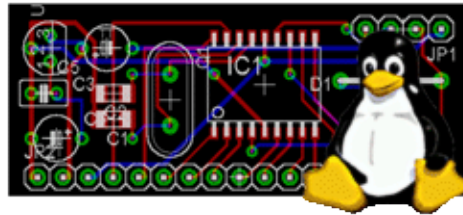




par Aleksandar Milovac
<amilovac(at)ptt.yu>

Faisons un peu d'électronique



L'auteur:

Aleksandar est jeune diplômé de la faculté des Sciences Techniques à Novi Sad, département électronique et télécommunications. Linuxien depuis 1999, il utilise Debian GNU/Linux depuis maintenant deux ans.

Résumé:

Alors que j'étais encore élève ingénieur en électronique (autrement dit, récemment), j'ai eu de nombreuses occasions de travailler avec des outils commerciaux tels que Protel, OrCAD et moult compilateurs commerciaux... Aussi, quand j'ai découvert Linux (début 99), je me suis posé la question suivante : "Tout cela est-il possible avec Linux ?". Après quatre ans d'investigations, je suis fier de pouvoir l'annoncer : "Oui, c'est tout à fait possible !". Et c'est pourquoi je tiens à vous présenter dans cet article les quelques logiciels dont vous aurez besoin pour cela sous Linux.

Traduit en Français par:

Guillaume Baudot

<guillaume.baudot(at)caramail.com>

Avant propos

Si Linux ne constituait guère qu'un hobby de passionnés à ses débuts en 1991, le projet a bien évolué depuis : il s'agit dorénavant d'un système moderne et performant, multi-utilisateurs, multi-tâches : sa réputation n'est plus à faire, que ce soit en termes de robustesse, de fiabilité... Alors, que vous soyez vous-même ingénieur en électronique ou simple amateur, vous pourrez apprécier ses nombreux avantages.

Mais il y'a encore plus important, Linux est un projet Open Source (disons libre, si ce n'est pas sujet à polémique). Et cela constitue un atout majeur en sa faveur, car la licence GPL est conçue de façon à favoriser le partage des connaissances et des idées. Vous pourrez donc, par exemple, gagner du temps en récupérant des programmes ou portions de code déjà écrits et vous consacrer ainsi plus pleinement à la conception matérielle. En outre, il se trouve que Linux et nombre de programmes libres sont particulièrement bien documentés, sans parler de l'excellence du support technique : il suffit le plus souvent d'un message à l'auteur d'un programme, ou d'une inscription dans la liste de diffusion appropriée pour constater la réactivité et la disponibilité de la communauté. Je vous laisse le soin de compléter par vous-même la longue liste des arguments en faveur de Linux.

Certains de mes amis, peu convaincus pour ne pas dire dubitatifs, m'ont objecté qu'ils s'interrogeaient sur l'intérêt de s'engager ainsi vers le Grand Inconnu, alors qu'ils s'étaient familiarisés avec tel outil ou tel autre. De tels doutes et appréhensions sont bien naturels, mais le fait est qu'il est impossible de se faire une opinion

sans en avoir soi-même fait l'expérience auparavant. Qu'ajouter alors, sinon : "essayez et voyez ce qu'il en est" ?

Que choisir ?

Quiconque s'essaie à la fabrication de circuits électroniques commence par concevoir les schémas, avant de réaliser le circuit imprimé qui servira ensuite à assembler les composants nécessaires. Et grâce aux progrès incessants de l'informatique, tout ceci est désormais accessible au commun des mortels : rien ne vous empêche de le faire dans votre propre salon (ou bureau, chambre...), bien calé dans votre fauteuil, entre une bonne bière et un paquet de chips – voire une ouïche lorraine ;-) pour les plus gourmands. Nous pourrions bien entendu nous tourner vers les grands classiques (Protel, Pcad, OrCAD...), mais rares sont les bourses à pouvoir s'offrir un tel luxe ! Fort heureusement pour nous, il existe des alternatives bien moins onéreuses, puisque gratuites.

Pour commencer, je vous présenterai le programme de modélisation EAGLE, édité par Cadsoft (www.cadsoftusa.com). Il n'est malheureusement pas distribué sous license GPL et son utilisation est donc soumise à quelques contraintes. En outre, les dimensions des cartes sont limitées à 100*80mm sur deux couches. Mais que ces imperfections ne vous freinent pas, EAGLE n'en reste pas moins un excellent logiciel, qui devrait vous permettre de belles réalisations. Il est léger, peu gourmand en mémoire, avec une interface claire et un gestionnaire de bibliothèques aussi simple qu'intuitif. Enfin, si sa logique nécessite un certain temps d'adaptation, il n'y a là rien d'insurmontable.

Aux côtés d'EAGLE, nous trouvons la gamme des programmes utilisés tant dans les universités que les entreprises, et parmi eux le célèbre Matlab (mathworks.com), qui constitue presque un standard en calcul numérique, traitement du signal, modélisation de systèmes... Toutefois, bien qu'il en existe une version pour Linux, je vous suggérerai plutôt de conserver vos deniers et d'essayer un autre programme nommé Scilab (scilabsoft.inria.fr). Le programme et son code source sont téléchargeables gratuitement, son installation nécessite au préalable X et la bibliothèque Tcl, que vous pourrez obtenir tout aussi aisément. Pratiquement, il est capable de tout ce que peut Matlab et sa syntaxe est similaire. Pour comble, il dispose aussi d'une boîte à outils analogue à celle de Matlab, Simulink : il s'agit de Scicos (www.scicos.org).

Venons en maintenant aux microcontrôleurs et à leur programmation. Et bien, parmi les points forts de Linux, il faut compter le support de multiples langages de programmation, ce qui en fait une fois de plus un outil parfaitement adapté à nos besoins : inutile donc de se ruiner avec un compilateur commercial ! Ainsi, si vous comptez programmer différents types de microcontrôleurs et autres EEPROM, je vous recommanderais PonyProg (www.lancos.com/prog.html). Accompagné du matériel adéquat, il vous permettra de programmer PIC, AVR et de nombreux autres types d'EEPROM. Fonctionnant sous X, c'est un modèle de simplicité et d'efficacité.

Pour les inconditionnels d'AVR, il existe aussi SP12 : fourni avec les schémas de la partie matérielle, c'est un programmeur en ligne de commande, d'utilisation très aisée. L'ayant testé pour un projet au cours de mes études, je puis vous affirmer qu'il fonctionne à merveille ! Mais ce n'est bien entendu pas le seul : entre autres solutions, il est même possible de programmer en C, voire en assembleur. À qui ne jure que par le C, je ne saurais trop recommander le compilateur AVR-GCC. Les intéressés sont invités à lire ou relire l'excellent article réalisé par Guido Socher à ce sujet.

Les amoureux des PIC devraient également trouver leur bonheur. Utilisateur de Debian GNU/Linux, j'ai recherché et trouvé ces quelques programmes dédiés au PIC. Le premier, Picasm porte bien son nom, car c'est un assembleur pour microcontrôleurs PIC. Et le second également : Simulpic est un programme de simulation de PIC. Faute d'avoir travaillé sur PIC, je n'en ai testé aucun pour ma part, mais que cela ne vous empêche pas d'essayer vous-même (que ce soit avec Debian, ou toute autre version de Linux...). Pour conclure sur les PIC,

il faut encore mentionner le programmeur [Picprg \(www.brianlane.com\)](http://www.brianlane.com) de Brian Lane : son utilisation ne devrait pas vous poser de problèmes...

Et enfin, je ne saurais traiter d'électronique, programmation et autres opérations sous Linux, sans m'autoriser cette digression : depuis quelques temps, un effort considérable a été fourni pour adapter Linux aux contraintes des systèmes embarqués ou temps réel. Donc, si vous désirez faire de votre Linux un système d'exploitation temps réel, par exemple pour contrôler un processus industriel ou un robot, apprenez que les deux extensions les plus populaires pour Linux sont [RTLlinux \(www.fsmlabs.com\)](http://www.fsmlabs.com) et [RTAI \(www.rtai.org\)](http://www.rtai.org) : l'une comme l'autre sont distribuées sous licence GPL et dotées d'une documentation pléthorique.

Conclusion

De ce rapide passage en revue, nous pourrions retenir que le monde du logiciel libre offre quantité de solutions aux électroniciens de tout poil, amateurs comme professionnels. Et puisque la qualité aussi est au rendez-vous, cela fait de Linux une solution de choix.

À titre personnel, je n'ai eu de cesse au cours des quatre dernières années, de trouver pour Linux des alternatives à l'ensemble des logiciels qui m'étaient coutumiers sous Windows. J'ai fait en sorte de tout remplacer et, bien que je sois toujours en quête d'améliorations, je suis d'avis que les choses se sont déjà grandement simplifiées.

Références

- [Debian GNU/Linux](#)
- [Eagle](#)
- [Scilab](#)
- [Le programmeur PonyProg](#)
- [Le programmeur SP12](#)
- [Le programmeur Picprg](#)
- [RTLlinux](#)
- [RTAI](#)
- [AVR-GCC libc](#)
- [Article LF231 : Programmer le micro-contrôleur AVR avec GCC](#)

| | |
|---|--|
| <p><u>Site Web maintenu par l'équipe d'édition</u> LinuxFocus © Aleksandar Milovac "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p> | <p>Translation information: sr --> -- : Aleksandar Milovac <amilovac(at)ptt.yu> sr --> en: Aleksandar Milovac <amilovac(at)ptt.yu> en --> fr: Guillaume Baudot <guillaume.baudot(at)caramail.com></p> |
|---|--|