al ejecutar uno de los programas creados por la compilación en scicos de scilab, el ejecutable se auto-finaliza o identifica un error de dispositivo no encontrado.

Solución:

Buscar si se encuentran cargados los dispositivos y sub-dispositivos de la tarjeta. Para esto se debe listar los elementos del directorio /dev/ , con el comando: **ls /dev/**

En caso de no encontrarse cargados los dispositivos, es decir, que no aparezcan listados con los nombres de comedi0, comedi1,… etc., se los debe de cargar con el script de carga de dispositivos que debe estar ubicado en las carpetas de scripts de ejecución al arranque, según el proceso de instalación que hemos detallado.

Para ejecutar ese script, en nuestro caso lo hemos llamado ***lmbrtai***, se debe ejecutar la siguiente línea de comando: **/etc/init.d/lmbrtai start.**

En caso de no encontrarse cargados los sub-dispositivos, es decir, que no aparezcan listados los nombres de comedi0_sub0, comedi0_sub1, comedi0_sub2, en el caso de la tarjeta PCI 6024E, se los debe de cargar con el script de carga de sub-dispositivos que debe estar ubicado en las carpeta de scripts de ejecución al arranque, según el proceso de instalación que hemos detallado.

Para ejecutar ese script, que nosotros hemos nombrado como **lmbrtai2**, se debe ejecutar la siguiente línea de comando: **/etc/init.d/lmbrtai2 start.**

COMEDI

Dispositivo no configurado

Error

Al ejecutar la línea de de código con el comando comedi-config, dentro del proceso de instalación de comedi, se produce un error en el que se detalla que el dispositivo no puedo ser instalado, y no se presentan una cadena semejante a **“0,0,0,0,0,0,0,0,0,0,0,0,0,0,…,0,0,0,0”.**

En el caso del sistema Operativo Ubuntu, se produce un mensaje, que puede ser confundido con un error crítico, pese a que este no signifique que el dispositivo no pudo ser configurado, este mensaje detalla que el dispositivo no puede ser configurado por que ya ha sido configurado previamente, pero si se muestra la cadena “0,0,0,0,0,0,0,0,0,0,0,0,0,0,…,0,0,0,0”, lo cual indica que el dispositivo se encuentra en funcionamiento.

Esta excepción se ha producido por el proceso de instalación. El caso identificado durante nuestras pruebas, estuvo vinculado a la versión de comedi y comedilib, que se produce cuando se descarga manualmente ambos paquetes, y a su vez que los paquetes no contengan los drivers apropiados para la tarjeta de adquisición de datos que se tenga instalada.

Solución

De acuerdo a los resultados generados por nuestras pruebas, utilizando un sistema de control de versiones como ell cvs o svn, no se producen estos problemas. Por lo que recomendamos, volver a descargar e instalar, comedilib y comedi, por un medio automático como cvs o svn. Caso contrario, se debe de buscar el paquete que contenga los drivers adecuados para la tarjeta que se esta usando, esta información se encuentra en la página de comedi.

Comando de Configuración no instalado

Error

Luego de proceder con los scripts de instalación hasta el script numero 4 sin que se presenten problemas visibles, no se puedo ejecutar por completo el script 4 ya que no se encontraba instalado el comando comedi-config. Se restrae e error hasta la parte de la instalación de comedi-lib, donde, luego de hacer las pruebas de instalación de dicho software, se pudo identificar el error:

```
configure: creating ./config.status
 cd  && /bin/sh ./config.status Makefile
/bin/sh: ./config.status: No existe el fichero o el directorio
make: *** [Makefile] Error 127
```

Este error se produce al ejecutar make dentro del proceso de instalación del comedi-lib

Solución

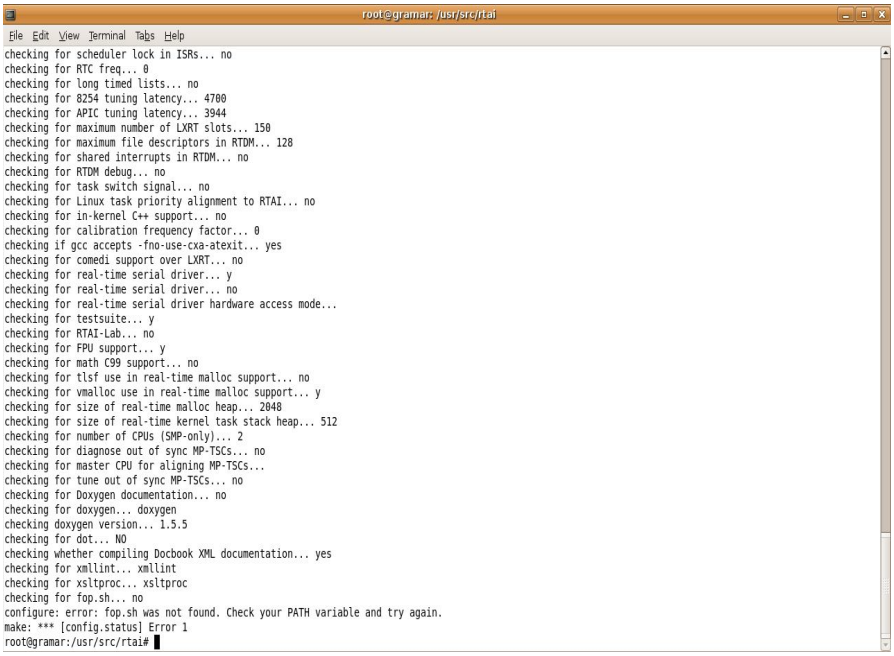
Para solucionar este error en caso de que se presente en sistemas operativos como fedora o centos, es necesario instalar las herramientas libtool-devel, ruby, docbook-pdf, en ambos casos, fedora o centos , usar el comando yum install.

Luego, en el proceso de instalación, en el scriptRtai2, en la sección de instalación de Comedilib, no ejecutar el comando sh autogen.sh, ya que este comando se lo utiliza para crear el archivo de configuración .configure, archivo que ya viene por defecto dentro del paquete comprimido de comedilib que se usa para la instalación, por lo que no es necesario re-escribirlo ya que esto puede producir el error en cuestión.

RTAI

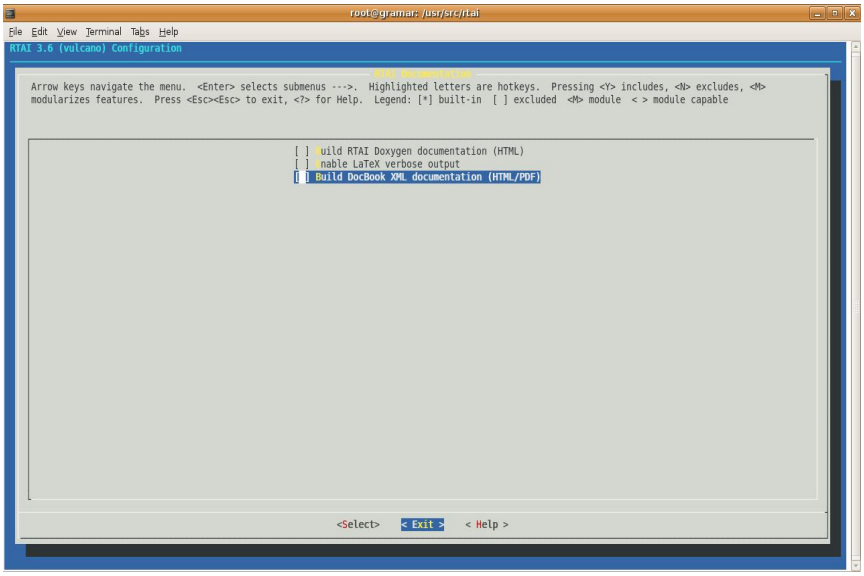
No se encontró el archivo fop.sh. en la compilación de RTAI parte 1

Luego de haber añadido la opción de documentación Docbook XML en el menu de configuración en la sección de 'RTAI Documentation' se presentó error en la compilación como se muestra en la siguiente figura.



Solución:

Volver a ingresar al menu de configuración de RTAI y deshabilitar la siguiente opción:



Falta de carga de módulos de rtai

Error:

No se pueden ejecutar los script que contiene bloques de salida o de entrada de dispositivos en comedi al momento de arranque del sistema operativo.

No se cargan los sub-dispositivos comedi_sub entre los nodos dentro de la carpeta /dev. Esto se debe a a que el computador no identifica el comando comedi-config, esto se debe a que nuestros script corren antes del script /root/.bashrc, el cual solo corre al autenticarse el usuario root, por lo tanto el computador al momento de ejecutar los scripts lmbrtai y lmbrtai2 no contiene en su path la ruta donde se encuentra almacenado le ejecutable del comando comedi-config, es decir aun no se ejecuta las lineas de codigo del script bashrc:

```
PATH=$PATH:/usr/realtime/bin

export PATH" >> /root/.bashrc
```

Producidas por las siguientes lineas en el scriptRtai2 durante la instalacion

```
echo "PATH=$PATH:/usr/realtime/bin" >> /root/.bashrc

echo "export PATH" >> /root/.bashrc
```

Solución:

Para poder corregir este error, durante el proceso de instalación o posteriormente, una vez identificado que no se cargan los sub dispositivos, se debe de ejecutar la siguiente linea de codigo:

```
echo "comedi_config -v /dev/comedi0 ni_pcimio" >> /root/.bashrc

echo "clear" >> /root/.bashrc
```

La ultima línea de código se la coloca para limpiar la pantalla y que no aparezca un mensaje sobre la carga de dispositivos cada vez que se abra un terminal. Si desea no visualizar el error producido por esta discrepancia en el orden de carga, durante la carga del sistema operativo, edite los scripts lmbrtai y lmbrtai2, ubicados en la carpeta /etc/init.d/ y remueva de ambos la siguiente línea de código:

```
comedi_config -v /dev/comedi0 ni_pcimio
```

Por lo tanto para corregir este error, las ultimas tres lineas de código del archivo /root/.bashrc deben ser las siguientes, en este preciso orden:

```
PATH=$PATH:/usr/realtime/bin

export PATH" >> /root/.bashrc

comedi_config -v /dev/comedi0 ni_pcimio

clear
```

Referencias

Cada proyecto tiene referencias diferentes, entre las básicas tenemos:

[1] http://www.mesa3d.org/

[2] http://www.fltk.org/

[3] http://www.comedi.org/

[4] http://www.scilab.org

[5] http://qrtailab.sourceforge.net/

Instalación Modnum para Scilab 4.2.1

ModNum (Modulaciones Numéricas) es una herramienta de código abierto que contiene librerías para la modelación y simulación de sistemas de comunicación.

El proceso de instalación descrito a continuación pertenece al grupo de herramientas diseñadas para el trabajo en base a la herramienta Scilab versión 4.1.2, versión utilizada durante las pruebas de la herramienta RTAI.

-Descargar el código fuente de la herramienta de la siguiente dirección:

http://www-scicos.inria.fr/ScicosModNum/modnum_web/files/modnum_412/modnum_412_src.tar.gz

-Copiar el paquete descargado a la carpeta /root

Los siguientes pasos deben ser realizados en la terminal como superusuario

-Ir a la carpeta /root

```
cd /root
```

-Descomprimir el paquete descargado

```
tar xvz modnum_412_src.tar.gz
```

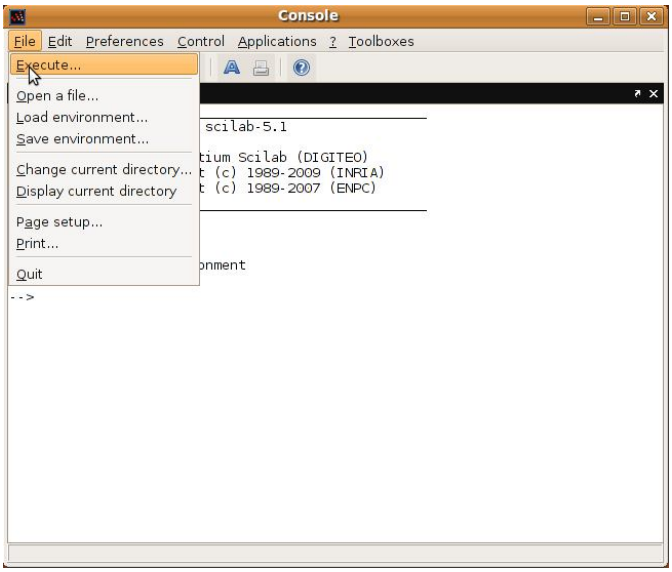
-Cambiar el nombre de la nueva carpeta

```
mv modnum_421_src nodmun
```

-Ejecutar scilab

```
Scilab
```

-Ir al Menu File -> Execute



-Ejecutar el script de construcción ubicado en:

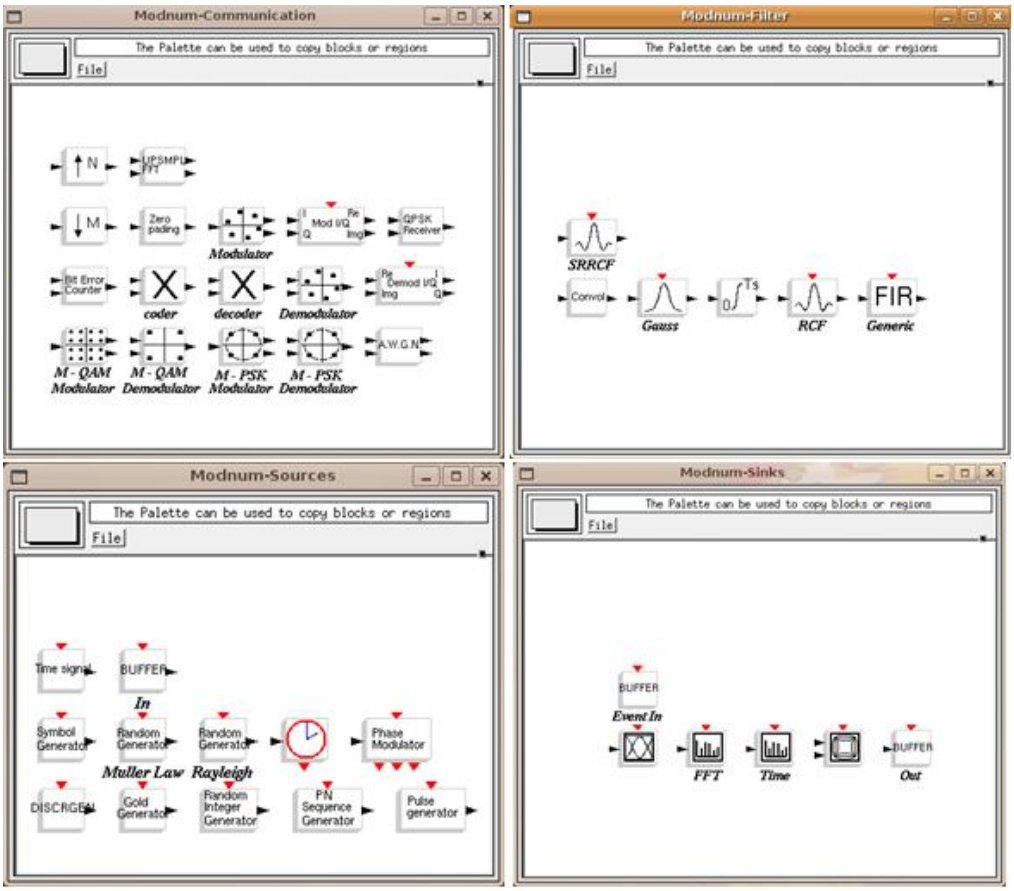
```
/root/modnum /builder.sce
```

-Ejecutar el script de carga ubicado en:

```
/root/modnum/loader.sce
```

Este último script deber ser ejecutado por el ejecutador de scilab cada que se inicie el programa y que se desee utilizar las paletas de Modnum.

La siguiente imagen muestra algunas de las paletas incluidas por la herramienta ModNum.



Apendice B

Creación de LiveCD RTAI

Creación de un LiveCD RTAI

Este documento muestra los pasos para la elaboración de un CD-Live de instalación autónoma supervisada para un sistema de Tiempo Real RTAI-COMEDI-SCILAB, en base al sistema operativo Ubuntu 8.4.3.

Los pasos indicados en este documento están orientados a generar un archivo con extensión, es decir imagen de sistema que contenga los primeros pasos de instalación de las herramientas, es decir los pasos de instalación de las herramientas Mesa3D, EFLTK y los primeros pasos en el proceso de parchado y compilación del kernel.

Los pasos restantes de la instalación incluyendo la compilación e instalación del kernel, será realizados automáticamente por script diseñados para auto ejecutarse luego de haber finalizado la instalación del sistemas operativo contenido en el archivo imagen iso. Estos scripts son automáticamente ejecutados cuando el usuario abre un terminal por primera vez, luego de que cada script sea ejecutado, le pedirá al usuario reiniciar la máquina e ingresar en el núcleo parchado de RTAI. Luego al volver a entrar a un terminal se ejecutara el siguiente script, hasta que el finalizar el proceso a ejecutarse se encarga de auto eliminar todos los scripts para no repetir el proceso de instalación.

Para poder llevar a cabo este proceso es necesario ingresar al sistema como super-usuario.

Proceso de Instalación de RTAI-COMEDI-SCILAB Previo a la Creación del Archivo ISO.

Para poder realizar la elaboración de la imagen del proceso de instalación es necesarios realizarlos primeros pasos que respectan al proceso de instalación de RTAI y las herramientas necesarias de tal forma de que se suprima la necesidad de descargar ciertos paquetes.
La ejecución de los comandos que se muestran a continuación se debe de realizar en la termina situándose en la carpeta donde se encuentren los paquetes descargados de las herramientas. Para fines de minimizar el tiempo de instalación, se asume que se cuenta con todos los paquetes de las herramientas a instalar dentro de una sola carpeta, esta carpeta la hemos denominado RtaiComediScilabInstallFolder/.

- Instalación de Dependencias Previas

apt-get install subversion build-essential automake checkinstall x11proto-xext-dev xlibs-static-dev libxext-dev libxt-dev libncurses5-dev fakeroot kernel-package swig python-dev libtool libboost-program-options-dev libgsl0-dev libxmu-dev libxi-dev bison doxygen g77 gfortran flex sablotron xaw3dg-dev libpvm3 pvm-dev libgtkhtml2-dev libzvt-dev libvte-dev tcl8.4 tk8.4 tcl8.5-dev tk8.5-dev libqt4-dev libqwt5-qt4-dev qt3-dev-tools libdrm-dev xorg-dev cvs tcl8.4-dev tk8.4-dev libglu1-mesa-dev
echo "Se han instalado las dependencias iniciales"

- Instalacion Mesa3D

```
cp  MesaLib-7.0.3.tar.gz /usr/local/src
cp rtai-3.6-cv.tar.bz2 /usr/src/
cp linux-2.6.24.7.tar.bz2 /usr/src/
cd /usr/local/src
tar xvf MesaLib-7.0.3.tar.gz
cd Mesa-7.0.3
make realclean
make linux-x86-static
make install
echo "Mesa3 instalado correctamente"
```

- Instalacion de EFLTK

```
cd /usr/local/src
echo "Espere mientras se descarga efltk"
svn co https://ede.svn.sourceforge.net/svnroot/ede/trunk/efltk
echo "Espere mientras se descarga ede"
svn co https://ede.svn.sourceforge.net/svnroot/ede/trunk/ede
cd efltk
autoconf
./configure --disable-mysql --disable-unixODBC
./emake
./emake install
echo "/usr/local/lib" >> /etc/ld.so.conf
/sbin/ldconfig
echo "EFLTK instalado correctamente"
```

- Kernel Parte 1

```
cd /usr/src/
tar xjvf rtai-3.6-cv.tar.bz2
ln -s rtai-3.6-cv rtai
tar xjvf linux-2.6.24.7.tar.bz2
```

```
mv /usr/src/linux-2.6.24.7 /usr/src/linux-2.6.24.7-rtai
ln -s linux-2.6.24.7-rtai linux
cd /usr/src/linux
patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-linux-2.6.24-x86-2.0-07.patch
```

Para facilidad de la ejecución de los scripts restantes de instalación como la carpeta RtaiComediScilabInstallFolder/ dentro de la carpeta /opt. Es decir a partir de este paso todos los paquetes de instalación y los scripts se encuentran dentro de la carpeta de ruta absoluta: /opt/RtaiComediScilabInstallFolder/.

Es también necesario tener instalada la herramienta a utilizar para la creación de la imagen del disco. Para instalar esta herramienta ejecutar los siguientes comandos

```
gedit /etc/apt/sources.list
```

Y escribir en la última línea del documento la siguiente línea:
deb http://www.geekconnection.org/remastersys/repository ubuntu/

Guardar los cambios en el documento y realizar la actualización de las bases del instalador y la instalación de remastersys

```
apt-get update
apt-get install remastersys
```

Para poder programa la ejecución de los scripts de instalación en el primer arranque de la computadora, es necesario colocar el siguiente código en la última línea del documento /root/.bashrc

```
sh /opt/RtaiComediScilabInstallFolder/scriptRtai1
```

Luego se procederá a realizar la creación de la imagen del sistema. Para eso diríjase al menú System –> Remastersys Backup

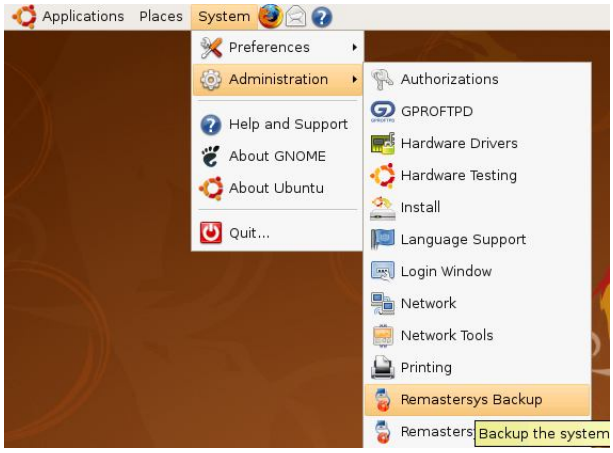


Figura 1. Ubicación de Remastersys Backup

Se mostrara una pantalla a la cual se le dará aceptar. Luego en la ventana emergente seleccionar la opción Backup Complete System

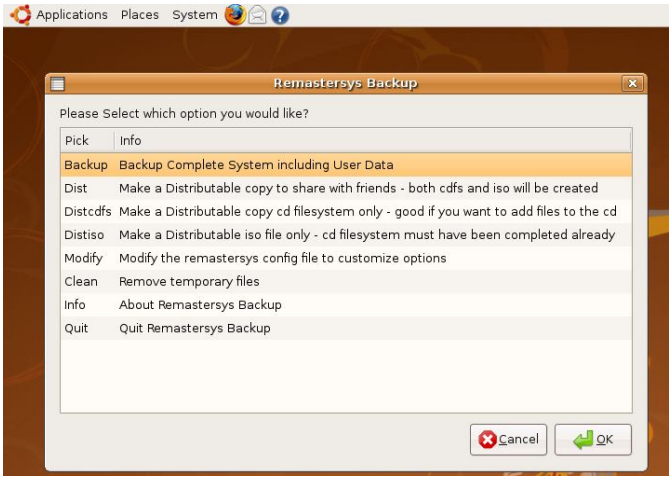


Figura 2. Ventana Remastersys Backup.

Al escoger esta opción se creará el archivo punto iso en la carpeta indicada por la herramienta.

El contenido que debe de tener cada script se muestra continuación:

Script1

```
cp /boot/config-2.6.24-24-generic .config
make oldconfig
make menuconfig
```

```
echo "Kernel 2.6.24.7-rtai Configurado con exito"
make
make modules_install
make install
cp -aR /lib/firmware/2.6.24-24-generic /lib/firmware/2.6.24.7-
rtai
#El siguiente comando es para Ubuntu
update-initramfs -c -k 2.6.24.7-rtai
update-grub
sed ‘/scriptRtai1/d’ /root/.bashrc > tmp
mv tmp /root/.bashrc
echo "sh /opt/RtaiComediScilabInstallFolder/scriptRtai2" >>
/root/.bashrc
zenity --info --text “Se reiniciara la computadora \npara
continuar con la instalacion \nPor favor ingrese al kernel:\n
2.6.24.7-rtai”
```

Script2

```
#!/bin/bash
#Script 2 de instalacion de Rtai
#Instalacion de Comedilib
pathinsta=$(pwd)
echo $pathinsta
cp comedilib-0.8.1.tar.gz /usr/local/src/
cp comedi-0.7.74.tar.gz /usr/local/src/
cd /usr/local/src
tar xzvf comedi-0.7.74.tar.gz
tar xzvf comedilib-0.8.1.tar.gz
mv comedi-0.7.74 comedi
mv comedilib-0.8.1 comedilib
cd comedilib
sh autogen.sh
./configure --sysconfdir=/etc
make
make install
make dev
echo "Comedilib correctamente instalado"
```

#Instalacion Rtai parte 1

```
cd /usr/src/rtai
clear
```

```
echo "Verifique los valores del menu de Rtai"
sleep 6
make menuconfig
make
make install
echo "PATH=$PATH:/usr/realtime/bin" >> /root/.bashrc
echo "export PATH" >> /root/.bashrc
sleep 3
sed ‘/scriptRtai2/d’ /root/.bashrc > tmp
mv tmp /root/.bashrc
echo "sh /opt/RtaiComediScilabInstallFolder/scriptRtai3" >>
/root/.bashrc
zenity --info --text “Se reiniciara la computadora \npara
continuar con la instalacion \nPor favor ingrese al kernel:\n
2.6.24.7-rtai”
```

Script3

```
#!/bin/bash
# Script de prueba de Rtai
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
cd /usr/realtime/testsuite/kern/latency
./run
insmod /usr/realtime/modules/rtai_hal.ko
insmod /usr/realtime/modules/rtai_up.ko
insmod /usr/realtime/modules/rtai_fifos.ko
insmod /usr/realtime/modules/rtai_lxrt.ko
cd /usr/src/rtai/testsuite/kern/latency/
insmod latency_rt.ko period=1000000
./display
rmmod latency_rt
sed ‘/scriptRtai3/d’ /root/.bashrc > tmp
mv tmp /root/.bashrc
echo "sh /opt/RtaiComediScilabInstallFolder/scriptRtai4" >>
/root/.bashrc
zenity --info --text “Se reiniciara la computadora \npara
continuar con la instalacion \nPor favor ingrese al kernel:\n
2.6.24.7-rtai”
```

Script4

```
#!/bin/bash
# Script 4 de Instalacion de Rtai
#Instalacion de Comedi
cp lmbrtai /etc/init.d/
cp lmbrtai2 /etc/init.d/
cd /usr/local/src/comedi
sh autogen.sh
./configure       --with-linuxdir=/usr/src/linux       --with-
rtaidir=/usr/realtime --enable-kbuild --disable-pcmcia
make
make install
depmod -a
make dev
cp       include/linux/comedi.h       include/linux/comedilib.h
/usr/include/
cp       include/linux/comedi.h       include/linux/comedilib.h
/usr/local/include/
ln -s /usr/include/comedi.h /usr/include/linux/comedi.h
ln -s /usr/include/comedilib.h /usr/include/linux/comedilib.h
insmod /usr/realtime/modules/rtai_hal.ko
sleep 1
insmod /usr/realtime/modules/rtai_up.ko
sleep 1
insmod /usr/realtime/modules/rtai_fifos.ko
sleep 1
insmod /usr/realtime/modules/rtai_sem.ko
sleep 1
insmod /usr/realtime/modules/rtai_mbx.ko
sleep 1
insmod /usr/realtime/modules/rtai_msg.ko
sleep 1
insmod                    /usr/realtime/modules/rtai_netrpc.ko
ThisNode="127.0.0.1"
sleep 1
insmod /usr/realtime/modules/rtai_shm.ko
sleep 1
insmod /usr/realtime/modules/rtai_signal.ko
sleep 1
insmod /usr/realtime/modules/rtai_tasklets.ko
sleep 1
modprobe comedi
```

```
sleep 1
modprobe kcomedilib
sleep 1
modprobe comedi_fc
sleep 1
modprobe 8255
sleep 1
modprobe ni_pcimio
sleep 1
zenity --info --text “Editar el siguiente archivo en caso de que no se tenga una tarjeta PCI”
sleep 5
gedit /opt/RtaiComediScilabInstallFolder/scriptRtai5
sleep 5
sh /opt/RtaiComediScilabInstallFolder/scriptRtai5
update-rc.d lmbrtai defaults
update-rc.d lmbrtai2 defaults
cd /usr/src/rtai
clear
echo "Seleccionar COMEDI support over LXRT"
echo "Escriba /usr en  Comedi installation directory"
echo "Seleccionar RTAI Lab"
echo "Escriba /usr/local en EFLTK installation direcotry"
echo "El proceso comenzara en 8 segundos"
sleep 9
make menuconfig
make
make install
echo "Por favor reinicie el computador en el kernel de rati"
sed ‘/scriptRtai4/d’ /root/.bashrc > tmp
mv tmp /root/.bashrc
echo  "sh  /opt/RtaiComediScilabInstallFolder/scriptRtai6"  >> /root/.bashrc
zenity --info --text “Se reiniciara la computadora \npara continuar con la instalacion \nPor favor ingrese al kernel:\n 2.6.24.7-rtai”
```

Script5

```
#!/bin/bash
# Script 5 de Instalacion de Rtai
comedi_config -v /dev/comedi0 ni_pcimio
```

Script6

```
#!/bin/bash
# Script 6 de Instalacion de Rtai
#Instalacion de Scilab
cp scilab-4.1.2-src.tar.gz /usr/local/
cd /usr/local/
tar xzvf scilab-4.1.2-src.tar.gz
cd /usr/local/scilab-4.1.2/
./configure --without-java --with-tcl-library=/usr/lib --with-
tcl-include=/usr/include/tcl8.4
make all
ln -s /usr/local/scilab-4.1.2/bin/scilab /usr/local/bin/scilab

# Instalacion Addons Rtai-Scilab
cd /usr/src/rtai/rtai-lab/scilab/macros
make install
make user
echo "* Instalacion de Rtai ha concluido *"
#Fin de Instalacion
sed ‘/scriptRtai6/d’ /root/.bashrc > tmp
mv tmp /root/.bashrc
```