

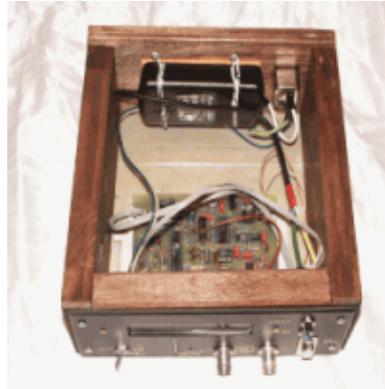
Contador de Frecuencia de 1Hz–100Mhz con visor LCD y RS232 interfase



by Guido Socher ([homepage](#))

About the author:

Guido ama a Linux no solo porque divertido descubrir las grandes posibilidades de estos sistemas, sino también por la gente involucrada en su diseño.



Abstract:

Este artículo continuamos con la serie de Microcontroladores AT90S4433. Les sugiero a ustedes que lean artículos anteriores sobre los microcontroladores Atmel programando con cuidado: 1. How to install and use the Linux AVR development environment and how to build the programmer hardware: March 2002, Programming the AVR Microcontroller with GCC 2. How to make your own printed circuit board: May 2002, A LCD control panel for your Linux server Esta vez diseñamos un contador de frecuencia el cual puede medir frecuencias de 1Hz a 100MHz. Como alternativa usted puede usarlo para contar eventos, tales como cuantas veces una persona cruza la calle (Lo que sea disponible en pulsos digitales). El contador tiene un visor LCD también como una interfase RS232 para leer los resultados del contador desde Linux.

Introducción

La frecuencia de una onda senoidal o cuadrada es expresada como número de oscilaciones por segundo. En orden de determinar la frecuencia de una señal continua una sola cosa es necesaria para contar esas oscilaciones. Este camino nos determina la frecuencia de la primera armónica de la señal continua oscilante. Para medir frecuencias que no son continuas "sonido" compuestas, necesitaremos un analizador de espectro. Este es si embargo una pieza diferente de hardware. Lo que hemos diseñado aquí es un contador de frecuencia para señales continuas oscilantes. Asumimos que la señal no cambia su frecuencia durante un intervalo dado donde muestreamos la señal.

Nuestro contador de frecuencia trabaja en 2 pasos:

1. Saque cualquier compensación de la C.C. de la señal y después conviértala en una onda cuadrada usando un comparador del voltaje.
2. Cuente el número de pulsos por vez y divída por intervalo de tiempo.

Qué usted necesita

Para construir el contador de la frecuencia usted necesita las siguientes piezas:

1 x Atmel At90S4433 Microcontrolador

1 x zócalo para CI 28pinos 7,25 mm

2 x zócalo para CI 16pinos

1 x zócalo para CI 1pin

1 x zócalo para CI 14pinos

No hay zócalo para CI en el MAX903 y el 74F74. Soldarlos directamente a la placa!

1 x MAX232

1 x Cristal 4,194304MHz

1 x LED (Verde)

1 x BC557 transistor PNP

4 x 1uF capacitor (polarizado)

2 x 27pF capacitor cerámico

4 x 10nF mini capacitor cerámico

3 x 100nF mini capacitor cerámico

1 x 200nF mini capacitor cerámico

1 x 0,47uF mini capacitor cerámico

2 x resistor 470 Ohm

1 x resistor 470K

2 x resistor 100 Ohm

3 x resistor 1k

5 x resistor 10k

3 x resistor 47K

1 x resistor 220 Ohm

3 x resistor 4K7

1 x resistor 3k3

1 x resistor 2k2

1 x resistor 47 Ohm

1 x 4K7 potenciómetro (tan pequeño como sea posible), si usted desea puede sustituir este potenciómetro por un par de resistores. Este potenciómetro se utiliza para fijar el contraste del LCD. Para mis visores LCD 100Ohm juntas con 1K da un buen contraste.

1 x diodo Zener 4.3V

2 x pequeños botones de contacto

1 x 470uF capacitor (polarizado)

1 x 4,7uF capacitor (polarizado)

1 x diodo 1N4001

1 x 74HC02 TTL IC

1 x 74F74 fast TTL IC

1 x 74HC390 TTL IC

1 x LM393 comparador de tensión

1 x MAX903 comparado de tensión de alta velocidad, 8 Pinos plásticos empaquetado DIP, usted puede pedirlo directamente de www.maxim-ic.com en caso de que su distribuidor local no lo tenga en la stock.

1 x 7805 regulador de 5V

1 x 2 líneas 20 caracteres visor LCD con compatibilidad a la interfase HD44780. Con o sin fondo luminoso (16 pines o 14 pines)

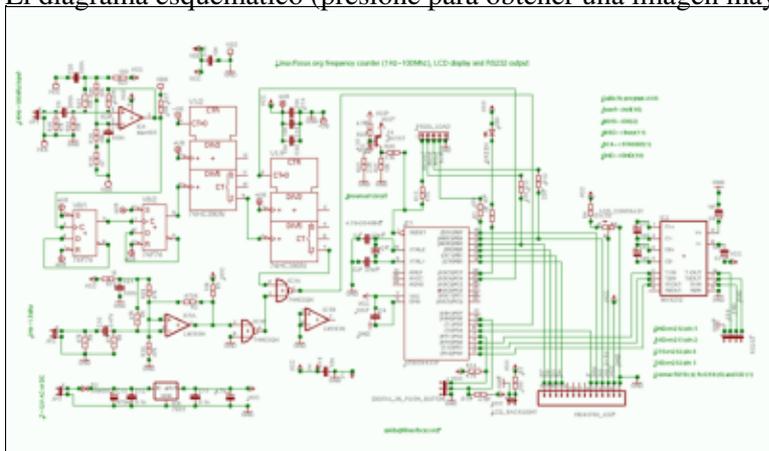
Todas los visores LCD que he visto siempre con 14 o 16 pines en el conector eran compatibles con HD44780. Usted puede usar un visor de 3 o 4 líneas pero deberá modificar el software un poquito.

Además de ésta usted necesita algunos cables, conectores (BCD, alimentación, RS232) y un transformador de 9V o alguna otra fuente de alimentación AC o DC(150mA). Usted consigue a veces las fuentes de alimentación muy baratas que entran directamente en el zócalo de pared y se utilizan para toda la clase de aparatos de electrónica.

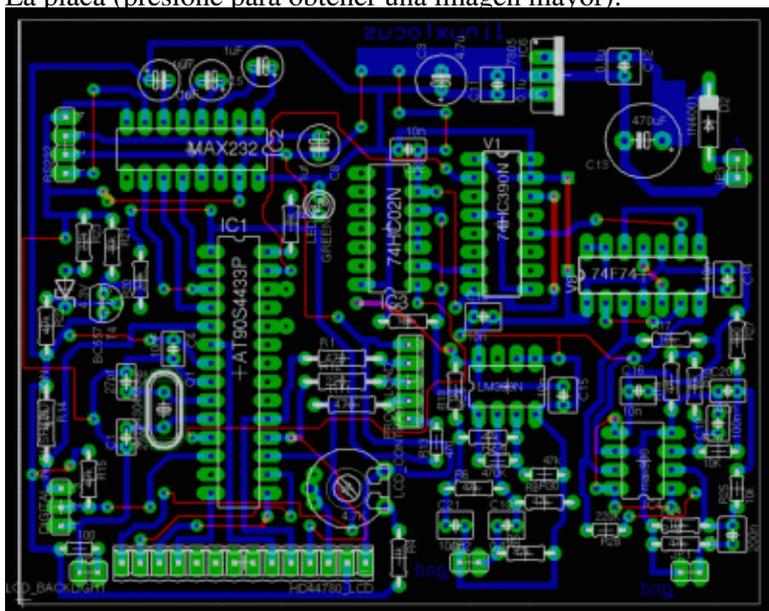
Diagrama esquemático y placa

Utilicé eagle para Linux para diseñar el diagrama esquemático y la placa. El programa tiene algunos problemas para entender que todos los contactos diferentes de la fuente de alimentación en los CI son 5V. Usted por lo tanto recibirá un error si activa la comprobación de reglas eléctricas. Sin embargo el diseño es correcto.

El diagrama esquemático (presione para obtener una imagen mayor):



La placa (presione para obtener una imagen mayor):



La placa es específicamente diseñada para pasatiempos electrónicos. Solo la capa azul da a entender que es para ser grabada al agua como una placa de circuito impreso. Las líneas rojas son cables. Es mucho más fácil y menos preciso se requiere para construir una sola placa de circuito impreso de solo lado. Usted puede poner los cables (rojos) tales que tenga una corta longitud. I could not do that in eagle.

el diseño de la placa con fondo blanco para mejor impresión: [placa con fondo blanco](#) (Nota: Éste no es el archivo que usted necesita hacer la placa de circuito impreso.)

Los archivos del Eagle están todos incluidos junto con el programa (linuxfreqcount-0.4) el cual usted podrá bajar al final del artículo.

Cómo trabaja

El microcontrolador AT90S4433 tiene dos contadores internos. Uno es de 16bit de ancho y uno de 8bit de ancho. utilizamos el contador de 8bit de ancho para generar base de tiempo precisa de la frecuencia de reloj del cristal del AT90S4433. Para este propósito usamos un cristal de 4194304Hz y da pulsos al contador de 8bit mediante un pre-escalador interno de 1/256 (véase la hoja de datos del AT90S4433, descárguela al final de este artículo). El contador de 8bit esta configurado para generar una interrupción ante un derrame. En otras palabras obtenemos $4194304\text{Hz} / (256 * 256) = 64\text{Hz}$ de base de tiempo. Por usar un lazo variable generamos de estas llamadas de funciones en intervalos de 1Hz o 64Hz.

Ahora tenemos una función la cual es llamada a intervalos de 1Hz o 64Hz dependiendo del modo que nuestro programa contador esta corriendo. Todo lo que necesitamos hacer ahora es leer el contador de 16bit de ancho desde esta función y mostrar el resultado. El contador de 16bit de ancho (contacto PD5 en el Microcontrolador) obtiene sus pulsos de señal de la señal que queremos medir.

El microcontrolador muestrea las señales de entrada para sincronizarlo con su reloj interno. Según el teorema de muestreo podemos por lo tanto medir señales máximas arriba de la mitad de la frecuencia del cristal. Ese es el límite teórico. En la practica podemos medir señales arriba de 1,5MHz con el Microcontrolador.

Para medir frecuencia mas altas necesitamos un pre-escalador/divisor. Eso es lo que nos proporcionan los CI 74F74 y 74HC390. El 74F74 es usado como 1/4 divisor rápido asíncronico y el 74HC390 divisor de 1/25. No podemos utilizar directamente el 74HC390 como un divisor 1/100 porque puede manejar un máximo de 25MHz

El circuito prevé 2 entradas. Una por el divisor 1/100 y uno directo a la entrada. Dependiente en la frecuencia de la señal que usted desea medir use uno de esas líneas de entrada (no ambas).

Si usted está solamente interesado en frecuencias hasta 1.5Mhz (Ej. para establecer medidas para la respuesta de audio de un amplificador de audio) usted puede también construir una versión simplificada del contador y dejar afuera el MAX903, 74F74 y el 74hc390. Usted puede utilizar aun el mismo programa software y placa de circuito.

El propósito de los 2 comparadores de tensión (MAX903, alta velocidad y LM393) es amplificar las señales y generar señales cuadradas por ejemplo ondas senoidales.

El cuidado especial fue tomado durante el diseño y la disposición de la placa para evitar oscilaciones de los comparadores cerca del umbral. Observe usted que el MAX903 es realmente un comparador muy de alta velocidad. Si usted utiliza una señal de entrada ruidosa, imagine por ejemplo una señal modulada de 100KHz con 1MHz de ruido, entonces usted contara algunos pulso adicionales de 1MHz cuando la señal de 100KHz cruce el nivel de los cero voltios.

El resistor de 220K en el MAX903 evita la influencia del ruido a un cierto grado proporcionando una

regeneración positiva. Usted debe sin embargo no alimentar el contador con señales ruidosas.

El microcontrolador puede ser controlado vía RS232 y vía 2 botones de presión (botón de limpieza y botón de cambio del modo de conteo).

Los 2 canales de entrada son combinados vía una compuerta O (74hc02). La compuerta O por supuesto suprimirá señales de entrada si desconecta la entrada 1/100 mientras tengamos un "1" lógico en las líneas de salida del 74hc390. Por lo tanto el 74hc390 es desbloqueado (contacto 2) cuando usted cambia el modo de conteo o cuando presiona el botón de limpieza.

El programa

El programa para el Microcontrolador establece 2 contadores en modo interrupción. Como esto es hecho es descrito muy bien en la hoja de datos del AT90S4433 (véase referencia). Usted tiene que fijar un número de registros. Es importante pero preferentemente seco. Por lo tanto no lo repetiré aquí. La mayoría de la lógica del contador es implementada en el archivo linuxfreqcount.c. El resto de los archivos son "bibliotecas" para el LCD, el UART, el etc.... Cuando escriba el programa para el Microcontrolador usted debe tener cuidado de no usar mas de 128Bytes de RAM. Es todo lo que tenemos. Por lo tanto es mejor no haber jerarquizado llamadas de la función con muchos de parámetros. Se prefieren el uso de banderas y variables globales.

Cuando los registros del contador se configuran correctamente entonces la función SIGNAL(SIG_OVERFLOW0) será llamada en intervalos de 64Hz. Aquí leemos el contador de 16bit y establecemos la bandera (hflag) para procesar el resultado del contador por otro lado, función handlecounterresult(). Con 1Hz de frecuencia de puerta podemos usar directamente el resultado del contador pero con 1/64 seg. de abertura de puerta debemos multiplicar el resultado por 64. Las matemáticas que usted puede hacer con el Microcontrolador son limitadas (no tenemos lindos números enteros de 32bit como en las CPU Pentium). Multiplíquese afortunadamente por 64 es justo 6 veces desplazadas a la izquierda. Almacenamos el resultado final del contador en variables de 3x 8bit (counterval[3]).

Ahora tenemos el valor correcto del contador representado en 24bit. Para exhibirla, necesita ser convertida al ASCII decimal. Printf haría normalmente esto pero no tenemos printf y excedería toda nuestra memoria. Haga do en la conversión binario decimal divida él numero de 24bit por 10 con recordatorio. Los recordatorios son números decimales. Nuestra CPU no puede hacer cálculos de 24bit por lo tanto procesamos el número en trozos de 8bit (función divby10()). Agregando a cada digito el valor ASCII del cero nos provee finalmente con una representación ASCII de nuestro número (función longtoascii()).

Esta cadena ASCII puede ser enviada a la línea RS232 y al visor LCD.

Haciendo el circuito impreso

El paquete del programa contiene un archivo postscript (linuxfcoun.ps) para la placa de circuito impreso. Personalmente encuentro que los cojines son siempre un poco más chicos. Por lo tanto recomiendo fuertemente agrandarlos un poquito con un marcador antes de grabar a agua fuerte la placa. El proceso cómo hacer la placa en casa es descrito en: May 2002, A LCD control panel for your Linux server. Algunos lectores me dijeron que el spray Kontakt Pausklar-21 esta difícil de conseguir. Usted puede usar petróleo, como los usados para las lámparas de petróleo. Usted tiene que deslavar el petróleo con un poco de jabón antes de poner la placa dentro del desarrollador.

¿Cómo construir una caja para el contador de frecuencia?



Un grande problema para los pasatiempos electrónicos es usualmente la caja o receptáculo para todos los lindos circuitos que usted construyó. Cajas de aluminio pre-construidas son generalmente caras. Exactas medidas de las hojas de metal son difíciles. Yo he encontrado una barata y profesional solución viable. Yo use una madera de abeto para los costados y las esquinas, usted puede pintarlas un poco para que luzca mas caro. Use una pintura soluble, pintura para interiores. Para el frente, parte superior y la parte inferior yo use una hoja de metal y no necesitan plegarla. Solo córtenlas en las medidas exactas. Ustedes pueden ver en la figura de la derecha como los marcos de madera se parecen. En la izquierda usted ve en la ultima caja (La tapa de la parte superior esta perdida). normalmente recomiendo usar un zócalo para CI porque hace más fácil el encuentro de errores problemáticos de "tiroteo" pero para estos 2 CI no deberías usar zócalos para reducir los efectos parasitarios no deseados

Montando la placa



Cuando suelde todas las partes de la placa usted deberá prestar atención a la área de alta frecuencia (MAX903 y 74F74 con resistores y capacitores). Yo recomiendo normalmente usar zócalos para los CI porque hace más fácil el encuentro de errores problemáticos, pero para estos 2 CI no deberías usar zócalos para reducir los efectos parásitos. Para los capacitores de desacople entre tierra y la fuente de alimentación usted deberá usar pequeños capacitores cerámicos. El valor de estos capacitores puede ser cualquiera entre 10nF y 100nF.

Este es lejos el circuito más complejo que hemos diseñado en LinuxFocus. Recomiendo construirlo en etapas y comprobando mientras tanto que los diferentes componentes estén funcionando:

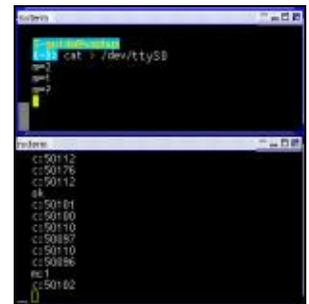
1. Soldar primero las necesarias de la fuente de alimentación (7805 etc...) en la placa y comprobar que funcione.
2. Soldar todas las partes en la placa pero solo ponga el Microcontrolador en el zócalo.

3. El paquete linuxfreqcount-0.4 incluye un programa de comprobación, avr_led_lcd_test.c, para comprobar el Microcontrolador, el LED y el LCD. Cárguelo. Una el cable del programador según lo descrito en el primer artículo del AVR (March 2002, Programming the AVR Microcontroller with GCC) y luego escriba "make testload" para cargar el programa avr_led_lcd_test del paquete linuxfreqcount-0.4. El LED debe parpadear y el LCD debe demostrar el texto "hola".
4. Ponga todos los CI en sus zócalos. Cargue el programa final (make load). Compruebe nuevamente primero la parte de baja frecuencia y después la parte de alta frecuencia. Usted necesita una cierta clase de oscilador como fuente de frecuencia (preferentemente señal cuadrada de salida).

Usando el contador

El contador puede operar de diferentes modos:

1. 1. cuenta ascendente continua
2. 2. Conteo de frecuencia con una frecuencia de disparo de 1Hz. En este modo usted puede medir frecuencia arriba del máximo del contador de 16bit, 65535Hz o 6553500Hz con el divisor 1/100.
3. 3. conteo de frecuencia con una frecuencia de disparo de 64Hz. En este modo usted puede contar frecuencias arriba de 1,5 MHz o 100MHz con el divisor 1/100.



RS232 ASCII command interface.

Usted puede cambiar el modo vía el botón de modo o vía la interfaz RS232 con comandos ASCII. La interfaz de comandos ASCII RS232 trabaja como lo descrito en el artículo [May 2002, A LCD control panel for your Linux server](#). Los comandos disponibles son descritos en [README.commands](#) file.

El LED verde indica que la compuerta esta abierta. Esto es: los pulsos pueden entrar en el contador y ser contados ahí.

El segundo botón es el botón de limpieza. Este botón puede ser utilizado para limpiar el contador (principalmente para modo conteo ascendente) y para seleccionar cambios entre multiplicar por 100 y visualización normal. El multiplicar por 100 causa que el resultado mostrado sea multiplicado por 100. De esta forma usted no tendrá que hacer complicados cálculos en su cabeza :-). Es útil cuando usted utiliza el divisor de 1/100.

¿Cuan exacto?

Cualquier contador electrónico tiene un limite exactitud de +/-1. El ultimo dígito siempre variará un poco aun si ocurriera que usted tenga una señal muy estable de entrada. Si la frecuencia de compuerta es 64Hz entonces el resultado es internamente multiplicado por 64 lo que significa que la exactitud es de +/-64. De la misma

forma el divisor 1/100 influye en la exactitud del resultado. Este contador sin embargo es muy bueno. El problema principal es el cambio de temperatura del cristal de 4194304Hz. Los cristales oscilan mas bajo cuando están calientes. Esto es físico lo podría eliminar solo controlando la temperatura exactamente.

Es posible calibrar el contador si usted encuentra una buena fuente de señal de referencia. Puede calibrarlo en el programa agregando pequeños retrasos en la función `handlecounterresult()`. Yo compare los resultados del contador contra una señal de referencia y encontré que mi contador era casi preciso. Therefore the calibrate delays are currently commented out in the source code. Puedes bien influir en la frecuencia del cristal un poco cambiando un de los capacitores de 27pF (Tendrá que experimentar con 10pF, 50pF etc..).

Para uso normal en casa no debería ser nada calibrado. Los cristales estándares de 4194304Hz con muy precisos.

El contador en operación

Esta es una foto final del contador en operación:



Para un estudio detallado

En este artículo ningún programa de aplicación es provisto para evaluar automáticamente los resultados bajo Linux. Uno podría imaginar por ejemplo usarlo para contar frecuencia para medir la velocidad del viento en áreas remotas o algún otro evento que pudiera ser traducido en frecuencias digitales. No es difícil escribir programas bajo Linux para hacer mediciones por un largo tiempo. Usted puede por ejemplo modificar los programas en perl desde el artículo [May 2002, A LCD control panel for your Linux server](#). Si usted esta solo interesado en anotar los resultados en un archivo podría usar simplemente estos comandos:

```
ttydevinit /dev/ttyS0 cat /dev/ttyS0 > your_logfile.txt
```

Esto asume que el contador esta conectado en el COM1 (=ttyS0).
Es fácil, o no :-)?

Referencias

- The uisp AVR programmer software: www.amelek.gda.pl/avr/
local copy: [uisp-20011025.tar.gz](#)

- How to build the programmer hardware and install the AVR compiler:
March 2002, Programming the AVR Microcontroller with GCC
- The source code for this article [linuxfreqcount-0.4.tar.gz](#). The circuit diagram, the Eagle files and screen shoots are as well included.
- All software (updates will be listed here) and documents :[software/datasheets](#)
- Datasheet for 74hc390 [74hc390.pdf 48K](#)
- Datasheet for 74f74 [sn54f74.pdf 84K](#)
- Datasheet for LM393 [LM193.pdf 348K](#)
- Datasheet for MAX903 [MAX900-MAX903.pdf 164K](#)
- Datasheet for MAX232 [MAX220-MAX249.pdf 448K](#)
- Datasheet for ST232, a cheap variant, often sold instead of the real MAX232 [st232.pdf 100K](#)
- Datasheet for Atmel AT90S4433 [avr4433.pdf 2356K](#)
- The atmel website: [www.atmel.com/](#)
- Eagle for Linux [cadsoftusa.com](#)

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Guido Socher (homepage) en --> es: Mariano Candido Ucha <uchamariano/at@yahoo.com></p>
--	---