

WIKI

LINUX

Willkommen

Grabación de voz en
una memoria EPROM.

Sistema mínimo con un 80C31

Caracterización
de un diodo túnel

Función SLEEP con
un 18F452 en PICC

Bienvenue

ようこそ

歓迎

BLASTED ROBOTICS

Año 1 - Abril 2009 - Número 4

Í n d i c e

爆風ロボット

Grabación de voz en una memoria EPROM 27c010A

3

LINUX: ¿Y eso qué es?

7

Caracterización de un Diodo Túnel

25

Sistema Mínimo con un 80C31

33

Función Sleep con un 18F452 en PICC

40

Artículo de tecnología. Fuera teclas

44

Trivia

45

Publicación del grupo estudiantil
Blasted Robotics

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica.

Edición:

Miguel Ángel San Pablo Juárez
Raúl García Chetla

Revista WIKI Número 4 -
Abril de 2009

blastedrobotics@yahoo.com.mx

La revista tiene como finalidad publicar artículos hechos por la comunidad estudiantil de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla, para apoyar a nuestros compañeros difundiendo un poco de la creatividad desarrollada y aplicada en el área de la electrónica. El lector podrá construir y probar los diferentes proyectos y prácticas descritas en esta publicación teniendo la certeza de que funcionarán al cien por ciento, todos los artículos han sido diseñados, innovados y probados en los laboratorios de nuestra universidad y otros varios en casa o lugares específicos de diseño como pasatiempo de algunos autores.

En este cuarto número de nuestra revista aportamos un poco de circuitos útiles en electrónica, comenzando con la grabación de voz en una memoria, el cual suele darnos problemas por no saber el procedimiento necesario. Se explica también sobre un diodo que pocas veces utilizamos, pero que tiene gran impacto en la electrónica, me refiero al diodo túnel; aquí explicamos cómo caracterizarlo. Se adentra sobre un tema que sigue siendo un tabú en nuestros días, como lo es el uso de otros sistemas operativos, aquí comenzamos a desmentir la mala popularidad que algunos le hacen y consejos prácticos para un mejor aprovechamiento. Algo que suele hacernos la vida de cuadritos, los sistemas mínimos, aquí vemos la manera de hacer uno con un 80C31.

Esperamos disfruten de los artículos y que les sea de utilidad, de antemano gracias.

Blasted Robotics.

Grabación de voz en una memoria EPROM 27c010A

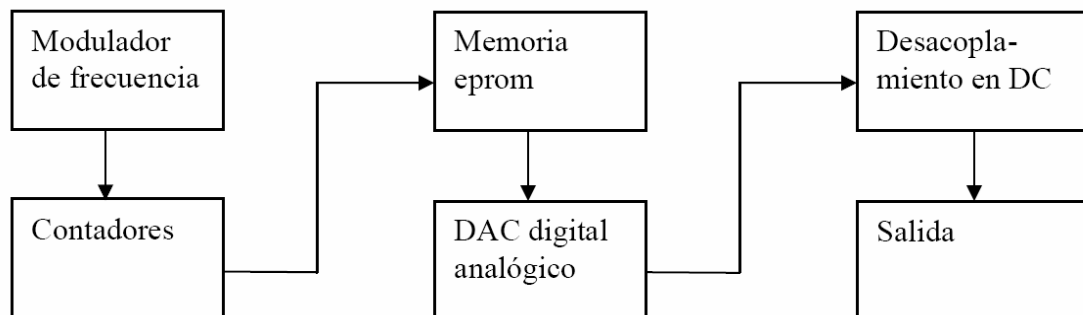
Introducción

Se trata de un reproductor de voz en donde se sabe que la frecuencia de la voz esta en un rango de 100hz a 4KHz aunque la percepción del sonido para el oído humano es de un rango mayor el cual es de

0Hz a 20KHz.

Objetivo

Reproducir una señal de audio frecuencia (voz), usando una memoria programable.



Desarrollo

Se utilizara el siguiente material:

2 CI CD4040B.

1 CI MC 14046.

1 memoria eprom 27c010A.

18 resistencias de 10KΩ.

1 capacitor electrolitico 10μF

2 capacitores cerámicos de 10nF.

1 capacitor cerámico de 0.001μF

1 potenciómetro de 100KΩ

1 protoboard

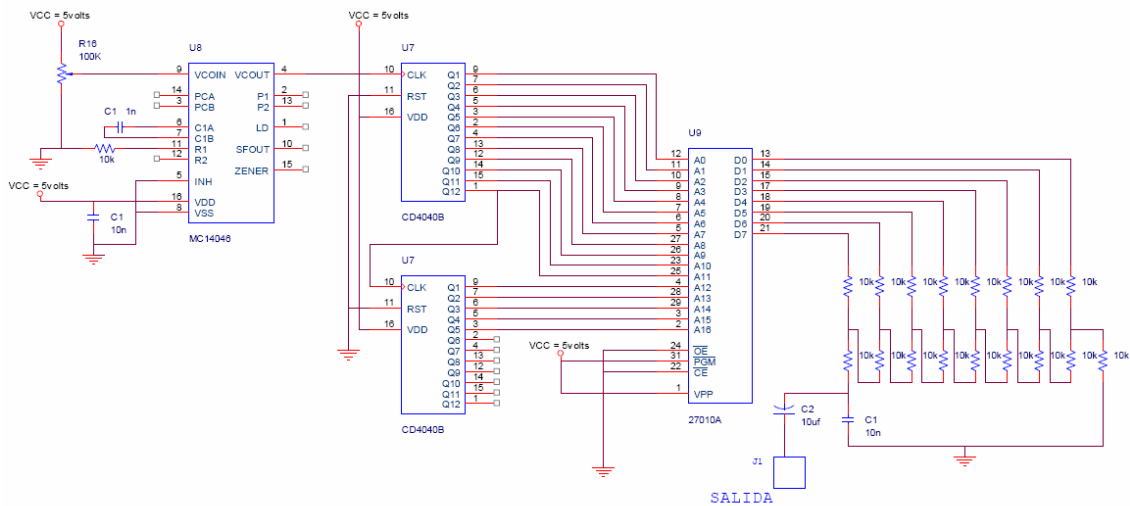
Como primer paso, se debe grabar un mensaje con un tiempo de de duración de 10 segundos como máximo. Haciendo uso del grabador de Windows.

Guardar el archivo con extensión (.wav); después guardar el mismo archivo pero ahora con extensión (.bin).

El archivo con extensión (.bin) es el que se usará para grabarlo en la memoria EPROM o EEPROM.

El circuito integrado MC14046 debe usarse como un VCO. Donde el potenciómetro ajusta el muestreo en un rango de 100Hz a

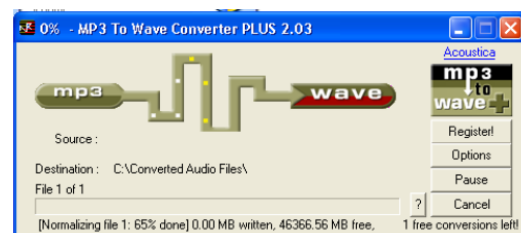
4KKz. En la etapa de salida se puede anexar un amplificador de audio.



Grabación del archivo de sonido

Se grabara un sonido que no contenga ruido, en este caso es una fragmento de un poema el cual ya esta libre de ruido, pues si grabamos un mensaje corto de voz utilizando un micrófono en el proceso de grabación se filtrara ruido y en el momento de reproducirlo en nuestra memoria se tendrá una salida con saturación de ruido.

Nuestro poema esta en formato .mp3 para reproducirlo con el grabador de voz de Windows es necesario convertirlo en formato .wav para esto emplearemos el programa Acústica Mp3 to convert wave plus.



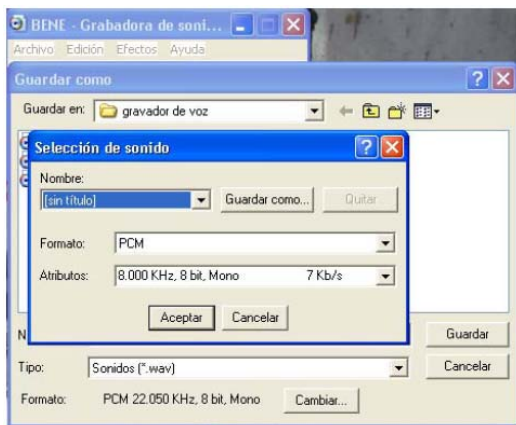
Una vez obtenido el archivo de sonido convertido en formato .wav, lo reproduciremos con el grabador de sonidos de Windows.



A continuación solo se tomara un fragmento de la reproducción con ayuda del menú edición > eliminar hasta la posición actual, o eliminar a partir de la posición actual.



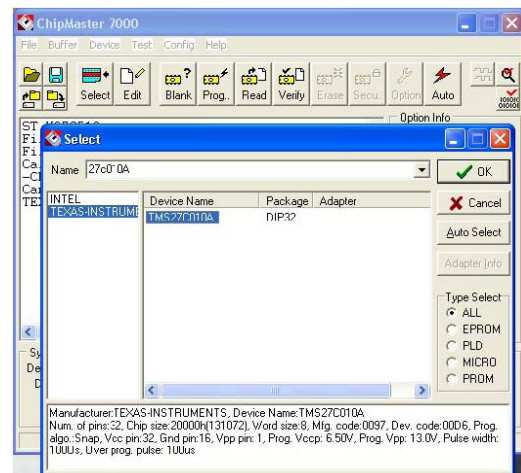
Una vez obtenido esto se debe guardar con las siguientes especificaciones PCM 8.000 KHz, 8 bits, mono con el formato (.wav).



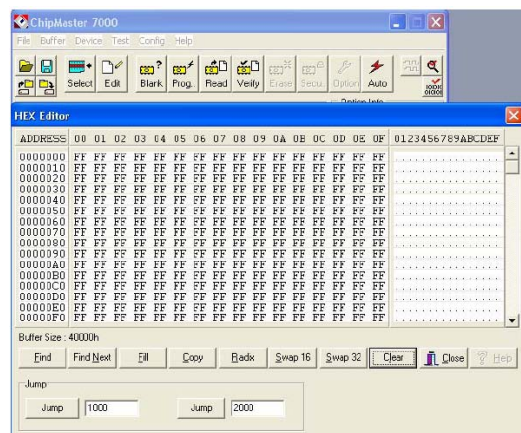
Nuevamente se vuelve a guardar con la extensión (.bin) este ultimo se grabara en la memoria EPROM.

Grabación en la memoria eprom

Utilizaremos el programa ChipMaster 7000. Como primer paso es seleccionar el dispositivo en el menú Device > Select seleccionamos la memoria que será grabada.

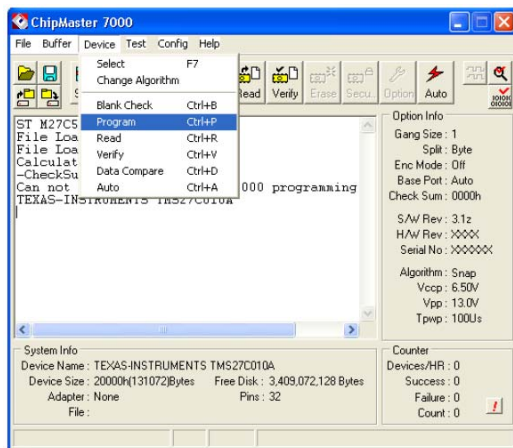


Se verifica que esta borrada; para esto damos clic en el menú Device-Read después clic en el icono Edit. Debemos observar que contenga letras FF.



En caso contrario se debe borrar la memoria si es EPROM, el cual es nuestro caso se coloca la memoria a luz ultravioleta un tiempo mínimo de 30 minutos.

Por ultimo se da clic en el menú Device > Program.



Una vez programada la memoria ya esta lista para el armado de el circuito.

Ahora se procede a armar el circuito de acuerdo al diagrama eléctrico.

Conclusión

En la primera etapa de nuestro circuito armado haciendo uso del osciloscopio se ajusta se observa a la salida del integrado MC14046 una frecuencia de 8.103Khz esto es el resultado esperado debido a que fue la frecuencia con la cual guardamos nuestro archivo de sonido.

Nuestra segunda etapa es observar la salida de los contadores es necesario conectar los dos contadores en cascada, para que estos puedan barrer toda la memoria y garantizar que no se recorte nuestra grabación; de no hacer esto el mensaje se recorta demasiado.

En la tercera etapa se utilizara un arreglo de resistencias las cuales serán nuestro convertidor digital-analógico; a este arreglo se le conoce como escalera.

En nuestra última etapa de salida es desacoplar el nivel de DC utilizando un capacitor electrolítico el cual filtra el ruido dejando pasar solo el nivel de DC y por ultimo es observar la forma de onda de la voz.

Angel Donado Arenas

Dudas: angeldonado@gmail.com

LINUX: ¿Y eso qué es?

Introducción a UNIX

Este artículo no explica cómo usar UNIX, (el cual, aclararemos que es un sistema de cómputo que se emplea en todo el mundo y que se ejecuta virtualmente en cualquier tipo de computadora). Sin embargo te ayudará a familiarizarte rápidamente con algunas de sus herramientas y aplicaciones, además de que te orientará en conceptos básicos.

Un programador desarrolló el primer sistema UNIX en 1969; en aquel entonces pudo ejecutar un programa denominado Space Travel.¹ Unix es hoy en día nada menos que una cultura mundial que abarca múltiples herramientas, ideas y costumbres.



Ken Thompson y Dennis Ritchie

¹ Space Travel simulaba el movimiento del sol y los planetas, así como los de una nave espacial que se podía aterrizar en sitios diferentes. El programador fue Ken Thompson, quien, con varias personas más de los laboratorios Bell de AT&T, desarrolló el sistema operativo UNIX inicial (presumiblemente después de que se cansaron de Space Travel).

Todos los detalles de Unix, desde hace mucho tiempo, más que un simple sistema de cómputo, trascienden el entendimiento. En otras palabras, no hay quien sepa todo sobre Unix; de hecho, no existe quien sepa casi todo acerca de Unix.

La cultura Unix, a la cual estás a punto de entrar, contiene un enorme número de herramientas a su disposición. Es posible crear y manipular información de más maneras de las que pueda imaginar; puede enviar mensajes y hablar con personas en cualquier lugar del planeta; puede jugar, crear documentos y escribir programas de cómputo propios.

Para usar Unix todo lo que necesita es tener acceso a una computadora Unix, de preferencia conectada a Internet. Una vez que cuente con tal acceso, las utilerías de Unix le estarán esperando. Lo único que necesita es aprender a usar el sistema y a participar en la cultura.

Este artículo te enseñará lo que necesitas saber en un principio y mucho más.

Aunque Unix (e Internet) quizá le parezcan baratos, debe saber que alguien, en algún sitio, paga mucho dinero y dedica gran

cantidad de tiempo para dar mantenimiento al sistema. Parte del aprendizaje de Unix consiste en entender qué significa ser un usuario responsable, honesto y amable; después de todo, compartimos recursos limitados.

El idioma Unix

En todo el mundo, el primer idioma de Unix es el inglés estadounidense. Sin embargo, la documentación y los sistemas Unix se han traducido a muchos otros idiomas, por eso no es necesario saber inglés, siempre y cuando el sistema que tenga comparta el idioma con usted. No obstante, cuando explore la red mundial basada en Unix, tal vez descubra que mucha de la información está en inglés.

Por otra parte, la comunidad Unix ha introducido diversos neologismos de creación propia. En este artículo (y a los posteriores a este), pondremos atención especial a tales términos. Cada palabra nueva se explica y define tan pronto surge en el texto. Para facilitar la referencia, todas las definiciones se agruparon un glosario al final del artículo.

Cuando aparece un nombre con una historia particularmente interesante, se brinda una explicación especial dentro de un cuadro similar al siguiente:

Qué significa... Unix

En la década de los sesenta, un número de investigadores de los laboratorios Bell (una parte de AT&T) trabajaron en el MIT en un proyecto llamado Multics, un antiguo sistema operativo de tiempo compartido. Multics fue un esfuerzo conjunto que involucró a programadores del MIT, GE y laboratorios Bell. El nombre "Multics" era un acrónimo de Multiplexed Information and Computing Service (Servicio de cómputo e información multiplexada).

Hacia el fin de los sesenta, la dirección de los laboratorios Bell decidió no continuar con el proyecto Multics, así que retiró a sus investigadores. En 1969, uno de ellos, Ken Thompson, desarrolló un sistema operativo pequeño y simple para una minicomputadora PDP-7. Al buscarle el nombre, Thompson comparó el nuevo sistema con Multics.

La meta de Multics era ofrecer múltiples herramientas a muchos usuarios al mismo tiempo. Por eso era grande, inmanejable y tenía muchos problemas.

El sistema de Thompson era más pequeño, menos ambicioso y (al menos en un principio) sólo podía usarlo una persona. Cada parte del sistema estaba

diseñada para hacer una sola cosa y hacerla bien. Thompson decidió llamar a su sistema Unics, nombre que pronto cambio a Unix. En otras palabras, el nombre de Unix es un retruécano del nombre de Multics.



Lo que se da por hecho

Estos artículos asumirán dos presupuestos importantes en cuanto al tipo de sistema Unix que uses.

Primero, existen muchas versiones de Unix basadas en dos variantes principales: Unix de Berkeley y System V.² Los sistemas Unix modernos combinan los elementos más importantes de ambos tipos de Unix. Por ello, casi nunca importa qué tipo de Unix utilices.

Este artículo está orientado a las necesidades de quienes usan Unix, no a un

² Hasta hace algunos años los sistemas Unix basados en lo desarrollado en la Universidad de Berkeley eran los más populares; sin embargo, la tendencia actual es desarrollar sistemas Unix basados en System V.

tipo específico de Unix. En algunos aspectos del trabajo que hagas, tendrás que elegir entre la funcionalidad de Berkeley o del System V. En tales cosas se optará por las convenciones de Berkeley, pues son las más populares.

Lo que no se da por hecho

Si usted es un usuario de computadoras experimentado que desea aprender Unix, este libro le proporciona los fundamentos, además de darle sólidos conocimientos en todas las áreas importantes.

No obstante, no se asume que usted tiene experiencia previa. No importa que nunca haya usado una computadora. No es necesario que aprenda todo acerca de Unix; tampoco que sea un programador, o un experto en electrónica o matemáticas.

¿Qué es UNIX?

En este artículo te encaminarás hacia el mundo de Unix, es decir: aprenderás qué hace, dónde se le encuentra, sus variantes y la cultura que ha surgido alrededor de él.

Para sentar los antecedentes, primero se describe el componente fundamental de cualquier sistema de cómputo: el sistema operativo.

¿Qué es un sistema operativo? Las computadoras ejecutan tareas automáticamente, siguiendo instrucciones. A una lista de instrucciones se le denomina PROGRAMA. Un SISTEMA OPERATIVO es un complejo programa de control maestro cuya función principal es usar con eficiencia el hardware. Para lograrlo, el sistema operativo actúa como la interfase primaria con el hardware. El sistema operativo te ayuda a ti (a hacer el trabajo) y a los programas (mientras éstos se ejecutan).

Cuando tú trabajas, el sistema operativo está presente, esperando servirte y listo para administrar los recursos de la computadora. Las tareas más importantes a su cargo son: controlar el uso de la memoria de la computadora, mantener un sistema de almacenamiento de archivos, planear el trabajo y proporcionar servicios de contabilidad y seguridad.

La mayoría de los sistemas operativos vienen con una gran variedad de programas a su disposición. Un sistema operativo Unix trae consigo, literalmente, cientos de programas; cada uno es una herramienta para ejecutar una tarea específica.

¿UNIX puede referirse a un sistema operativo específico? En líneas anteriores

se describió cómo un solo programador desarrolló el primer sistema Unix (de hecho, un sistema operativo antiguo). Desde entonces, un gran número de personas ha desarrollado Unix hasta convertirlo en una moderna familia de sistemas operativos.

Por muchos años, los laboratorios Bell fueron unos de los centros de desarrollo de Unix. En 1990 AT&T formó una organización nueva, denominada Unix Systems Laboratory (USL) para proseguir el trabajo. En junio de 1993 T&T vendió USL a Novell Corporation. En octubre del mismo año Novell transfirió los derechos del nombre "Unix" a X/Open, una organización de estándares internacionales. Entonces el nombre "UNIX" hizo referencia al producto de AT&T. Hoy en día usamos el término más genérico, "Unix", para describir cualquier sistema operativo que satisfaga ciertos estándares.

El descendiente directo más moderno del UNIX de AT&T original es System V, versión 4, a menudo llamado System V.4.

¿UNIX puede referirse a una familia de sistemas operativos? Desde la década de los setenta, se han desarrollado muchos otros sistemas operativos Unix. Por ser una familia, éstos comparten dos características importantes: son sistemas

de tiempo compartido, multitarea y multiusuario. MULTITAREA significa que un sistema Unix puede ejecutar más de un programa simultáneamente; MULTI-USUARIO que Unix puede aceptar más de un usuario al mismo tiempo.

Cuando la gente habla de Unix como sistema operativo se refiere a cualquier sistema de apariencia Unix. En este sentido, la palabra "Unix" se refiere a cualquier miembro de la familia de sistemas operativos similares a Unix.

Uno de los Unix más importantes proviene de la Universidad de Berkeley, en California. Al principio, el Unix de Berkeley se basó en el UNIX de AT&T. De cualquier modo, la versión más nueva se diseñó para ser lo más libre posible de la programación del System V del UNIX de AT&T.

El nombre oficial del Unix de Berkeley es BSD, acrónimo que significa Berkeley Software Distribution (Distribución del Software de Berkeley). Así, aunque parezca extraño, "BSD", al igual que "Unix", es el nombre de un sistema operativo. La versión más reciente de este sistema operativo es BSD 4.4.

Aunque existen muchos tipos de Unix, casi todos se basan en BSD, en System V, o en ambos. En las comunidades académicas y de investigación, la mayoría de las

personas usan un Unix basado en ambas vertientes, aunque con cierto énfasis en el Unix de Berkeley.

La figura 1 muestra los nombres de los sistemas Unix y parecidos a Unix que puede encontrar. El más común es Solaris, o SunOS (el nombre antiguo), ejecutándose en una computadora de Sun Microsystems. Todos estos Unix derivan, al menos en parte, del Unix de los Berkeley y del System V. Ésta, por supuesto, no es una lista completa.

Qué significa... BSD

El Unix de Berkeley no comenzó con un nombre típico. La versión original (una variación del UNIX AT&T) se conoció como "la primera distribución del software de Berkeley", que se abreviaba 1BSD. La siguiente versión importante se conoció como 2BSD y así sucesivamente.

Hoy en día el Unix de Berkeley ha comenzado su propia vida y, como sistema operativo, se le conoce con el nombre de BSD. Por lo tanto un purista podría decir que la versión más nueva debe llamarse en realidad 4.4BSD, no BSD 4.4.



Existen muchas variaciones de Unix, pero por ahora no te preocupes por conocer todos los tipos. En términos generales Unix es Unix, y lo que aprenderás de estos artículos será importante sea cual sea el tipo de Unix que utilices.

Nombre de UNIX	Compañía u organización
386BSD	Gratuito en Internet *
AIX	IBM
A/UX	Apple
BSD	Universidad de Berkeley, en California
BSD-Lite	Universidad de Berkeley, en California
BSD/386	Berkeley Software Design (BSDI)
Coherent	Compañía Mark Williams
Dynix	Sequent
FreeBSD	Gratuito en Internet *
HP-UX	Hewlett-Packard (HP)
Hurd (GNU)	Free Software Foundation
Interactive Unix	Sun Microsystems
Irix	Silicon Graphics
LINUX	Gratuito en Internet *
Mach	Universidad Carnegie-Mellon
Minix	Incluido en el libro de Andy Tanenbaum
MKS Toolkit	Mortice Kern Systems
NetBSD	Gratuito en Internet *
Nextstep	Next
OSF/1	Digital Equipment Corporation (DEC)
SCO Unix	Santa Cruz Operation
Solaris	Sun Microsystems
System V Unix	Diversas versiones para computadoras
Ultrix	Digital Equipment Corporation (DEC)
Unicos	Cray Research
Unixware	Novell

Figura 1. Tipos de Unix y organizaciones que los desarrollan

* Software de dominio público

A menudo la gente pregunta: “¿Cómo saber si un sistema operativo es Unix?”. Aunque no existe un consenso universal, la mejor respuesta es: si un sistema operativo parece Unix cuando usted trabaja con él, y si los programas en uso parecen Unix al momento de ejecutarse, entonces es Unix. Existen más respuestas técnicas a esta pregunta, pero en términos reales, Unix es todo lo que parezca Unix.

UNIX es el nombre de una cultura

Unix significa mucho más que una simple familia de sistemas operativos. Unix quiere decir redes de cómputo, correo electrónico, una gran cantidad de programas (incluyendo juegos) y una cultura muy real basada en estas herramientas. La tercera y más importante acepción de la palabra “Unix” es como sinónimo de la comunidad global descrita anteriormente, es decir, la comunidad que encarna la cultura Unix.

Algunas personas van más lejos en sus reflexiones y considera que Unix es una idea abstracta: una filosofía real y aplicada que dictamina un enfoque particular para solucionar problemas. Esto cobrará mayor sentido para ti mientras más aprendas. Al usar Unix, conocerá el enfoque y resolverá problemas combinando programas

simples, como si construyera con bloques hasta formar estructuras elegantes.

Quizá la mejor definición de Unix (y la que es preferible que recuerdes al leer este artículo) es la siguiente: Unix es un conjunto de herramientas para gente inteligente.

Cómo usar Unix

Usar Unix significa interactuar con una computadora mediante un teclado, un desplegado en pantalla y, posiblemente un ratón. Es decir, mientras teclee (o mueva y haga clic con el ratón), observará la pantalla. De vez en cuando, quizá sea necesario imprimir algo sobre papel.

Para iniciar una sesión de trabajo “entras” al sistema. Al terminar de trabajar, “sales” del sistema. El funcionamiento de esto se explicará en artículos posteriores. Básicamente, para entrar al sistema debe teclear un nombre y una contraseña. Una vez que Unix verifique que te está permitido usar el sistema de cómputo, puedes comenzar a trabajar. Cuando sales del sistema Unix entiende que la jornada ha concluido y que estás listo para detenerte por ese día.



*Bueno, hasta ahora hemos visto qué es Unix, algunas de sus características que lo distinguen como plataforma única, pero muchos se preguntarán sobre el principal motivo del artículo: **aclarar qué es LINUX**. Para esto, vamos a ver un poco de sus antecedentes y cómo ha evolucionado a lo largo de los años desde sus primeras versiones, hasta algunas de las últimas.*

Será entonces necesaria la aclaración de que LINUX y Unix no es lo mismo, en realidad LINUX es un sistema UNIX, el cual tiene una gran cantidad de distribuciones comerciales y gratuitas. Veremos entonces, cómo se han desarrollado algunas de sus distribuciones más conocidas (las cuales si serán abordadas en algunos de los artículos posteriores a este), y cómo han influido hasta nuestros días.

GNU

El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU. El 27 de septiembre de 1983 se anunció públicamente el proyecto por primera vez en el grupo de noticias net.unix-wizards. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman como el "Manifiesto GNU", que establecieron sus motivaciones para realizar el proyecto GNU, entre las

que destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

GNU es un acrónimo recursivo que significa GNU is Not Unix (GNU No es Unix). Puesto que en inglés "gnu" (en español "ñu") se pronuncia igual que "new", Richard Stallman recomienda pronunciarlo tal y como se deletrea (G-N-U) para su mejor comprensión.

UNIX es un Sistema Operativo **no libre** muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable. El sistema GNU fue diseñado para ser totalmente compatible con UNIX. El hecho de ser compatible con la arquitectura de UNIX implica que GNU esté compuesto de pequeñas piezas individuales de software, muchas de las cuales ya estaban disponibles, como el sistema de edición de textos TeX y el sistema gráfico X Window, que pudieron ser adaptados y reutilizados; otros en cambio tuvieron que ser reescritos.

Para asegurar que el software GNU permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evitase restricciones

posteriores de los mismos. La idea se conoce en Inglés como copyleft -'copia permitida'- (en clara oposición a copyright -'derecho de copia'-), y está contenida en la Licencia General Pública de GNU (GPL).

En 1985, Richard Stallman creó la Free Software Foundation (FSF o Fundación para el Software Libre) para proveer soportes logísticos, legales y financieros al proyecto GNU. La FSF también contrató programadores para contribuir a GNU, aunque una porción sustancial del desarrollo fue (y continúa siendo) producida por voluntarios. A medida que GNU ganaba renombre, negocios interesados comenzaron a contribuir al desarrollo o comercialización de productos GNU y el correspondiente soporte técnico. El más prominente y exitoso de ellos fue Cygnus Solutions, ahora parte de Red Hat.

En 1990, el sistema GNU ya tenía un editor de texto llamado Emacs, un exitoso compilador (GCC), y la mayor parte de las bibliotecas y utilidades que componen un sistema operativo UNIX típico. Pero faltaba un componente clave llamado núcleo (kernel en inglés).

En el manifiesto GNU, Stallman mencionó que "un núcleo inicial existe, pero se necesitan muchos otros programas para emular Unix". Él se refería a TRIX, que es un núcleo de llamadas remotas a

procedimientos, desarrollado por el MIT y cuyos autores decidieron que fuera libremente distribuido; TRIX era totalmente compatible con UNIX versión 7.

En Diciembre de 1986 ya se había trabajado para modificar este núcleo. Sin embargo, los programadores decidieron que no era inicialmente utilizable, debido a que solamente funcionaba en "algunos equipos sumamente complicados y caros" razón por la cual debería ser portado a otras arquitecturas antes de que se pudiera utilizar. Finalmente, en 1988, se decidió utilizar como base el núcleo Mach desarrollado en la CMU. Inicialmente, el núcleo recibió el nombre de Alix (así se llamaba una novia de Stallman), pero por decisión del programador Michael Bushnell fue renombrado a Hurd. Desafortunadamente, debido a razones técnicas y conflictos personales entre los programadores originales, el desarrollo de Hurd acabó estancándose.

En 1991, **Linus Torvalds** empezó a escribir el núcleo Linux y decidió distribuirlo bajo la licencia GPL. Rápidamente, múltiples programadores se unieron a Linus en el desarrollo, colaborando a través de Internet y consiguiendo paulatinamente que Linux llegase a ser un núcleo compatible con UNIX. En 1992, el núcleo Linux fue combinado con el sistema GNU, resultando en un sistema operativo libre y

completamente funcional. El Sistema Operativo formado por esta combinación es usualmente conocido como "GNU/Linux" o como una "distribución Linux" y existen diversas variantes.

Linus Torvalds

Linus Benedict Torvalds (nacido el 28 de diciembre de 1969 en Helsinki), es un ingeniero de software finlandés; es más conocido por desarrollar la primera versión del núcleo (kernel) del sistema operativo GNU/Linux, basándose en el sistema operativo libre Minix creado por Andrew S. Tanenbaum y en las herramientas, los compiladores y las utilidades desarrollados por el proyecto GNU. Actualmente Torvalds es responsable de la coordinación del proyecto.



También es frecuente hallar componentes de GNU instalados en un sistema UNIX no libre, en lugar de los programas originales para UNIX. Esto se debe a que muchos de

los programas escritos por el proyecto GNU han demostrado ser de mayor calidad que sus versiones equivalentes de UNIX. A menudo, estos componentes se conocen colectivamente como "herramientas GNU". Muchos de los programas GNU han sido también transportados a otros sistemas operativos como Microsoft Windows y Mac OS X.

GNU/Linux

Linux es un término genérico para referirse a sistemas operativos similares a Unix basados en GNU con el núcleo Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la Licencia Pública General de GNU (GNU GPL) y otras licencias libres.

Las variantes de estos sistemas se denominan distribuciones GNU/Linux y su objetivo es ofrecer una edición que cumpla con las necesidades de determinado grupo de usuarios.

Algunas distribuciones GNU/Linux son especialmente conocidas por su uso en servidores y supercomputadoras. No obstante, es posible instalar Linux en una amplia variedad de hardware como

ordenadores de escritorio y portátiles, computadores de bolsillo, teléfonos celulares, dispositivos empotrados, videoconsolas y otros.

Distribuciones GNU/Linux

Una distribución es una variante del sistema GNU/Linux que se enfoca a satisfacer las necesidades de un grupo específico de usuarios. De este modo hay distribuciones para hogares, empresas y servidores. Algunas incorporan programas comerciales (como Mandriva PowerPack) o solamente software libre (como gNewSense).

Las distribuciones son ensambladas por individuos, empresas u otros organismos. Cada distribución puede incluir cualquier número de software adicional, incluyendo software que facilite la instalación del sistema. La base del software incluido con cada distribución incluye el núcleo Linux y las herramientas GNU, al que suelen añadirse también varios paquetes de software.

Las herramientas que suelen incluirse en la distribución de este sistema operativo se obtienen de diversas fuentes, y en especial de proyectos de código abierto o software libre, como GNU y el BSD o GNOME y KDE. También se incluyen utilidades de BSD y de

muchos otros proyectos como Mozilla, Perl, Ruby, Python, PostgreSQL, MySQL, Xorg, casi todas con licencia GPL o compatibles con ésta (LGPL, MPL) otro aporte fundamental del proyecto GNU.

La mayoría de las distribuciones están, en mayor o menor medida, desarrolladas y dirigidas por sus comunidades de desarrolladores y usuarios. En algunos casos están dirigidas y financiadas completamente por la comunidad, como ocurre con Debian GNU/Linux, mientras que otras mantienen una distribución comercial y una versión de la comunidad, como hace RedHat con Fedora, o SuSE con OpenSuSE.



Personajes que distinguen a GNU (el ñu) y a Linux, Tux (el pingüino).

En muchas ciudades y regiones, asociaciones locales conocidas como grupos de usuarios de Linux promueven este sistema operativo y el software libre. Suelen ofrecer conferencias, talleres o

soporte técnico de forma gratuita o introducción a la instalación de Linux para nuevos usuarios.

En entornos de escritorio, "GNU/Linux" ofrece una interfaz gráfica alternativa a la tradicional interfaz de línea de comandos de Unix. Existen en la actualidad numerosas aplicaciones gráficas, ya sean libres o no, que ofrecen la funcionalidad que está permitiendo que GNU/Linux se adapte como herramienta de escritorio.

Algunas distribuciones permiten el arranque de Linux directamente desde un CD/DVD (llamados LiveCDs) sin modificar en absoluto el disco duro del ordenador en el que se ejecuta GNU/Linux. Para este tipo de distribuciones, en general, los archivos de imagen (archivos ISO) están disponibles en Internet para su descarga.

Otras posibilidades incluyen iniciar el arranque desde una red (ideal para sistemas con requerimientos mínimos), desde un disco flexible o disquete o desde unidades de almacenamiento USB.

LINUX

Linux es un proyecto que cada vez gana más terreno en el ámbito de la computación y gracias a las políticas del software libre hacen que este sistema operativo esté más al alcance de todos

para adquirirlo, especialmente en la rama de la educación, ya que es más viable si no se cuenta con los recursos necesarios para hacerse de una licencia de un software propietario.

Se han desarrollado una gran diversidad de herramientas y aplicaciones así como diferentes distribuciones GNU/Linux dedicadas a la educación, lo que hace de la unión entre las herramientas y las aplicaciones de GNU y el núcleo Linux un S.O. casi indispensable, fácil de adquirir, de configurar, de personalizar, brindando además una sólida seguridad a un bajo costo tanto a largo, mediano, como a corto plazo, comparado con otros sistemas operativos distribuidos bajo licencias propietarias. Linux está siendo empleado en las escuelas, universidades, incluso por grandes empresas, y organismos gubernamentales, los cuales se benefician y obtienen un gran desempeño por ser estable, escalable y seguro.

Como creó Linux

En Finlandia, Linus Torvalds, entonces estudiante de informática de la universidad de Helsinki, decidió realizar la cuantiosa inversión de US. \$3500 de la época para adquirir un nuevo PC 386 (33 Mhz., 4MB de RAM; una de las más avanzadas de su época).

Habitualmente lo utilizaba para acceder a la red de su universidad, pero por desgracia no le gustaba nada el sistema operativo con el cual trabajaba, Minix, así que decidió crear uno él mismo.

Como realmente le interesaba aprender el funcionamiento de su nuevo 386, decidió aprovechar esta oportunidad para realizar un programa a bajo nivel prescindiendo de este Minix.

En los primeros intentos consiguió arrancar el ordenador y ejecutar dos procesos que mostraban "AAAAABBBBB". Uno lo utilizaría para leer desde el módem y escribir en la pantalla, mientras que el otro leería desde el teclado y escribiría en el módem. Con el tiempo acabó el programa y lo utilizaba frecuentemente arrancando desde un disquete. La siguiente necesidad que tuvo fue la de poder descargar y subir archivos de su universidad, pero para implementar eso en su emulador era necesario crear un controlador de disco.

Así que después de un trabajo continuo y duro creó un controlador compatible con el sistema de ficheros de Minix. En ese momento se percató que estaba creando algo más que un simple emulador de terminal, así que, descontento con su sistema Minix, emprendió la aventura de crear un sistema operativo partiendo de cero, dando un paso a la creación de Linux.

Aunque Linus Torvalds únicamente creó el Kernel, que representa sólo el 2% del código del Linux actual, en su persona sigue descansando la paternidad de este núcleo del sistema operativo.

Qué significa ... Linux

De forma privada, Linus nombraba "Linux" a su nuevo sistema, pero cuando decidió hacer una presentación pública (pues ya era capaz de mostrar una shell y ejecutar el compilador GCC), pensó que era demasiado egocéntrico llamarlo así y propuso llamarlo Freax. Aunque después se le siguió conociendo como Linux.



Después de anunciar el 25 Agosto de 1991 su intención de seguir desarrollando su sistema para construir un reemplazo de Minix, el 17 de septiembre de 1991 sube al servidor de FTP proporcionado por su universidad la versión 0.01 de Linux con 10.000 líneas de código (en la actualidad tiene más de 24 Millones). A partir de ese momento Linux empezó a evolucionar rápidamente.

Lo que más necesitamos saber sobre Linux

¿Qué es GNU/Linux?

GNU/Linux es, a simple vista, un Sistema Operativo. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta muchos más procesadores, entre ellos Pentium, AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0. Como sistema operativo, GNU/Linux es muy eficiente y tiene un excelente diseño.

Es multitarea, además de multiusuario, multiplataforma y multiprocesador; en las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas; y soporta redes tanto en TCP/IP como en otros protocolos.

Más allá de todas estas notables características técnicas, GNU/Linux es mucho más que lo dicho anteriormente. Lo que hace realmente único a GNU/Linux entre otros sistemas operativos es que conlleva una idea del modo en que debería desarrollarse el software.

Los componentes de un sistema GNU/Linux no están en el dominio público, ni son shareware. Son lo que se llama "software libre". Esto significa que el código fuente está disponible a todo el que lo quiera y siempre lo estará. El software libre puede ser vendido o regalado, a discreción de todo aquel que posea una copia, pero a todo aquel que lo distribuye se le obliga a distribuirlo con el código fuente. Todo esto está reglamentado por la Licencia Pública General GNU (GPL). Esta licencia se encarga de que GNU/Linux permanezca siempre libre.

Linux sigue con el modelo de "desarrollo abierto". Se desarrolla por un grupo no estructurado de programadores de todo el mundo unidos a través de la Internet. Todas las nuevas versiones se liberan al público, se consideren o no con "calidad de producción". El método que se sigue para determinar si se tiene una versión estable o no es el siguiente: Las versiones se numeran como *x.y.z*; aquellas con *y par* son versiones estables; aquellas con *y impar* son versiones de desarrollo. Por ejemplo, la

versión 1.2.13 es estable; la 2.1.44 es de desarrollo. La última versión estable es la 2.2.10. Sin embargo, este modelo de desarrollo abierto mantiene siempre la última versión disponible, y las nuevas versiones se liberan cada pocas semanas. En ocasiones las funciones nuevas contienen errores, pero con cientos de personas en Internet probando y leyendo el código fuente suelen corregirse en cuestión de horas.

Todo esto resulta en un sistema de alta calidad tecnológica, con menos errores que los sistemas comerciales, a un costo cero o muy bajo y con la disponibilidad del código fuente que permite aprender, modificar o ayudar al desarrollo del sistema.

¿Qué hardware se necesita para correr GNU/Linux?

Debido a su eficiente aprovechamiento de recursos, GNU/Linux tiene requisitos de hardware mínimos muy bajos: Una configuración mínima puede ser una 386 SX/16 con 1MB de RAM, y una diskettera (más teclado, tarjeta de vídeo, monitor, etc.). Esto es suficiente para arrancar y entrar al sistema.

Para tener un sistema con todos los comandos importantes y una o dos

aplicaciones pequeñas se requieren alrededor de 10 MB de disco duro.

Para un sistema más completo, se aconsejan 4 MB de memoria, u 8 si se piensa utilizar una interfaz gráfica. Si se vana tener muchos usuarios y/o muchos procesos a la vez, serían aconsejables hasta 16 MB. 32 MB es más que suficiente para cargas pesadas a un máximo rendimiento. En lo que respecta a disco duro, depende de las aplicaciones que se instalen, se va desde los 10 MB básicos hasta los 350 MB de una distribución instalado con varias aplicaciones (incluye compiladores, paquetes de oficina, interfaz gráfica, etc.). Obviamente, un procesador más veloz siempre será ventajoso. El coprocesador matemático nunca es requisito, pero acelera aquellas aplicaciones de cálculo de punto flotante intensivo.

GNU/Linux frente a los otros sistemas operativos

GNU/Linux es una muy buena alternativa frente a los demás sistemas operativos. Más allá de las ventajas evidentes de costo, ofrece algunas características muy notables.

En comparación con las otras versiones de Unix para PC, la velocidad y confiabilidad de GNU/Linux son muy superiores.

También está en ventaja sobre la disponibilidad de aplicaciones, ya que no hay mucha difusión de estos otros Unix (como Solaris, XENIX o SCO) entre los usuarios de PC por sus altos costos.

Comparado con sistemas operativos como los diferentes de Microsoft Windows, GNU/Linux también sale ganando. Los bajos requisitos de hardware permiten hacer un sistema potente y útil de aquel 486 que algunos guardan en un armario. Esta misma característica permite aprovechar al máximo las capacidades de las computadoras más modernas.

Es poco práctico tener una PC con 16 Mb de RAM y ponerle un sistema operativo que ocupa 13 (que es lo que reporta sobre Windows 95 el System Information de Symantec). No solo es superior respecto a el sistema de multitarea y de administración de memoria, sino también en la capacidades de networking (conectividad a redes) y de multiusuario (aún comparando con sistemas multiusuario como NT).

La única desventaja de GNU/Linux frente a estos sistemas, es la menor disponibilidad de software, pero este problema disminuye con cada nuevo programa que se escribe para el proyecto GNU, y con algunas empresas que están desarrollando software comercial para GNU/Linux.



Los falsos mitos sobre GNU/Linux

Hay varios falsos mitos circulando acerca de GNU/Linux y el software libre en general. Usualmente son emitidos como antipropaganda por parte de las empresas perjudicadas por la competencia (táctica llamada FUD), o como prejuicios de gente que en realidad no conoce del tema del que habla. Es muy fácil probar la falsedad de los siguientes mitos:

"GNU/Linux es difícil de usar". En los primeros momentos de GNU/Linux esto era cierto, pero el esfuerzo que se ha dedicado a este tema ha llevado a GNU/Linux a tener interfaces gráficas tanto o más intuitivas que las disponibles en sistemas comerciales. Los procedimientos de instalación también se han vuelto simples gracias a las "distribuciones", paquetes prearmados de GNU/Linux y otras aplicaciones con programas de instalación y soporte.

"GNU/Linux es poco confiable". Los defensores de este argumento dicen que el desarrollo descentralizado impide el control de calidad y las garantías del software no libre. Los hechos han demostrado precisamente lo contrario: las mediciones reportan que los sistemas GNU son más confiables que todos los sistemas operativos más conocidos. El modelo abierto de desarrollo permite que miles de personas efectúen el control de calidad y corrección de errores a la vez; la Internet permite que esas correcciones alcancen los usuarios en tiempos fuera de alcance para una empresa cerrada.

"GNU/Linux no es apto para trabajos grandes" o "GNU/Linux es un clon de UNIX recortado para correr en una PC".

Una vez más, los hechos desmienten esto. Las características tecnológicas superiores de GNU/Linux y la alta confiabilidad mencionada antes han llevado a GNU/Linux a ser la elección de entidades que requieren sistemas críticos para una misión (<http://wauug.erols.com/mclinux/>). Estos usuarios han probado que un sistema GNU/Linux bien mantenido puede funcionar más de 600 días sin caerse ni rearrancarse. GNU/Linux no es un UNIX recortado: Tiene todas las características de un UNIX moderno y evoluciona paralelamente a los últimos desarrollos de la industria del software. Además, no sólo

corre en la PC, sino en muchas otras plataformas importantes.

"Es necesario saber programar para usar GNU/Linux" o "GNU/Linux es un sistema solo para programadores". A pesar de que muchos usuarios de GNU/Linux son programadores interesados en desarrollar software para GNU/Linux o Unix en general, muchos no lo son, al igual que con cualquier otro sistema operativo. Este argumento es tan fundado como decir que para ver televisión es necesario saber electrónica.

"Linux es inseguro". Estrictamente hablando, Linux es un kernel, y un kernel de sistema operativo es inherentemente seguro por no poseer comunicación con el exterior. Las fallas de seguridad vienen a través de programas que proveen servicios de soporte, usualmente de red. Los programas de este tipo (ftp, telnet, servidores de web) que viene en un sistema GNU/Linux, no son software de Linux, sino software de UNIX. Por lo tanto GNU/Linux es tan seguro como otros sistemas UNIX. La seguridad es tomada en serio por la comunidad GNU/Linux. Cuando surgen alertas de problemas de seguridad, las reparaciones suelen hacerse en pocas horas gracias al modelo abierto.

Varios otros argumentos se dan contra el software libre. Las mejores respuestas al

respecto las da Richard Stallman en el manifiesto GNU, disponible en la página: <http://www.gnu.org/gnu/manifesto.html>

Distribuciones más conocidas

En realidad no hay un número definido de distribuciones con las que cuente el grupo Linux, mucho menos las de Unix. Esto se debe a que continuamente surgen nuevas, y las existentes van actualizándose, dando lugar a nuevas distribuciones, además de las variaciones de una misma distribución para dar origen a otra.

Entre algunas de las distribuciones GNU/Linux más conocidas encontramos: OpenSuse, Fedora, Debian, FreeBSD, Ubuntu, Kubuntu, Mandriva, Red Hat, Mandrake, Vixta, Maryan, Sabayon, Wifislax, etc.

Y aunque en apariencia todas parezcan distintas a primera vista, en realidad todas son prácticamente iguales, debido a que en común todas son sistemas UNIX.

En artículos posteriores nos vamos a estar enfocando principalmente en lo que sería OpenSUSE y Ubuntu. También habrá sobre UNIX en sí, pero habrá que recordar, que en ocasiones no todo lo de UNIX será igual de funcional en GNU/Linux; habrá algunas herramientas sí, pero algunas otras no.

¿Quiénes son las mascotas de Linux?



Su nombre es **Tux** y es la mascota oficial. Creado por Larry Ewing en 1996, es un pequeño pingüino de aspecto risueño y cómico.

La idea de que la mascota de kernel de Linux provino del mismo Linus Torvalds.

Tux se ha vuelto un icono para las comunidades de Linux y de software libre.



Es mucho más famoso que su gran amigo, GNU, un ñu pacífico y algo tímido que representa el

proyecto GNU (y más que amigo, algunos lo consideran como el padre de Tux).

Existen otras mascotas de Linux como Beastie, la mascota de BSD, que es considerado como el peor enemigo de Tux.



Además de que cada distribución por lo regular está acompañada de su propia mascota que suele ser el logo, por ejemplo en OpenSUSE es un camaleón.



Raúl Chetla: raulinux.1010@hotmail.com

Caracterización de un Diodo Túnel.

Introducción

El diodo túnel (Tunnel diode ó diodo Esaki) es un diodo de unión PN sumamente impurificado en comparación con un diodo semiconductor común, que tiene una

región estrecha de vaciamiento, lo cual permite que los electrones establezcan un túnel como el de la figura 1.

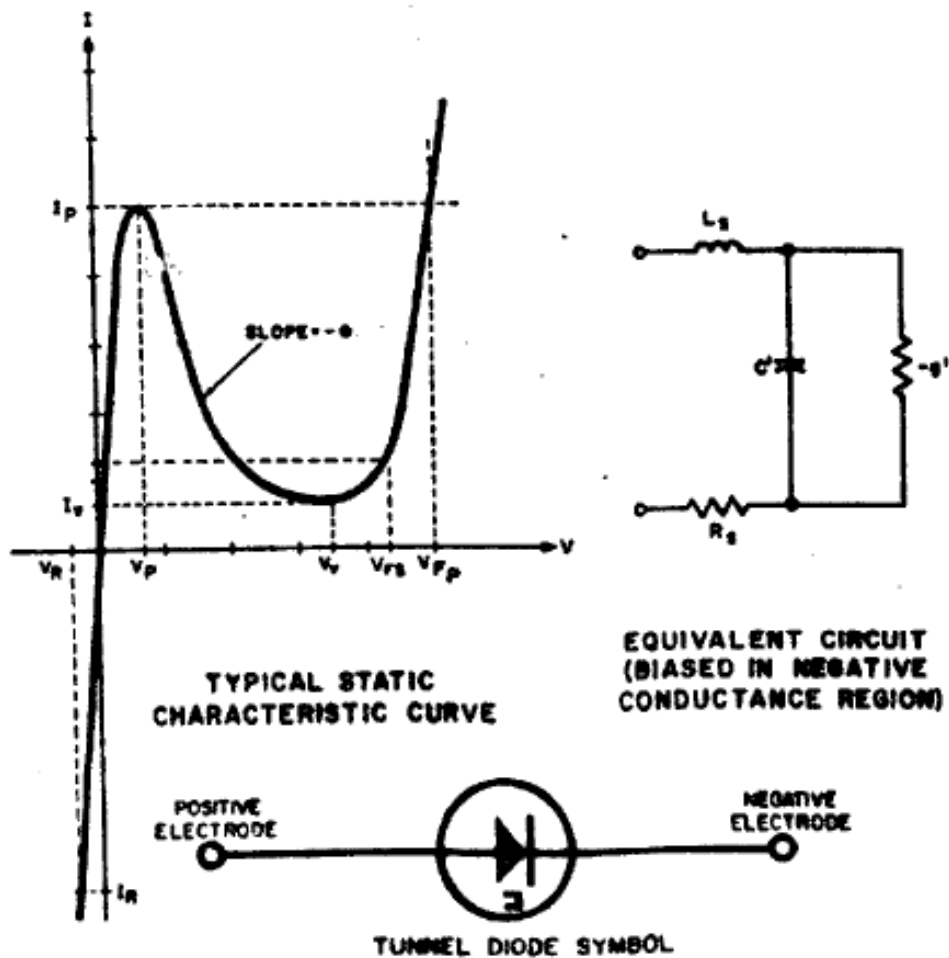


Figura 1. Diagrama de terminales y curva característica del Diodo Túnel.

Los materiales semiconductores que se utilizan con mayor frecuencia en la construcción del diodo túnel son el Germanio y el Arsenuro de Galio. Puede verse que su circuito equivalente es:

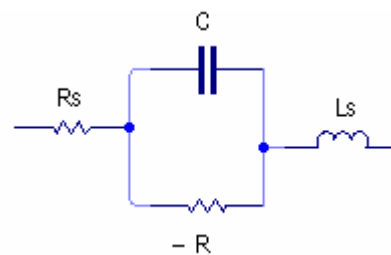


Figura 2. Circuito equivalente.

Donde la inductancia L_s se debe principalmente a las terminales del dispositivo.

A niveles muy bajos de tensión de polarización, el diodo túnel conduce más que el común, ya que hay un flujo apreciable de corriente a través de él, ya sea en sentido directo o inverso, a niveles bajos de polarización. Se caracterizan por sus clasificaciones de muy baja potencia, debido a que reaccionan a esos niveles bajos de tensión.

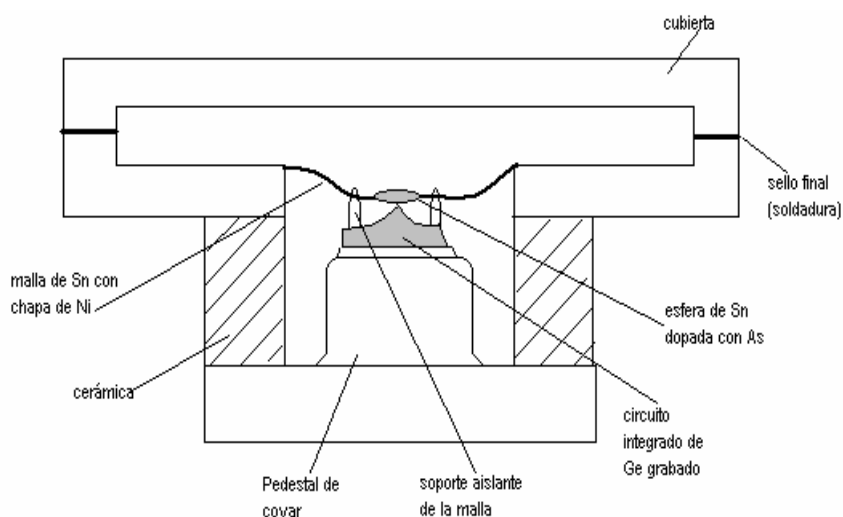
Al aumentar la polarización directa, la corriente a través del diodo de túnel alcanza bruscamente un valor máximo (pico), luego desciende a un valor mínimo suave (valle) y, finalmente, aumenta con la tensión, como sucede en el diodo común. La disminución de corriente al aumentar la polarización positiva, produce una característica de resistencia negativa; esta propiedad le permite convertir la energía de una fuente de cd (DC) en energía de ca

(AC) pudiendo ser empleado como un amplificador u oscilador. En condiciones normales de operación, el diodo túnel funciona entre el punto de tensión del pico y el de tensión del valle de su curva característica. La tensión de la fuente debe regularse correctamente debido a que cualquier variación pequeña en la tensión de entrada cambiará el punto de operación y afectará el funcionamiento del diodo túnel.

Para que el diodo túnel oscile, la resistencia de carga a cd debe ser inferior a la mínima resistencia negativa del diodo, y la línea de carga debe interceptar su curva característica en la región de resistencia negativa.

El dispositivo está construido de la siguiente manera:

Figura 3.
Construcción física
del semiconductor.



Elementos necesarios para la caracterización

Fuente de voltaje 0-20 Vcd ó más.

Osciloscopio.

R₁-----12 ohms, ½ W.

R₂-----47 ohms, ½ W.

R₃-----390 ohms, 1W.

C₁-----0.0022µF

D₁-----Diodo túnel (en este caso el

RCA40571 de American µSemiconductors)

L₁-----Bobina.

Protoboard, Cables, Caimanes, Multímetro & otros.

Práctica

Para obtener la caracterización de la curva característica de nuestro diodo se construyó el siguiente circuito, pero **sin aplicar ninguna tensión inicialmente:**

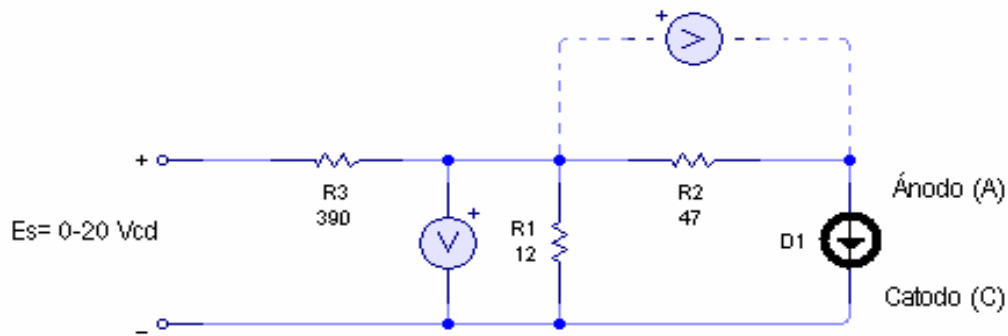


Figura 4. Circuito eléctrico necesario para la caracterización.

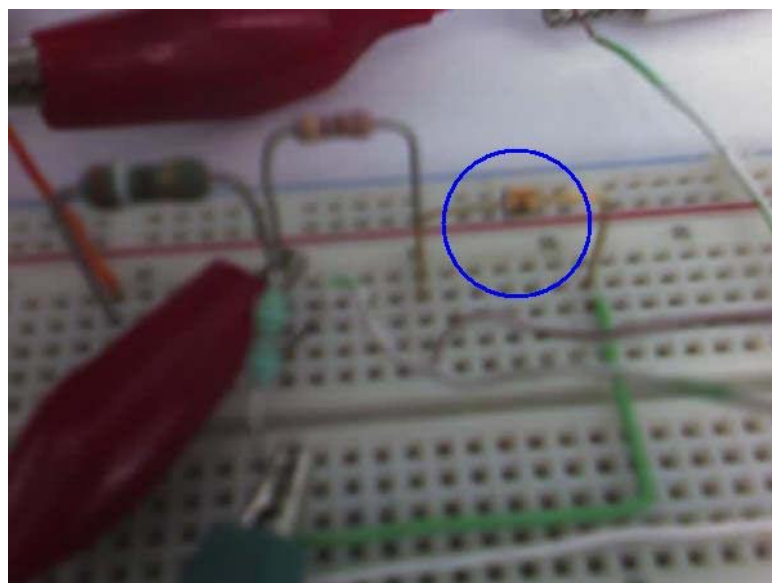


Figura 5. Dispositivo montado en el Protoboard.

Los diodos túnel **son muy sensibles** a la resistencia que se conecta en serie o en paralelo con ellos. En lugar de medir las corrientes con el amperímetro, debe conectarse en serie una resistencia de valor bajo (R_2). La corriente a través del diodo puede medirse, midiendo la caída de tensión en R_2 con un voltímetro.

Se conecta el voltímetro (o el osciloscopio o un multímetro) a R_1 (12ohms) y se ajusta la tensión de la fuente para obtener 25mV en R_1 (E_{R1}), se cambia el voltímetro a R_2 (47ohms) y se anota la caída de tensión en ella (E_{R2}) para el nivel de 25mV, en el espacio correspondiente de la tabla.



Figura 6. Diodo Túnel.

Se ajusta alternativamente la tensión de la fuente a los niveles de tensión que se indican en R_1 (E_{R1}) y luego se anota la caída de tensión resultante en R_2 (E_{R2}) en la columna correspondiente.

La resistencia R_2 el diodo túnel están en paralelo con R_1 , por tanto, la suma de las caídas de tensión en R_2 y el diodo, es igual a la caída de tensión en R_1 .

Se resta la caída de tensión que se registró en R_2 (E_{R2}) de la caída de tensión en R_1 (E_{R1}), para cada uno de los niveles de E_{R1} que están en la tabla, se anotan los resultados en la columna E_{D1} ; la tensión resultante es la caída de tensión en el diodo túnel

$$(E_{D1}) = E_{R1} - E_{R2}.$$

La resistencia R_2 el diodo túnel están en serie, por tanto la corriente que pasa por

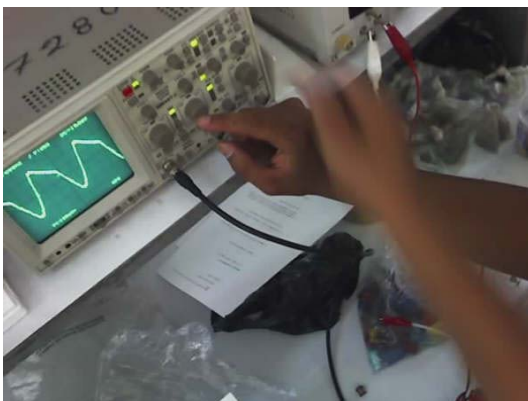


Figura 7. Mediciones en el laboratorio.

R_2 es también la que pasa por el diodo, utilizando la Ley de Ohm, $I_{D1} = E_{R2}/R_2$, se calcula la corriente que circula por R_2 a las caídas de tensión registradas (E_{R2}) y se anota el resultado en el apartado I_{D1} .

La tabla con los resultados medidos y calculados es la siguiente:

Tabla 1.

E_{R1} (mV)	E_{R2} (mV)	E_{D1} (mV)	I_{D1} (mA)
0	0	0	0
25	22.4	2.6	0.476
50	44.7	5.3	0.951
75	66.7	8.3	1.419
100	88.7	11.3	1.887
125	110.5	14.5	2.351
150	131.9	18.1	2.806
175	153	22	3.255
200	174	26	3.702
225	194	31	4.127
250	213	37	4.531
275	229	46	4.872
300	22	278	0.468
325	20	305	0.425
350	19	331	0.404
375	20	355	0.425
400	23	377	0.489
425	28	397	0.595
450	37	413	0.787
475	47	428	1
500	60	440	1.276
525	76	449	1.617
550	92	458	1.957
575	110	465	2.340
600	129	471	2.744
625	149	476	3.170

Haciendo marcas de las caídas de tensión del diodo túnel E_{D1} en función de su corriente I_{D1} para cada uno de los valores vemos el resultado graficado:

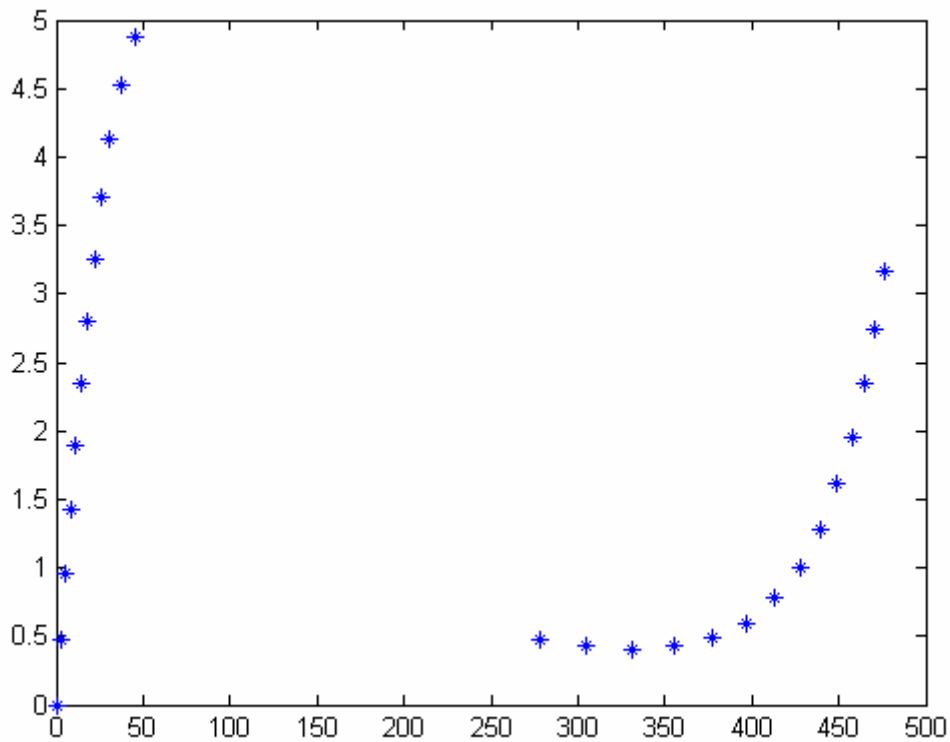


Figura 8. Curva característica del diodo con el que se hizo la medición experimental (x→Voltaje mV vs y→Corriente mA).

Que es exactamente el resultado mencionado en la definición de su curva característica, si graficáramos esos puntos como una línea se vería muy parecida a la curva del inicio de este documento:

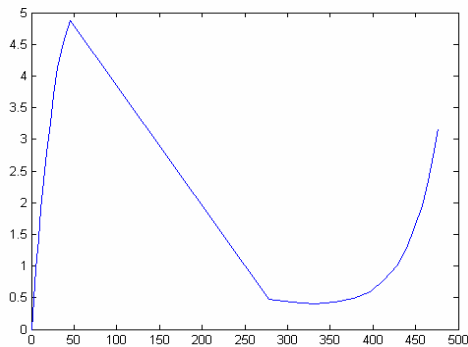


Figura 9. Gráfica experimental del diodo.

Con valores pico de 46 mV y 4.872 mA.

Valores en el valle de 331mV y 0.404 mA.

Entonces por Ley de Ohm, se calcula la resistencia del diodo en el pico y en el valle:

$R_{pico}=9.44$ ohms

$R_{valle}=8.19$ ohms

Aplicación

Si el diodo túnel funciona a nivel de tensión comprendido entre los valores de pico y de valle, a lo largo de la porción de resistencia negativa de la curva característica, entonces podrá actuar como oscilador.

En la figura se muestra el diagrama del circuito oscilador armado con el diodo túnel: *no* se aplica ninguna tensión todavía de alimentación, L_1 puede ser una bobina de inductancia pequeña y después se puede variar.

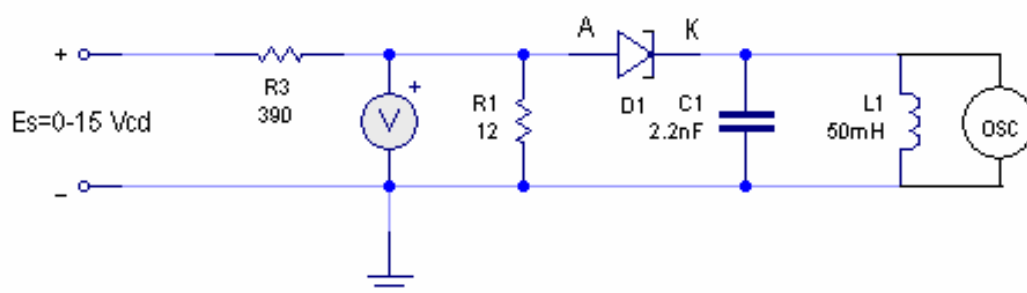


Figura 10. Aplicación del diodo como oscilador.

Se ajusta la tensión de la fuente para obtener en el display del voltímetro 150mV en R_1 (12ohms) y se observa la forma de onda resultante que se ve en el osciloscopio.

En este caso el oscilador resultante fue uno a 1.18MHz. al variar la inductancia ésta cambia en orden inversamente proporcional.

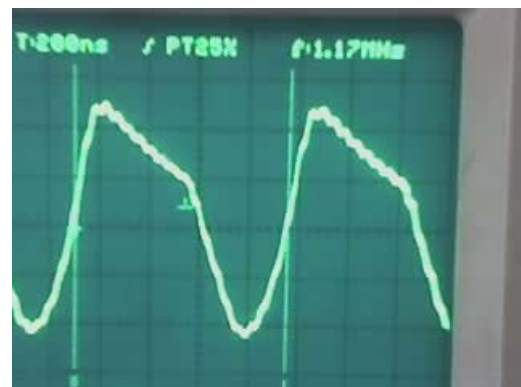


Figura 11. Señal vista en el osciloscopio del oscilador.

Información extra

(Diodo túnel usado)

Part Number = 40571

Manufacturer Name = RCA Solid State

Description = General Purpose Tunnel
Diode

I_{peak} Max. (A) = 5.0m

Peak Current Tol. (A) = .25m

V(P) (V) Peak-Point Voltage = 50m

V(V) (V) Valley-Point Voltage = 305m

Ratio I_{peak}/I_{valley} Min. = 8.0

I(F) Max. (A) Forward Current = 10m

r(s) Max. (Ohms) Series Res. = 3.0

C(t) Max.(F) Total Capacitance = 8.0p

P(D) Max.(W) Power Dissipation = 5.0m

Semiconductor Material = Germanium

Status = Discontinued

Package = Axial-A

Military = N

Referencias

Electrónica, teoría de circuitos y dispositivos, Boylestad- Nashelsky, Pearson Prentice Hall.

Prácticas con transistores.

<http://www.webelectronica.com.ar>

Miguel Ángel San Pablo Juárez

Dudas: sanpablojuarez@yahoo.com.mx

Sistema Mínimo con un 80C31

Objetivo

Diseñar un sistema mínimo con el microprocesador 80C31, tal que al ingresar números con un teclado matricial hexadecimal sean vistos en displays de 7 segmentos.

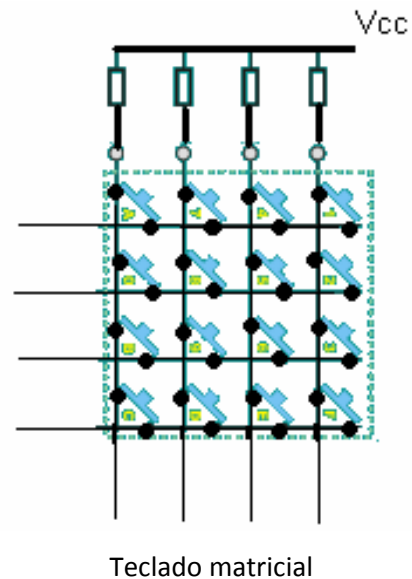
Material

- 1 microprocesador 80C31
- 1 memoria eeprom AT28C64
- 1 latch 74LS373
- 1 teclado matricial de 4x4, hecho con push button
- 1 PIA 82C55
- 1 cristal de cuarzo a 4MHz
- 2 capacitores de 22pF
- 1 capacitor de 10uF
- 1 resistencia de 220ohm
- 1 resistencia de 8.2kohm
- 2 displays de 7 segmentos ánodo común
- 7 transistores 2N2222
- 2 transistores 2N2907
- 7 resistencias de 2.2kohm
- 2 resistencias de 680ohm
- 7 resistencias de 39ohm
- 1 push button (para el circuito de reset)

Varios: cable para protoboard, protoboards, fuente 5v, caimanes, etc.

Desarrollo

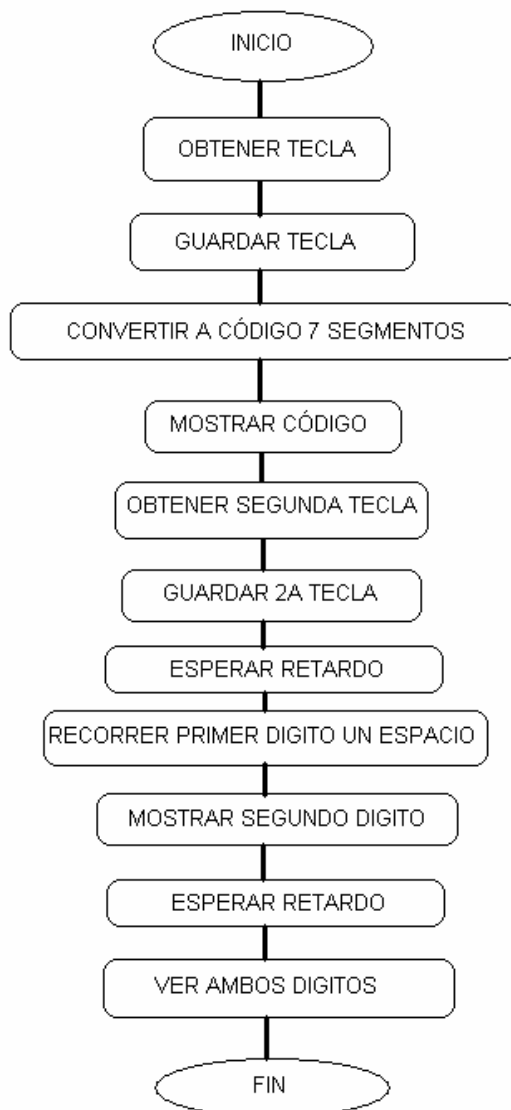
Se tiene el siguiente diagrama para hacer el teclado de 16 push button y resistencias de 10 kohm, en este caso lo fabrica el autor de este reporte, pero se puede comprar ya hecho.



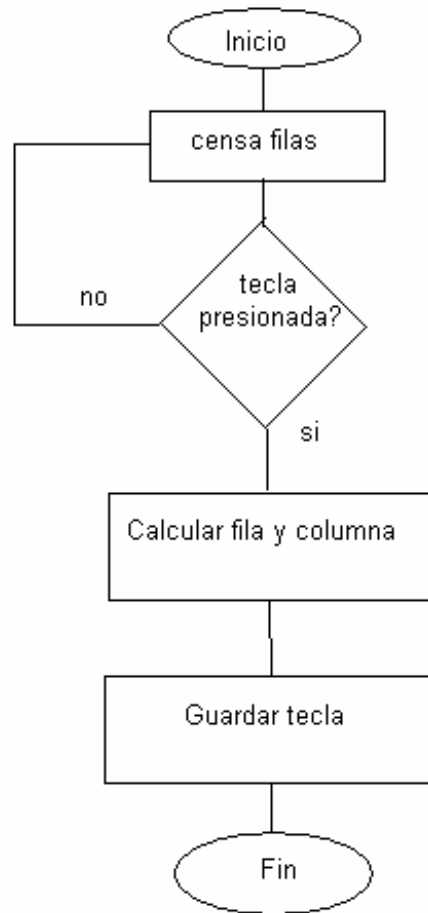
Se hace un barrido de un cero "0" lógico en cada una de las filas del teclado para sensar cuál tecla es la que está presionada, esto es, se pone a cero la primera fila y a uno las demás, y si se oprime un botón en esa fila el voltaje caerá en una de las columnas, es

decir en la columna que se oprimió, y así se obtiene el código de bits que dan tanto filas como columnas, después el cero se recorre una fila y las demás se ponen a uno, y se hace el censado; esto se hace sucesivamente y el método que se utiliza aquí es el barrido con el puerto 1 del 80C31.

El diagrama de flujo del censado es el siguiente:



Inicialmente se debe de hacer el barrido en el puerto 1 para poder obtener y guardar una tecla presionada. Esto se hace recorriendo un cero en cada línea, el diagrama de flujo de esta subrutina es el siguiente:



Posteriormente se convierte a código 7 segmentos y se muestra en uno de los displays.

Después se obtiene la segunda tecla y se espera un retardo para mostrar el segundo dígito.

Se hace la misma subrutina de obtención de tecla y se mostrará ahora sí dos dígitos en dos displays.

El código en ensamblador del programa hecho es el que a continuación se muestra:

```

;Miguel Angel San Pablo Juárez FCE BUAP.
;PARA LEER TECLADO
;Blasted Robotics::::::::::::::::::::::::::
;PARA LEER TECLADO
ORG 00H ;COMIENZA EN LA DIRECCION 00H
PRINCIPI: MOV A,#80H
MOV DPTR,#03H
MOVX @DPTR,A
MOV A,#3FH
MOV DPTR,#00H
MOVX @DPTR,A
MOV DPTR,#01H
MOVX @DPTR,A
PRINCIPL: CALL ENT_HEX ;LLAMA A LA SUBROUTINA ENT_HEX
CALL CODIG ;LLAMA LA CONVERSION A CODIGO 7 SEG
MOV A,R0
MOV DPTR,#0000H
MOVX @DPTR,A
MOV R1,A
PRINCIPI2: CALL ENT_HEX
CALL CODIG
MOV A,R0
MOV DPTR,#0000H
MOVX @DPTR,A
MOV A,R1
MOV DPTR,#0001H
MOVX @DPTR,A
SJMP PRINCIPL ;SALTA A PRINCIPL

ENT_HEX: MOV R3,#50 ;MUVE EL VALOR 50 AL REGISTRO 3 (CUENTA
DE REBOTE)
REGRESO: CALL OBT_TEC ;TECLA PRESIONADA? (C=1=SI C=0=N0)
JNC ENT_HEX ;SALTA A ENT_HEX SI C=0,(SI NO SE PRESIONA
TECLA CHECA DE NUEVO )
DJNZ R3,REGRESO ;DECREMENTA R3 Y BRINCA SI NO ES CERO A
REGRESO,50 VECES
PUSH ACC ;GUARDA EL CODIGO HEX ??? MUEVE ACC A
@SP
REGRESO2: MOV R3,#50 ;ESPERA A QUE LA TECLA SE SUELTE, MUEVE
EL VALOR 50 A R3
REGRESO3: CALL OBT_TEC ;LLAMA A LA SUBROUTINA OBT_TEC PARA
SABER SI SE SOLTO YA LA TECLA
JC REGRESO2 ;SALTA A REGRESO2 SI C=1,CHECA DE NUEVO SI LA
TECLA ESTÁ PRESIONADA

```

```

        DJNZ      R3,REGRESO3;DECREMENTA Y BRINCA SI R3 NO ES CERO A
REGRESO3,50 VECES
        POP ACC           ;RECUPERAEL CODIGO HEX, MUEVE @SP AL ACC
        RET              ;RETORNA A LA RUTINA PRINCIPAL

```

```

CODIG:   NOP                ;CONVIERTE CODIGO HEX A 7 SEGMENT
CERO:    CJNE  A,#00H,UNO
         MOV   R0,#3FH
         JMP   FIN
UNO:     CJNE  A,#01H,DOS
         MOV   R0,#06H
         JMP   FIN
DOS:     CJNE  A,#02H,TRES
         MOV   R0,#5BH
         JMP   FIN
TRES:    CJNE  A,#03H,CUATRO
         MOV   R0,#4FH
         JMP   FIN
CUATRO:  CJNE  A,#04H,CINCO
         MOV   R0,#66H
         JMP   FIN
CINCO:   CJNE  A,#05H,SEIS
         MOV   R0,#6DH
         JMP   FIN
SEIS:    CJNE  A,#06H,SIETE
         MOV   R0,#7DH
         JMP   FIN
SIETE:   CJNE  A,#07H,OCHO
         MOV   R0,#07H
         JMP   FIN
OCHO:    CJNE  A,#08H,NUEVE
         MOV   R0,#7FH
         JMP   FIN
NUEVE:   CJNE  A,#09H,DIEZ
         MOV   R0,#6FH
         JMP   FIN
DIEZ:    CJNE  A,#0AH,ONCE
         MOV   R0,#77H
         JMP   FIN
ONCE:    CJNE  A,#0BH,DOCE
         MOV   R0,#7CH
         JMP   FIN
DOCE:    CJNE  A,#0CH,TRECE
         MOV   R0,#39H
         JMP   FIN
TRECE:   CJNE  A,#0DH,CATORCE
         MOV   R0,#5EH
         JMP   FIN
CATORCE: CJNE  A,#0EH,QUINCE
         MOV   R0,#79H
         JMP   FIN
QUINCE:  MOV   R0,#71H

```

```

FIN:      NOP
          RET

OBT_TEC:  MOV    A,#0FEH                ;MUEVE FE=1111 1110 AL
ACUMULADOR(COMIENZA CON LA COLUMNA 0)
          MOV    R6,#4                  ;MUEVE EL VALOR 4 AL REGISTRO R6 (R6
COMO CONTADOR)
PRUEBA:   MOV    P1,A                  ;MUEVE EL ACUMULADOR AL PUERTO P1
(ACTIVAR LINEA DE COLUMNA)
          MOV    R7,A                  ;MUEVE EL ACUMULADOR AL REGISSTRO R7
(GUARDA ACC)
          MOV    A,P1                  ;MUEVE LO QUE HAY EN EL PUERTO 1 AL
ACUMULADOR (LEE DE NUEVO EL PUERTO 1)
          ANL    A,#0F0H                ;A="A AND F0"="ACUMULADOR AND
11110000" (AISLA LINEAS DE FILAS)
          CJNE   A,#0F0H,TECL          ;COMPARA Y BRINCA SI A ES DIFERENTE
DE F0 (LINEA DE RENGLON ACTIVA?)
          MOV    A,R7                  ;MUEVE R7 AL ACUMULADOR (NO: VAMOS A
LA SIGUIENTE)
          RL     A                      ;ROTA EL ACC A LA IZQUIERDA 1 BIT
(LINEA DE COLUMNA)
          DJNZ   R6,PRUEBA             ;DECREMENTA Y BRINCA SI R6 ES
DIFERENTE DE CERO
          CLR    C                      ;PONE A C=0 (NO HAY TECLA PRESIONADA)
          SJMP   SALIR                 ;SALTA A SALIR (RETORNA CON C=0)
TECL:    MOV    R7,A                  ;MUEVE ACUMULADOR A R7, GUARDA A
EN R7
          MOV    A,#4                  ;MUEVE EL VALOR 4 AL ACUMULADOR
          CLR    C                      ;PONE A C=0, PARA PESAR COLUMNA
          SUBB   A,R6                  ;A=A-R6-C,4-R6=PESO
          MOV    R6,A                  ;LO GUARDA EN R6
          MOV    A,R7                  ;MUEVE R7 AL A, RECUPERA CODIGO
ESCANEADO
          SWAP   A                      ;CANJEA A,LO PONE EN EL NIBLE MAS
BAJO,
          MOV    R5,#4                  ;MUVE EL VALOR 4 A R5 PARA USARLO
COMO CONTADOR
OTRVZ:   RRC    A                      ;ROTAR HASTA CERO,ROTA A LA
DERECHA A TRAVES DEL CARRY
          JNC    FINAL                 ;SALTA A FINAL SI C=0
          INC    R6                    ;R6=R6+1
          INC    R6                    ;R6=R6+1
          INC    R6                    ;R6=R6+1
          INC    R6                    ;R6=R6+1
          DJNZ   R5,OTRVZ              ;DECREMENTA Y BRINCA SI R5 NO ES
CERO:SE SUMAN 4 HASTA ACTIVAR LINEA ENCONTRADA
FINAL:   SETB   C                      ;PONE A C=1(TECLA PRESIONADA)
          MOV    A,R6                  ;MUEVE R6 AL ACUMULADOR, CODIGO EN A
SALIR:   RET                          ;RETORNAR A PRINCIPAL

          END                          ;FIN

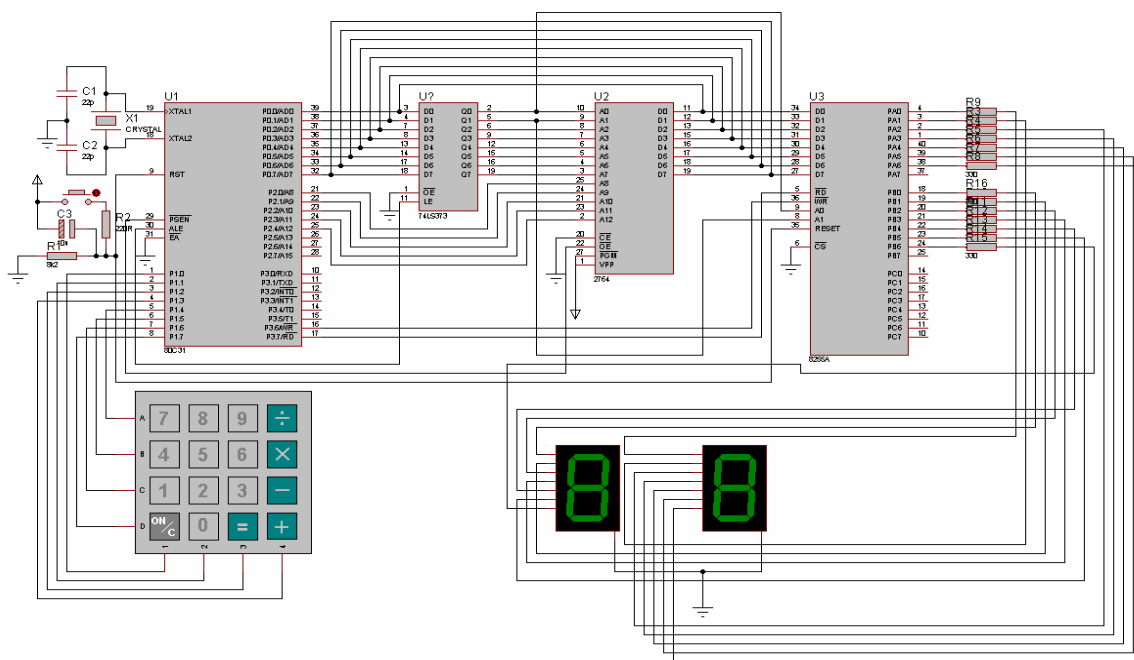
```

Como se puede observar el código está provisto para obtener una tecla y mostrarla en el puerto 3, hasta el momento es el desarrollo de la práctica que funciona.

El código fue compilado en el software MIDE51 Pack, pero puede ser compilado en

cualquier otro como por ejemplo los AtmelCompilers y es grabado en Chipmaster sobre una EEPROM AT28HC64, y conectada como lo muestra el diagrama.

El diagrama final armado es el siguiente:



Sistema mínimo

Note que el PPI 82C55 es conectado para expandir el puerto para cada display, como si se tratara de una RAM, y es necesario el latch en la parte de la salida de direcciones y datos para multiplexar entre datos y direcciones.

Si sólo usáramos un puerto no sería necesario el 82C55, pero el circuito de reset es básicamente útil para inicializar el micro, una vez conectado todo hay que pulsar el reset para comenzar a trabajar con el sistema mínimo.

Note que también es posible mapear memorias ROM en caso de no conseguir esta, sólo necesitaríamos grabar el código .hex generado por bloquitos en cada memoria pequeña desde el software grabador.

Este sistema mínimo es ideal para comenzar a desarrollar sistemas más complejos y grandes que los microcontroladores con memoria interna grabable no efectuarían tan fácil ni tan rápido. 😊

Referencias:

-Apuntes de la materia Arquitectura de computadoras, verano 2005, R.Alvárez. FCE. BUAP

-Los Microprocesadores Intel, Barry. B. Brey. Ed. Pearson

-The 8051 Microcontroller. I. Scott. MacKenzie. 3a. Ed. Paperback

R. Doradus

sanpablojuarez@yahoo.com.mx

Función Sleep con un 18F452 en PICC.

Este ejemplo muestra como utilizar la función "sleep" de un PIC. Cuando un botón es presionado el procesador se pone en modo sleep. Cuando el botón es soltado, el procesador despierta y continúa con su rutina normal, en este caso continúa contando, ya que para este caso se programa un contador simple.

Este ejemplo está basado en el demo proporcionado por el software PICC de CCS

llamado EX_WAKUP.C contenido en la carpeta /examples/ del mismo software.

Sin más, comencemos creando un nuevo source file ó archivo de código fuente, entonces incluimos los clásicos parámetros iniciales del código fuente, como en esta ocasión usaremos un PIC18F452 escribimos lo siguiente:

```
#include <18F452.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=2000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
```

Donde elegimos un oscilador de 20MHz y configuramos los bits para la interfaz RS-232. Declaramos un bit para que nos sirva como bandera de indicador par el sleep.

```
short modo_sleep;
```

Vamos a utilizar la interrupción externa tomando el pin B0 del PIC, para ello declaramos la interrupción como sigue:

```
#INT_EXT
```

Con su respectiva función de censado de interrupción:

```
void ext_isr() {
static short boton_presionado=FALSE;

    if(!boton_presionado)
    {
        boton_presionado=TRUE;
        modo_sleep=TRUE;
        printf("El Procesador esta en modo sleep.\r\n");
        ext_int_edge(L_TO_H);
    }
    else
    {
        boton_presionado=FALSE;
        modo_sleep=FALSE;
        ext_int_edge(H_TO_L);
    }
    if(!input(PIN_B0))
        boton_presionado=TRUE;
        delay_ms(100);
}
```

En esta rutina declaramos el flag que indica si el botón ha sido presionado, para después tomar las decisiones de salida del PIC, si detectamos que en el pin B0 hay un cero lógico relacionamos este valor con el flag de modo_sleep por medio de los otros if, así, si oprimimos el botón y lo soltamos tomaremos sólo el valor de censado del flanco, es decir, lee solamente una pulsada

```
void main()
{
    long contador;

    modo_sleep=FALSE;

    ext_int_edge(H_TO_L);
    enable_interrupts(INT_EXT);
    enable_interrupts(GLOBAL);

    printf("\n\n");

    contador=0;
    while(TRUE)
    {
        if(modo_sleep)
            sleep();
        printf("El valor del contador es: %5ld  \r\n",contador);
        output_d(contador);
        contador++;
        delay_ms(1000);
    }
}
```

En esta rutina primero declaramos la variable contador, luego indicamos la inicialización de la bandera de modo con el valor de cero, luego inicializamos la interrupción para el censado del botón de acción, y activamos las interrupciones, limpiamos la pantalla para después declarar el valor inicial de la variable contador, así, podemos ahora ciclar el programa con un while, entonces repetiremos las siguientes instrucciones:

a la vez, esto indica que cuando la lectura sea 1 ó 0 respectivamente del pin B0, asignaremos su respectivo valor a la variable botón_presionado, así pues soltaremos a la función principal y haremos el conteo o no dependiendo de el pin de interrupción.

Para la parte de la función principal:

Primero, si el modo sleep es activo, es decir, si el botón que en este caso es activo bajo está presionado, mandamos a dormir el PIC, recordar que si sólo tenemos una línea de instrucción después de un if no se colocan llaves y la instrucción para el caso contrario se toma desde la segunda línea después de el mismo if. Entonces si no detecta botón presionado se imprime y se muestra el valor de nuestro contador en el puerto D, hacemos con el clásico ++ que se incremente la variable y para visualizarlo esperamos 1 segundo.

El diagrama de la simulación se muestra en la figura 1.

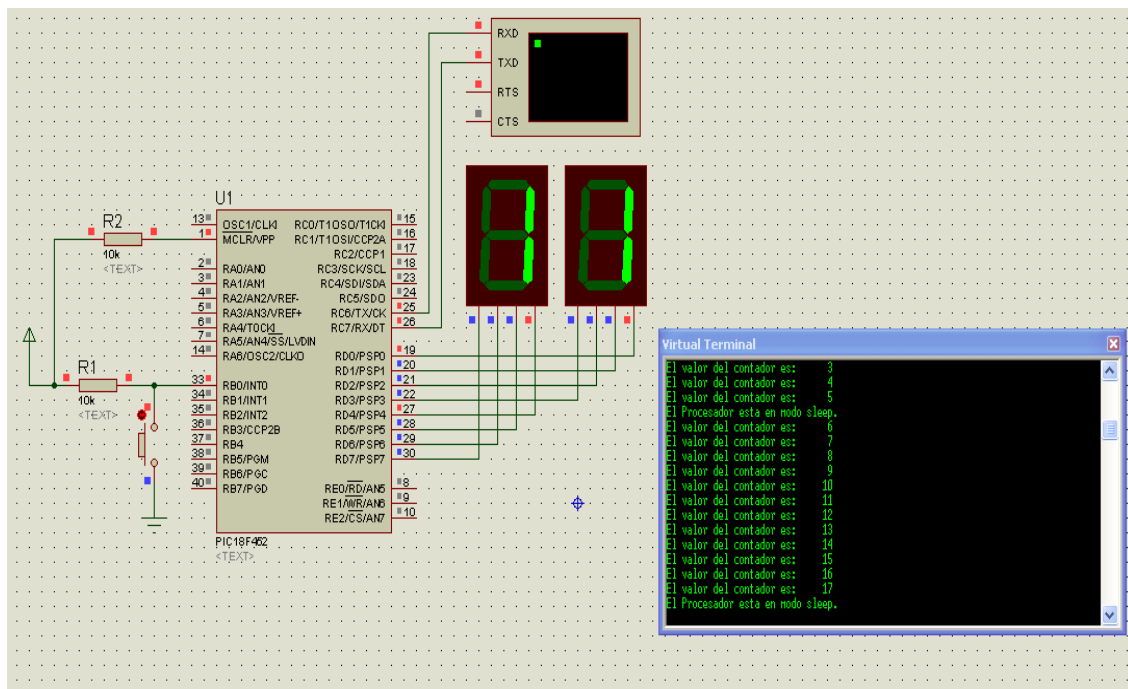


Figura 1. Diagrama de la simulación.

El programa compilado se muestra a continuación.

```

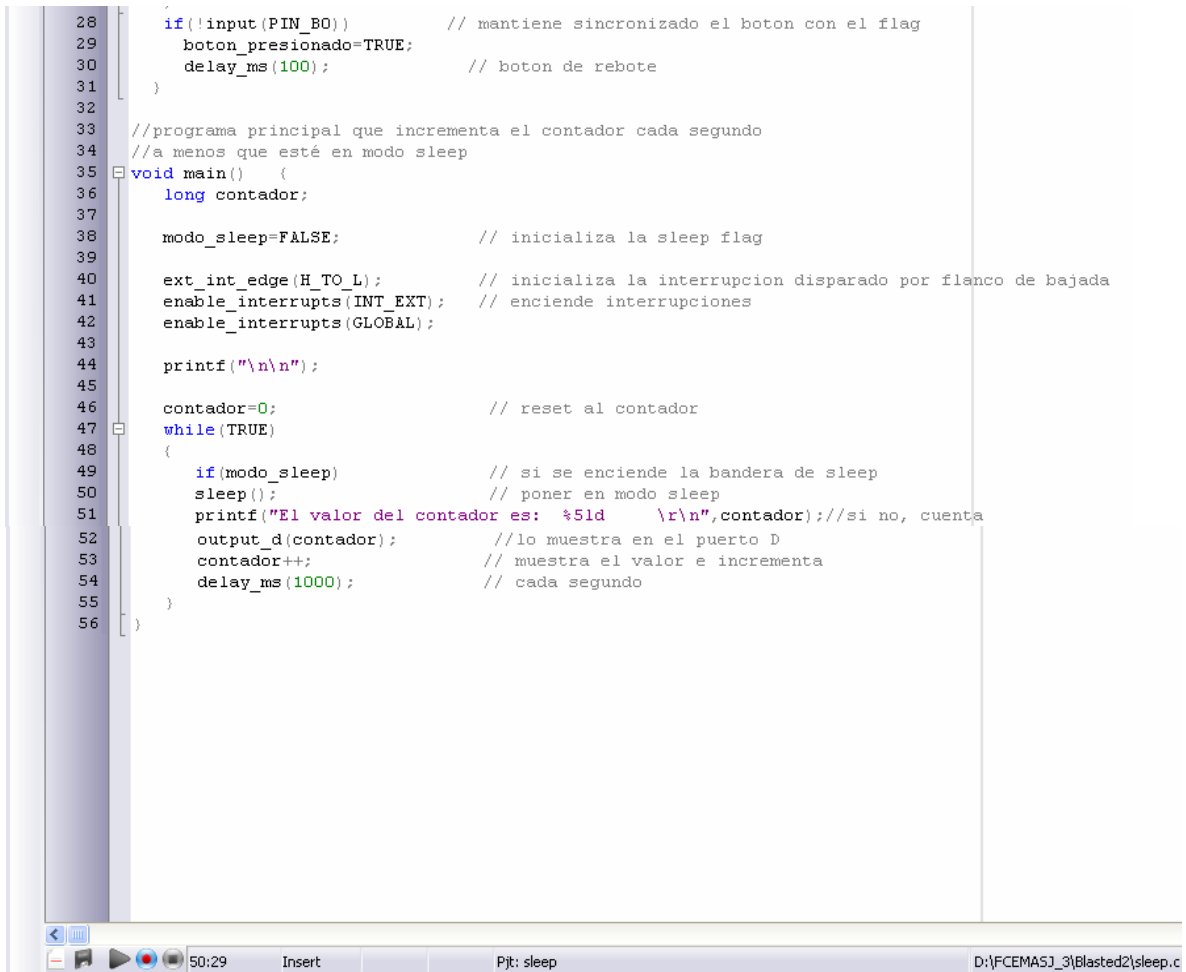
Project Edit Search Options Compile View Tools Debug Document User Toolbar
Compile Build Build All Clean Lookup Part Program Chip Debug C/ASM List Symbol Map
sleep.c
1 #include <18F452.h>
2 #fuses HS, NOWDT, NOPROTECT, NOLVP
3 #use delay (clock=20000000)
4 #use rs232 (baud=9600, xmit=PIN_C6, rcv=PIN_C7)
5
6 //bandera global para enviar al procesador al modo sleep
7 short modo_sleep; //sleep_mode;
8
9 //interrupción externa cuando el boton sea presionado o soltado
10 #INT_EXT
11
12 void ext_isr() {
13 static short boton_presionado=FALSE;
14
15 if (!boton_presionado) // Si el botón fue presionado
16 {
17 boton_presionado=TRUE;
18 modo_sleep=TRUE; // activa sleep
19 printf("El Procesador esta en modo sleep.\r\n");
20 ext_int_edge(L_TO_H); // cambio para las interrupciones
21 }
22 else // si el boton no fue presionado
23 {
24 boton_presionado=FALSE;
25 modo_sleep=FALSE; // resetea la bandera de sleep
26 ext_int_edge(H_TO_L); // cambio a las interrupciones al presionar
27 }

```

```

28     if(!input(PIN_B0)) // mantiene sincronizado el boton con el flag
29         boton_presionado=TRUE;
30         delay_ms(100); // boton de rebote
31     }
32
33     //programa principal que incrementa el contador cada segundo
34     //a menos que esté en modo sleep
35     void main() {
36         long contador;
37
38         modo_sleep=FALSE; // inicializa la sleep flag
39
40         ext_int_edge(H_TO_L); // inicializa la interrupcion disparado por flanco de bajada
41         enable_interrupts(INT_EXT); // enciende interrupciones
42         enable_interrupts(GLOBAL);
43
44         printf("\n\n");
45
46         contador=0; // reset al contador
47         while(TRUE)
48         {
49             if(modo_sleep) // si se enciende la bandera de sleep
50                 sleep(); // poner en modo sleep
51             printf("El valor del contador es: %5ld \r\n",contador);//si no, cuenta
52             output_d(contador); //lo muestra en el puerto D
53             contador++; // muestra el valor e incrementa
54             delay_ms(1000); // cada segundo
55         }
56     }

```



Figuras 2 y 3. Programa compilado en PICC.

Referencias

PICC help EX_WAKEUP.C

R. Doradus

Dudas: sanpablojuarez@yahoo.com.mx

Artículo de Tecnología. Fuera Teclas. Y usted, ¿Qué opina?

Todos conocemos los ya famosísimos iPods, además de que usan la clásica ruedita giratoria que es la que permite navegar entre sus menús, ahora imaginen que el teclado de una portátil (Laptop) sea sustituido por esta pequeña ruedita,

Ya sé, suena como una locura; tal vez ya hayan escuchado sobre la nueva MacBook Wheel, pues actualmente se ha presentado en la Anual-Mac-Expo esta nueva Mac, con la cual se pretende que el estar tecleando, para redactar o escribir, quede en el pasado; o simplemente el uso de un teclado para transitar a través de la computadora, se vuelva lo mas arcaico del planeta. Y eso no es todo, también se ha simplificado la organización de los archivos guardados en el equipo, simplemente oprimiendo la letra inicial del nombre del archivo 'voilà' se genera una lista de posibles nombres de archivos guardados y simplemente el usuario tiene que buscar entre esa lista el archivo.

El precio varia de 2,500 UDS hasta los 9,960 USD. Otra de las características de este revolucionario equipo es que virtualmente irrompible y su Hummingbird battery (lo que se traduce como 'batería de colibrí'),

brinda 20 minutos de carga antes de que necesite ser cargada la batería

De manera personal pienso que el sustituir un teclado por una simple ruedita no sea del todo cómoda, como sabemos el uso de nuestros 10 dedos a la hora de escribir (lo digo por los que saben mecanografía), se facilita mucho la redacción dentro del ambiente de una computadora, ahora también los que no sabemos mecanografía solo usamos cuando menos 4 dedos a la hora de escribir o redactar un texto. Ahora imaginen tener que estar girando la ruedita para escribir el texto; si a veces en el mismo iPod, por querer atinarle a la canción, nos pasamos y entonces hay que avanzar o regresar para poder llegar al punto deseado, así que les dejo a su consideración este nuevo producto.

Bueno si esto no quedo del todo claro pueden verlo en la siguiente página:

<http://www.youtube.com/watch?v=9BnLbv6QYcA>

Autor: Apolos_mr

Trivia

Hola lectores, en esta ocasión van unas preguntillas para aquellos que van hacia el área de comunicaciones, son unas preguntas de cajón, haber si te las sabes!

1. ¿Qué sistema de codificación y transmisión de Televisión a color analógica se utiliza en México?
2. ¿Qué ancho de banda utiliza un canal de televisión transmitido en ese sistema para contener la señal de vídeo, la señal de audio y unas bandas de resguardo?
3. ¿Por qué en algunos países se utiliza el sistema PAL en vez del que se usa aquí en México?
4. ¿Por qué se dice que vemos las imágenes de televisión a 30 cuadros por segundo?
5. ¿Qué es lo que en realidad sucede en la pantalla de un TV y que no somos capaces de distinguir por la rapidez en que sucede?
6. Un clásico de todos los tiempos: ¿Qué es SEÑAL?

¡¡¡ Ya saben, si la contestan correctamente obtendrán premios chidos !!!

Manda tus respuestas a:

blastedrobotics@yahoo.com.mx