

Compteur de fréquence 1Hz–100Mhz avec afficheur LCD et interface RS232



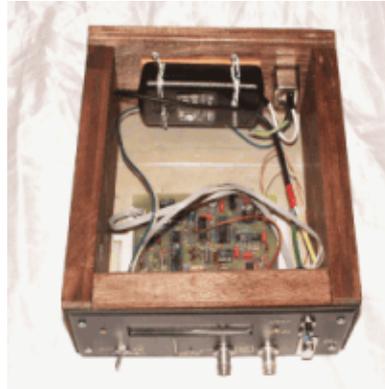
par Guido Socher ([homepage](#))

L'auteur:

Guido aime Linux, non seulement parce qu'il est intéressant d'en découvrir les immenses possibilités mais aussi pour les personnes qui sont impliquées dans sa conception.

Traduit en Français par:
Iznogood

[<iznogood/at/iznogood-factory.org>](mailto:iznogood/at/iznogood-factory.org)



Résumé:

Cet article continue la série sur le micro-contrôleur AT90S4433. Je vous suggère de lire les articles précédents sur la programmation d'Atmel :

1. Comment installer et utiliser l'environnement de développement d'AVR sous Linux et comment construire le programmeur : [March 2002, Programmer le micro-contrôleur AVR avec GCC](#)
2. Comment créer votre propre circuit imprimé : [Mai 2002, Un panneau de contrôle LCD pour votre serveur Linux](#)

Cette fois, nous concevons un compteur de fréquence capable de mesurer les fréquences de 1Hz à 100MHz. Vous pouvez aussi l'utiliser pour simplement compter les événements, comme le nombre de personnes ayant traversé la rue (tout ce qui est gérable sous forme d'impulsion numérique). Le système possède un afficheur LCD et une interface RS232 pour récupérer les résultats du comptage sous Linux.

Introduction

La fréquence d'une onde sinusoïdale ou carrée est exprimée en nombre d'oscillations par seconde. De manière à déterminer la fréquence d'un signal continu, il suffit de compter ces oscillations. De cette manière, nous déterminons la fréquence de la première harmonique d'un signal oscillant de manière continue. Pour mesurer les fréquences d'un « son » non continu, il faut un analyseur de spectre. C'est néanmoins un autre type de matériel. Ce que nous concevons ici est un compteur de fréquence pour les signaux oscillants de manière continue. Nous supposons que le signal ne change pas sa fréquence pendant un intervalle donné durant lequel nous faisons son échantillonnage.

Notre compteur de fréquence fonctionne en 2 étapes :

1. Il supprime toute composante continue du signal et la convertit en une onde carrée par l'utilisation d'un comparateur de tension.
2. Il compte le nombre d'impulsions par heure et le divise par cet intervalle.

Ce qu'il vous faut

Pour construire le compteur de fréquence, vous avez besoin des composants suivants :

- 1 x micro–contrôleur Atmel At90S4433
- 1 x support IC 28 broches 7,25 mm
- 2 x supports IC 16 broches
- 1 x support IC 1 broche
- 1 x support IC 14 broches
- Il n'y a pas de supports IC pour les MAX903 et 74F74. Il faut les souder directement sur la carte !
- 1 x MAX232
- 1 x quartz 4,194304MHz
- 1 x DEL (verte)
- 1 x transistor PNP BC557
- 4 x condensateurs 1uF (non polarisé)
- 2 x condensateurs céramique 27pF
- 4 x mini condensateurs céramique 10nF
- 3 x mini condensateurs céramique 100nF
- 1 x mini condensateur céramique 200nF
- 1 x mini condensateur céramique 470nF
- 2 x résistances de 470 Ohm
- 1 x résistance de 470K
- 2 x résistances de 100 Ohm
- 3 x résistances de 1k
- 5 x résistances de 10k
- 3 x résistances de 47K
- 1 x résistance de 220 Ohm
- 3 x résistances de 4K7
- 1 x résistance de 3k3
- 1 x résistance de 2k2
- 1 x résistance de 47 Ohm
- 1 x potentiomètre de 4K7 (aussi petit que possible) ; si vous voulez, vous pouvez remplacer ce potentiomètre par une paire de résistances. Ce potentiomètre est utilisé pour initialiser le contraste du LCD. Pour mon affichage LCD, une 100 Ohm et une 1K fournissent un bon contraste.
- 1 x diode zener 4.3V
- 2 x boutons petit contact
- 1 x condensateur 470uF (non polarisé)
- 1 x condensateur 4,7uF non polarisé)
- 1 x diode 1N4001
- 1 x 74HC02, TTL IC
- 1 x 74F74, TTL IC rapide
- 1 x 74HC390, TTL IC
- 1 x LM393, comparateur de tension

1 x MAX903, comparateur de tension rapide, en boîtier plastique 8 broches, vous pouvez le commander directement sur www.maxim-ic.com dans le cas où vous ne pouvez pas le trouver chez votre revendeur du coin.

1 x 7805, régulateur 5V

1 x afficheur LCD de 2 lignes de 20 caractères compatible HD44780. Avec ou sans rétro-éclairage (16 broches ou 14)

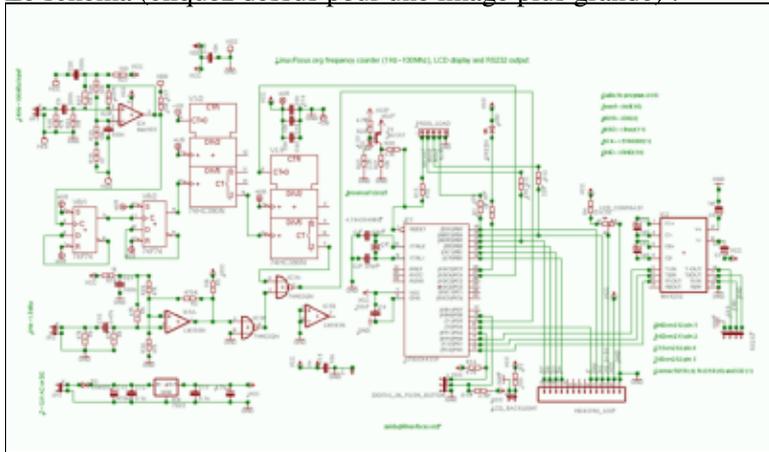
Tous les afficheurs LCD que j'ai vus avec des connecteurs 14 ou 16 broches étaient compatibles HD44780. Vous pouvez aussi utiliser des afficheurs de 3 ou 4 lignes mais vous devrez un peu modifier le logiciel.

En plus, vous aurez besoin de fils, de connecteurs (BCD, alimentation, RS232) et d'un transformateur 9V ou d'une alimentation de 150mA. Vous pouvez obtenir des alimentations très bon marché qui se branchent directement sur une prise et qui s'utilisent pour tout type d'électronique grand public.

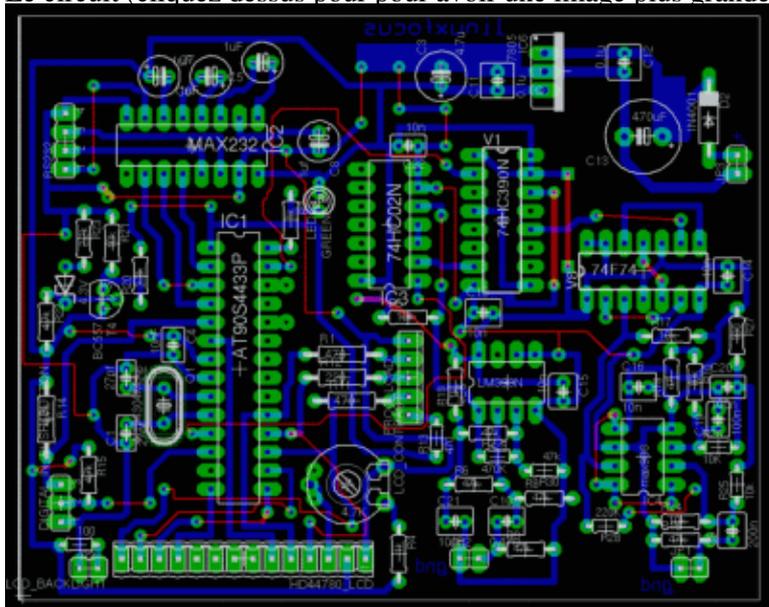
Schéma et circuit

J'ai utilisé [eagle](#) pour Linux afin de concevoir le schéma et le circuit. Le programme a eu quelques problèmes pour comprendre que toutes les broches d'alimentation du circuit sont en 5V. Vous obtiendrez donc quelques erreurs si vous lancez la vérification. La conception est néanmoins correcte.

Le schéma (cliquez dessus pour une image plus grande) :



Le circuit (cliquez dessus pour avoir une image plus grande) :



Le circuit est spécialement conçu pour les amateurs d'électronique. Seule la couche bleue est conçue pour être gravée en tant que circuit imprimé. Les lignes rouges sont les fils. Il est beaucoup plus facile et moins méticuleux de faire un circuit imprimé simple face. Vous pouvez disposer les fils (en rouge) de manière à ce qu'ils soient le plus courts possible, ce que je n'ai pas réussi à faire avec eagle.

La disposition du circuit avec un fond blanc pour une meilleure impression : [carte sur fond blanc](#) (Note: ce n'est pas le fichier dont vous avez besoin pour faire un circuit imprimé.)

Les fichiers eagle sont tous inclus dans le logiciel (linuxfreqcount-0.4) que vous pouvez télécharger à la fin de l'article.

Comment ça marche ?

Le micro-contrôleur AT90S4433 possède deux compteurs internes. L'un d'eux fait 16 bits et l'autre 8 bits. Nous utilisons le compteur 8 bits pour générer une base de temps appropriée à partir de la fréquence du quartz de l'AT90S4433. Dans ce but, nous utilisons un quartz de 4194304Hz et nous incrémentons le compteur 8 bits via un diviseur interne 1/256 (voir le téléchargement de la fiche technique de l'AT90S4433, à la fin de l'article). Le compteur 8 bits est configuré pour générer une interruption lors du débordement. En d'autres termes, nous obtenons une base de temps de $4194304\text{Hz} / (256 * 256) = 64\text{Hz}$. En utilisant une variable de boucle, nous générons depuis cette fonction des appels à intervalles de 1Hz ou de 64Hz.

Maintenant, nous avons une fonction qui est appelée à des intervalles de 1Hz ou 64Hz selon le mode de fonctionnement de notre compteur. Tout ce que nous devons faire maintenant est de lire le compteur 16 bits depuis cette fonction et d'afficher le résultat. Le compteur 16 bits (broche PD5 sur le micro-contrôleur) obtient ses informations d'impulsion à partir du signal que nous souhaitons mesurer.

Le micro-contrôleur échantillonne les signaux d'entrée pour les synchroniser avec son horloge interne. En fonction du théorème d'échantillonnage, nous pouvons donc mesurer un signal ayant au maximum la moitié de la fréquence du quartz. C'est la limite théorique. En pratique, nous pouvons mesurer des signaux jusqu'à 1,5 MHz avec le micro-contrôleur.

Pour mesurer des fréquences plus élevées, il nous faut un étalonneur/diviseur. C'est ce que proposent les circuits 74F74 et 74HC390. Le 74F74 est utilisé comme rapide diviseur 1/4 asynchrone et le 74HC390 est un diviseur 1/25. Nous ne pouvons pas utiliser le 74HC390 directement comme un 1/100 car il ne peut traiter plus de 25MHz.

Le circuit possède deux entrées. Une par le diviseur 1/100 et une entrée directe. En fonction de la fréquence du signal que vous souhaitez mesurer, vous utilisez une des deux lignes d'entrée (pas les deux).

Si vous êtes seulement intéressés par les fréquences jusqu'à 1.5 Mhz (i.e., pour faire des mesures de fréquence d'un amplificateur audio), vous pouvez aussi fabriquer une version simplifiée du compteur en abandonnant les MAX903, 74F74 et 74hc390. Vous pouvez toujours utiliser le même logiciel et le même circuit intégré.

L'objectif des deux comparateurs de tension (MAX903, haute vitesse et LM393) est d'amplifier les signaux et de générer des signaux d'ondes carrées à partir, par exemple, de signaux d'ondes sinusoïdales.

Une attention particulière a été nécessaire pendant la conception et l'élaboration du circuit imprimé afin d'éviter les oscillations du comparateur à proximité du seuil. Il est à noter que le MAX903 est réellement un comparateur à très haute vitesse. Si vous utilisez un signal d'entrée très bruyant, par exemple, un signal de 100KHz modulé avec un bruit à 1MHz, vous compterez alors quelques impulsions de 1 MHz lorsque le signal de 100KHz croise le niveau du 0 V.

La résistance de 220K sur le MAX903 évite jusqu'à un certain point l'influence du bruit en fournissant un retour positif. Vous ne devriez pas, néanmoins, alimenter le compteur avec des signaux bruyants.

Le micro–contrôleur peut être piloté par une RS232 et par deux boutons poussoir (bouton remise à zéro et mode de changement de comptage).

Les deux canaux d'entrée sont combinés par une porte OR (74hc02). La porte OR supprimera bien sûr le signal d'entrée si vous déconnectez l'entrée 1/100 alors qu'il y a encore un « 1 » logique sur la ligne de sortie du 74hc390. Le 74hc390 est alors remis à « 0 » (broche 2) lorsque vous changez le mode de comptage ou lorsque vous pressez le bouton de remise à zéro.

Le logiciel

Le logiciel du micro–contrôleur initialise deux compteurs en mode interruption. La manière de le faire est très bien décrite dans la fiche technique de l'AT90S4433 (voir les références). Vous devez définir un jeu de registres. C'est important mais assez ardu. Je ne vais donc pas le reproduire ici. La plupart de la logique du compteur de fréquence est implémentée dans le fichier [linuxfreqcount.c](#). Tous les autres fichiers sont des « bibliothèques » pour les LCD, UART, etc.... Lors de l'écriture de logiciel pour le micro–contrôleur, vous devez faire attention à ne pas utiliser plus de 128 octets de Ram. C'est tout ce que nous avons. Il est donc préférable d'éviter les appels de fonctions imbriquées avec beaucoup de paramètres. Les drapeaux et les variables globales sont plus adaptés.

Lorsque les registres du compteur sont configurés correctement alors la fonction SIGNAL(SIG_OVERFLOW0) est appelée avec des intervalles de 64 Hz. Ici, nous lisons le compteur 16 bits et initialisons un drapeau (hflag) pour pouvoir traiter le résultat du compteur avec la fonction handlecounterresult(). Avec une fréquence de porte de 1Hz, nous pouvons utiliser le résultat du compteur directement, mais avec une porte dont le temps d'ouverture est de 1/64 de seconde, nous devons multiplier le résultat par 64. Les opérations mathématiques réalisables avec le micro–contrôleur sont assez limitées (nous n'avons pas d'entiers 32 bits comme dans un processeur Pentium). Heureusement, multiplier par 64 correspond simplement à six décalages à gauche (6 fois shift left). Nous stockons le résultat dans des variables de 3 x 8 bits (counterval[3]).

Nous avons maintenant la valeur du compteur correcte sous forme de représentation 24 bits. De manière à l'afficher, il est nécessaire de le convertir en ASCII décimal. Printf serait capable de le faire mais nous ne l'avons pas car cela dépasserait nos possibilités de mémoire. Pour convertir le binaire en décimal, nous divisons le nombre 24 bits par 10 avec retenue. Les retenues sont des nombres décimaux. Notre processeur ne sait pas traiter les 24 bits, nous gérons donc le nombre par blocs de 8 bits (fonction divby10()). Ajouter la valeur ASCII zéro à chaque chiffre nous permet d'avoir enfin, une représentation ASCII de notre nombre (fonction longtoascii()).

Cette chaîne ASCII peut être envoyée par la ligne RS232 à l'afficheur LCD.

Réaliser le circuit imprimé

Le paquetage logiciel contient un fichier postscript (linuxfcount.ps) pour le circuit imprimé. Je trouve personnellement que les pastilles sont un peu trop petites. Je vous recommande donc de les agrandir avec un marqueur avant de graver le circuit. La marche à suivre pour réaliser le circuit chez soi est décrite dans l'article : [Mai 2002, Un panneau de contrôle LCD pour votre Serveur Linux](#). Quelques lecteurs m'ont indiqué que le spray de Kontakt Pausklar–21 est difficile à obtenir. A la place, nous pouvons utiliser du pétrole, comme celui utilisé dans les lampes. Vous devez nettoyer le pétrole avec un peu de savon avant de placer votre circuit dans le développeur.

Comment construire un boîtier pour notre compteur de fréquence ?



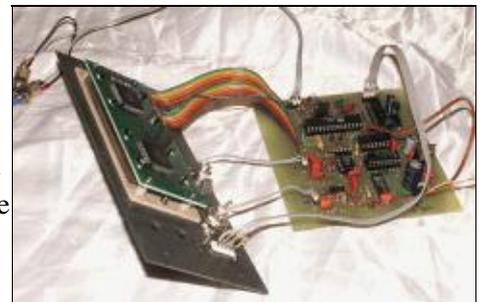
Un problème majeur pour les amateurs d'électronique est habituellement le boîtier contenant les circuits fabriqués. Les boîtiers pré-construits en aluminium sont assez chers. Travailler des feuilles de métal avec précision est aussi très difficile. J'ai trouvé une solution bon marché à l'aspect



professionnel. J'utilise du bois de sapin pour les cotés et le fond qu'il est possible de peindre pour lui donner un aspect assez cossu. Utilisez de la peinture soluble dans l'eau, à usage intérieur. Pour la façade, le dessus et le dessous, j'utilise des plaques de métal et elles n'ont pas besoin d'être pliées. Il faut simplement les couper à la bonne taille. Vous pouvez voir sur l'image de droite à quoi ressemble la partie en bois. Sur la gauche, vous pouvez voir la boîte terminée (le dessus est manquant).

Assembler la carte

Lors de la soudure des composants sur le circuit, vous devez faire attention à la zone haute fréquence (MAX903 et 74F74 avec des résistances et des condensateurs). Je recommande habituellement d'utiliser des supports pour circuit car cela facilite la recherche de pannes mais ces deux circuits ne doivent pas en utiliser afin de réduire les effets parasites indésirables. Les condensateurs de découplage entre la masse et l'alimentation doivent être en céramique. Leur valeur peut être comprise entre 10nF et 100nF.



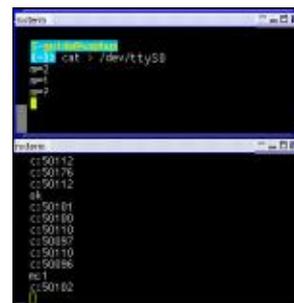
C'est le circuit le plus complexe qui ait été conçu dans LinuxFocus. Je vous recommande de le construire par étapes et de vérifier si les différents composants fonctionnent :

1. Soudez d'abord les composants nécessaires à l'alimentation (7805, etc...) sur la carte et vérifiez que tout fonctionne.
2. Soudez tous les composants sur la carte mais n'enfonchez que le micro-contrôleur dans son support.
3. Le paquetage linuxfreqcount-0.4 inclut un programme, `avr_led_lcd_test.c`, pour tester le micro-contrôleur, la LED et le LCD. Chargez-le. Branchez le câble du programmeur comme décrit dans le premier article sur l'AVR ([Mars 2002. Programmer le micro-contrôleur AVR avec GCC](#)) et tapez ensuite « `make testload` » pour charger le logiciel `avr_led_lcd_test` du paquetage `linuxfreqcount-0.4`. La LED doit clignoter et le LCD doit afficher le texte « Hello ».
4. Insérez tous les circuits dans leur support. Chargez le logiciel (`make load`). Testez d'abord la partie basse fréquence puis la haute fréquence. Vous avez besoin d'un oscillateur comme source de fréquence (préférez un signal de sortie sous forme d'onde carrée).

Utiliser le compteur

Le compteur peut opérer dans différents modes :

1. comptage continu
2. comptage de fréquence avec une porte de fréquence à 1 Hz. Dans ce mode, vous pouvez mesurer les fréquences jusqu'au maximum du compteur 16 bits, 65535 Hz ou 6553500 Hz avec un diviseur 1/100.
3. comptage de fréquence avec une porte de fréquence de 64 Hz. Dans ce mode, vous pouvez compter les fréquences jusqu'à 1,5 MHz ou 100 MHz avec un diviseur 1/100.



Interface de commande ASCII RS232.

Quelle précision ?

Tout compteur numérique a une limite de précision de ± 1 . Le dernier chiffre variera un peu même si vous avez un signal d'entrée très stable. Si la fréquence de la porte est de 64 Hz, alors le résultat est multiplié en interne par 64, ce qui signifie que la précision est de ± 64 . De la même manière, le diviseur 1/100 influence la précision du résultat. Le compteur est néanmoins très bon. Le principal problème est la dérive en température du quartz 4194304 Hz. Les quartz oscillent plus lentement lorsqu'ils sont plus chauds. C'est physique et vous pouvez seulement éliminer le problème en contrôlant exactement la température.

Il est possible de calibrer le compteur si vous trouvez un bon signal de référence. Vous pouvez le calibrer logiciellement en insérant des petits délais dans la fonction `handlecounterresult()`. J'ai comparé les résultats du compteur avec un signal de référence et mon compteur était assez exact. C'est la raison pour laquelle les délais de calibration sont en commentaires dans le code source. Vous pouvez aussi influencer un peu la fréquence du quartz en changeant un des condensateurs de 27pF (vous devez tester les 10pF, 50pF, etc..).

Pour une utilisation personnelle normale, il n'y a pas besoin de calibrer quoi que ce soit. Les quartz de 4194304 Hz sont très précis.

Le compteur en service.

Vous avez ici la photo finale du compteur en opération :



Pour une étude approfondie

Dans cet article, aucun logiciel d'application n'est fourni pour évaluer automatiquement les résultats sous Linux. On peut imaginer, par exemple, que vous utilisiez ce compteur de fréquence pour mesurer la vitesse du vent dans des zones éloignées ou tout autre événement pouvant être traduit en fréquences numériques. Ecrire des logiciels sous Linux qui font de la mesure de temps très long n'est pas très difficile. Vous pouvez par exemple modifier les programmes perl de l'article [Mai 2002, Un panneau de contrôle pour votre serveur Linux](#). Si vous êtes seulement intéressés dans l'enregistrement des résultats dans un fichier, vous devez alors utiliser les commandes :

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

Ceci suppose que le compteur est connecté sur COM1 (=ttyS0).
C'est facile, n'est-ce pas :-)?

Références

- Le logiciel de programmation AVR uisp: www.amelek.gda.pl/avr/
copie locale : [uisp-20011025.tar.gz](#)
- Comment construire le matériel de programmation et installer le compilateur AVR :
[Mars 2002, Programmer le micro-contrôleur AVR avec GCC](#)
- Le code source pour cet article [linuxfreqcount-0.4.tar.gz](#) . Le schéma du circuit, les fichiers Eagle et les captures d'écran sont aussi inclus.
- Tous les logiciels (les mises à jour seront listées ici) et les documents : [software/datasheets](#)
- Fiche technique pour le 74hc390 [74hc390.pdf](#) 48K
- Fiche technique pour le 74f74 [sn54f74.pdf](#) 84K
- Fiche technique pour le LM393 [LM193.pdf](#) 348K
- Fiche technique pour le MAX903 [MAX900-MAX903.pdf](#) 164K
- Fiche technique pour le MAX232 [MAX220-MAX249.pdf](#) 448K
- Fiche technique pour le ST232, une variante bon marché, souvent vendue à la place du MAX232
[st232.pdf](#) 100K
- Fiche technique pour l'Atmel AT90S4433 [avr4433.pdf](#) 2356K
- Le site web atmel : www.atmel.com/
- Eagle pour Linux cadsoftusa.com

Site Web maintenu par l'équipe d'édition LinuxFocus

© Guido Socher

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher ([homepage](#))

en --> fr: Iznogood <iznogood@iznogood-factory.org>