

# Nous voulons (aussi) du matériel libre !

*...nous dit Richard Stallman. Le logiciel a ses licences libres, la culture et la création artistique ont aussi leurs licences libres. Mais depuis quelques années les créations physiques libres se multiplient, ce qui rend urgent de poser les questions des libertés associées aux objets, comme aux modèles de conception qui permettent de les réaliser. Avec beaucoup d'acuité et de précision, Richard Stallman examine ces nouveaux problèmes et évalue la possibilité d'adapter à ce nouveau domaine les principes libristes employés pour les logiciels.*

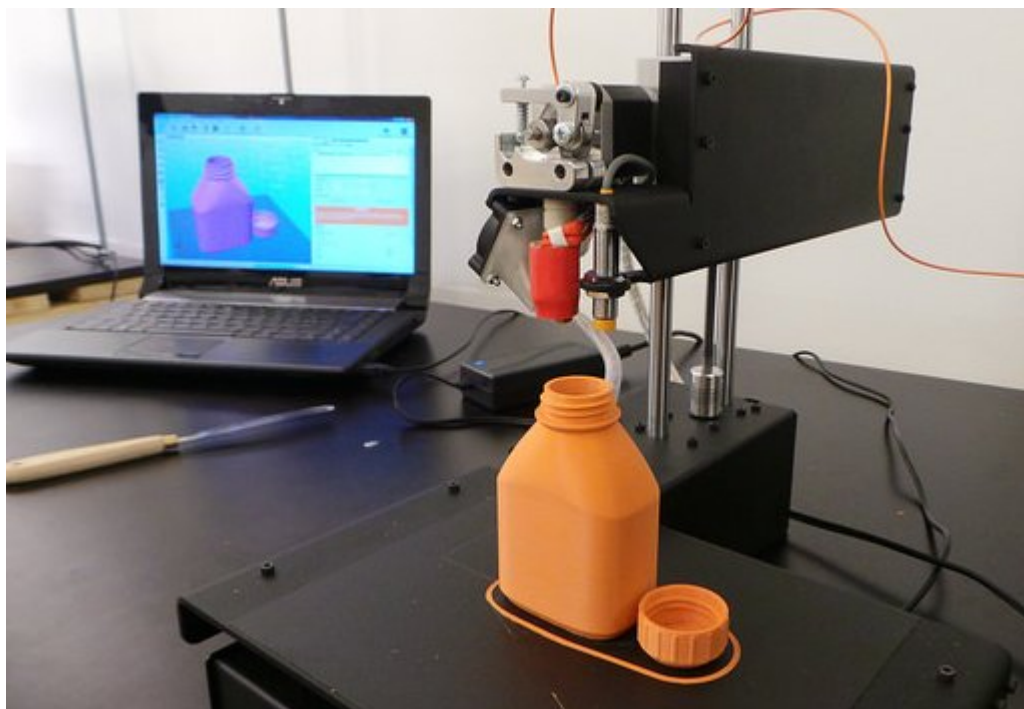
La version originale en anglais de cet article a été publiée en deux parties dans [wired.com](http://wired.com) en mars 2015 :

1. [Pourquoi nous avons besoin de plans libres pour le matériel](#)
2. [Les plans du matériel doivent être libres, voici comment procéder](#)

L'article ci-dessous reprend [le texte publié sur le site \[gnu.org\]\(http://gnu.org\)](#). C'est le fruit d'une traduction collaborative entre Sébastien Poher et Framalang (Thérèse Godefroy, aziliz, Omegax, Piup, r0u, sebastienc, goofy) – Révision : trad-gnu@april.org.

## Matériel libre et plans libres pour le matériel

par [Richard M. Stallman](#)



Dans quelle mesure peut-on appliquer les idées du logiciel libre au matériel ? Est-ce une obligation morale de rendre libres les plans de notre matériel, tout comme c'en est une de rendre libre notre logiciel ? Est-ce que conserver notre liberté nous oblige à refuser le matériel construit sur des plans non libres ?

## Définitions

L'expression « logiciel libre »<sup>1</sup> se réfère à la liberté et non au prix ; en gros, elle signifie que les utilisateurs du logiciel sont libres de le faire fonctionner, de le copier et de le redistribuer, avec ou sans modification. Plus précisément, la définition repose sur [les quatre libertés essentielles](#). Pour bien faire ressortir que *free* fait référence à la liberté et non au prix, nous accolons souvent le mot français ou espagnol « libre » à *free*.

Si l'on transpose directement ce concept au matériel, *matériel libre* signifie que l'utilisateur est libre de l'utiliser, de le copier et de le redistribuer, avec ou sans modification. Toutefois, il n'y a pas de système de copie pour les objets matériels, à

part la reproduction des clefs, de l'ADN et de la forme extérieure d'objets plastiques. La plupart des composants matériels sont fabriqués sur plan. Le plan précède le matériel.

Par conséquent, le concept dont nous avons vraiment besoin est celui de *plan libre pour le matériel*. C'est simple, il s'agit d'un plan dont l'utilisateur peut se servir (c'est-à-dire à partir duquel il peut fabriquer le matériel) et qu'il peut copier et redistribuer, avec ou sans modification. Un tel plan doit offrir les quatre libertés, évoquées plus haut, qui définissent le logiciel libre. Le terme « matériel libre » désigne alors un matériel pour lequel un plan libre est disponible.

Lorsque les gens découvrent le concept de logiciel libre pour la première fois, ils l'assimilent souvent à « possibilité d'en obtenir gratuitement un exemplaire ». Il est vrai que beaucoup de programmes libres sont disponibles à prix nul, puisque cela ne vous coûte rien de télécharger votre propre copie, mais ce n'est pas le sens de *free* dans *free software* (de fait, certains programmes espions tels que [Flash Player](#) ou [Angry Birds](#) sont gratuits, bien qu'ils ne soient pas libres). Accoler le mot « libre » à *free* aide à clarifier ce point.

Pour le matériel, cette confusion tend à aller dans l'autre direction ; les composants matériels ont un coût de production, donc ceux qui sont produits dans un but commercial ne peuvent être gratuits (à moins qu'il ne s'agisse de produits d'appel ou de vente liée), mais cela n'empêche pas leurs plans d'être libres. Les objets que vous fabriquez avec votre imprimante 3D peuvent être très bon marché, mais ils ne seront pas totalement gratuits car vous devez payer pour le matériau brut. D'un point de vue éthique, l'enjeu de la liberté prime absolument sur l'enjeu du prix, car un appareil qui refuse la liberté à ses utilisateurs vaut moins que rien.

Les expressions « matériel ouvert » et « matériel open source

» sont utilisées par certains avec la même signification que « matériel libre », mais elles minimisent la problématique de la liberté. Elles sont dérivées de l'expression « logiciel open source », qui correspond plus ou moins au logiciel libre, mais [passe sous silence la liberté et ne présente pas la problématique en termes d'opposition bien-mal](#). Pour souligner l'importance de la liberté, nous ne manquons pas d'y faire référence chaque fois que cela est pertinent ; puisqu'« open » ne permet pas cette distinction, ne le substituons pas à « libre ».

## **Matériel et logiciel**

Le matériel et le logiciel sont fondamentalement différents. Un programme, même sous sa forme compilée exécutable, est une collection de données qui peut être interprétée comme instructions par un ordinateur. Comme toute autre création numérique, on peut le copier et le modifier en se servant d'un ordinateur. Un exemplaire de programme n'a pas de forme physique ni matérielle intrinsèque.

Au contraire, le matériel est une structure physique et sa nature physique est cruciale. Alors que le plan du matériel peut être représenté par des données, même par un programme dans certains cas, le plan n'est pas le matériel. Le plan d'un CPU ne peut pas exécuter un programme. Vous n'irez pas très loin en essayant de taper sur le plan d'un clavier ni en affichant des pixels sur le plan d'un écran.

De plus, alors que vous pouvez vous servir d'un ordinateur pour modifier ou copier le plan d'un composant matériel, un ordinateur ne peut pas convertir ce plan dans la structure physique qu'il décrit. Cela demande un équipement de fabrication.

## **La frontière entre matériel et logiciel**

Où se situe la frontière, dans un appareil numérique, entre

matériel et logiciel ? Nous pouvons la déterminer en appliquant les définitions suivantes : le logiciel est la partie opérante d'un appareil qui peut être copiée et modifiée dans un ordinateur ; le matériel est la partie opérante qui ne peut pas l'être. C'est le meilleur moyen de faire la distinction car cela renvoie à des conséquences pratiques.

Entre le matériel et le logiciel, il existe une zone grise occupée par les micrologiciels [*firmware*] qui *peuvent* être mis à jour ou remplacés, mais ne sont pas conçus pour être mis à jour ou remplacés une fois le produit vendu. En termes conceptuels, la zone grise est plutôt mince. En pratique, ceci est important car de nombreux produits sont concernés. Nous pouvons donc presque considérer ces micrologiciels comme du matériel.

Certaines personnes ont prétendu que préinstaller des micrologiciels et des circuits logiques programmables de type « FPGA » (réseau de portes programmables *in situ*) « effacerait la frontière entre matériel et logiciel », mais je pense qu'il s'agit là d'une mauvaise interprétation des faits. Un micrologiciel installé lors de l'utilisation est un logiciel ordinaire ; un micrologiciel intégré à un appareil et non modifiable est un logiciel par nature, mais nous pouvons le traiter comme si c'était un circuit. Dans le cas des FPGA, le circuit FPGA lui-même est du matériel, mais le schéma logique de portes chargé dans ce circuit FPGA est une sorte de micrologiciel.

Faire fonctionner des schémas de portes libres sur des FPGA pourrait être une méthode utile pour créer des appareils numériques libres au niveau des circuits. Toutefois, pour rendre ces FPGA utilisables dans le monde du libre, nous avons besoin d'outils de développement libres adaptés. L'obstacle à cela est que le format du fichier de schéma logique de portes chargé dans le FPGA est secret. Jusqu'à récemment, il n'existait *aucun* modèle de FPGA pour lequel ces fichiers pouvaient être reproduits sans outils non libres (privateurs).

Grâce aux efforts de rétroingénierie, il est maintenant possible de compiler des programmes écrits en C et de les exécuter sur le circuit FPGA Xilinx Spartan 6 LX9. Ces outils ne gèrent pas encore le code HDL (langage de description de matériel), cela n'offre donc pas une alternative utilisable pour de vrais circuits numériques. D'un autre côté, ce modèle de FPGA se fait vieux. Ces outils constituent une formidable avancée en comparaison de la situation d'il y a quelques années, mais la route est encore longue avant que les FPGA ne soient totalement utilisables en toute liberté.

Quant au code HDL lui-même, il peut agir en tant que logiciel (lorsqu'il s'exécute dans un émulateur ou après chargement dans un FPGA) ou en tant que schéma de composant matériel (lorsqu'il est mis en œuvre dans des semiconducteurs inamovibles ou dans un circuit imprimé).

## **La question éthique posée par les imprimantes 3D**

Sur le plan éthique, [le logiciel doit être libre](#), car un programme non libre est source d'injustice. Faut-il avoir la même optique en ce qui concerne la conception de matériel ?

Il le faut, sans aucun doute, dans les domaines concernés par l'impression 3D (ou plus généralement par toute fabrication personnelle). Les modèles servant à imprimer des objets utiles, pratiques (c'est-à-dire fonctionnels plutôt que décoratifs) *doivent* être libres parce qu'ils sont réalisés dans un but utilitaire. Les utilisateurs doivent en avoir le contrôle, tout comme ils doivent avoir le contrôle des logiciels qu'ils utilisent. Diffuser un modèle non libre pour un objet fonctionnel est aussi mal que de diffuser un programme non libre.

Soyez attentifs à choisir des imprimantes 3D fonctionnant avec du logiciel exclusivement libre ; la Free Software Foundation [homologue de telles imprimantes](#). Certaines imprimantes 3D sont

conçues à partir de plans libres mais la [Makerbot est de conception non libre](#).

## **Devons-nous refuser le matériel numérique non libre ?**

La conception non libre d'un matériel numérique (\*) engendret-elle une injustice ? Devons-nous, au nom de nos libertés, rejeter tous les matériels numériques créés à partir de plans non libres, tout comme nous devons rejeter le logiciel non libre ?

Étant donné qu'il y a un parallèle conceptuel entre les plans du matériel et le code source du logiciel, beaucoup de bidouilleurs de matériel informatique sont prompts à condamner les plans non libres au même titre que le logiciel non libre. Je ne suis pas d'accord, car la situation est différente pour le matériel et le logiciel.

De nos jours, les technologies de fabrication des puces et des circuits ont beaucoup de similitudes avec la presse à imprimer : elles se prêtent parfaitement à une production de masse dans une usine. Cela ressemble plus à la façon dont on copiait des livres en 1950 qu'à la façon dont on copie du logiciel aujourd'hui.

La liberté de copier et de modifier le logiciel est un impératif éthique car ces activités sont à la portée de ceux qui utilisent le logiciel : l'équipement qui vous permet d'utiliser le logiciel (un ordinateur) est suffisant pour le recopier et le modifier. Les ordinateurs mobiles d'aujourd'hui sont trop peu puissants pour être adaptés à cet usage, mais n'importe qui peut se procurer un ordinateur assez puissant.

De plus, même si vous-même n'êtes pas programmeur, il vous suffit d'un ordinateur pour télécharger et exécuter la version modifiée par une personne capable d'effectuer cette modification. Et de fait les non-programmeurs téléchargent des

logiciels et en font tourner tous les jours. C'est pourquoi le logiciel libre change véritablement les choses pour eux.

Dans quelle mesure cela est-il transposable au matériel ? Ceux qui utilisent du matériel numérique ne savent pas toujours modifier le schéma d'un circuit ou d'une puce, mais tous ceux qui possèdent un ordinateur ont l'équipement nécessaire pour le faire. Jusque-là, le matériel et le logiciel sont similaires, mais ensuite vient la grande différence.

Vous ne pouvez pas compiler et exécuter les plans d'un circuit ou d'une puce sur votre ordinateur. Construire un circuit complexe représente beaucoup de travail méticuleux et cela suppose que vous disposiez de la carte. Fabriquer une puce n'est pas faisable par une personne isolée aujourd'hui ; seule la production de masse permet de réduire suffisamment les coûts. Avec la technologie actuelle, les utilisateurs ne peuvent pas télécharger et faire tourner le plan d'un composant numérique modifié par Amélie la bricoleuse, alors qu'ils peuvent le faire avec un logiciel modifié par Amélie la programmeuse. Ainsi, les quatre libertés ne donnent pas aux utilisateurs le même contrôle collectif sur un plan de matériel que sur un logiciel. C'est à partir de là que le raisonnement démontrant que tout logiciel doit être libre cesse de s'appliquer aux technologies actuelles de fabrication du matériel.

En 1983, il n'y avait pas de système d'exploitation libre mais il était clair que si nous en disposions, nous pourrions l'utiliser immédiatement et accéder à la liberté du logiciel. Tout ce qui manquait était le code d'un système de ce type.

En 2015, même si nous avons des plans libres pour un processeur de PC, les puces produites en série à partir de ces plans ne nous donneraient pas la même liberté dans le domaine du matériel. Si nous achetons un produit fabriqué en série dans une usine, cette dépendance envers l'usine cause la plupart des problèmes rencontrés avec les plans non libres.



Pour que des plans libres nous donnent la liberté du matériel, nous avons besoin de technologies de fabrication nouvelles.

Nous pouvons envisager un avenir dans lequel nos « fabricateurs » personnels pourront fabriquer des puces et nos robots pourront les assembler et les souder avec les transformateurs, interrupteurs, touches, écrans, ventilateurs, etc. Dans ce monde futur, nous fabriquerons tous nos propres ordinateurs (ainsi que nos fabricateurs et nos robots) et nous serons en mesure d'utiliser à notre avantage les plans modifiés par ceux qui maîtrisent la conception du matériel. Les arguments pour rejeter le logiciel non libre s'appliqueront alors aussi aux plans du matériel.

Ce monde futur se situe à plusieurs années de distance, au moins. En attendant, il n'est pas nécessaire de rejeter par principe le matériel dont les plans ne sont pas libres.

---

\* Ici, « matériel numérique » recouvre également le matériel utilisant quelques circuits et composants analogiques en plus des circuits et composants numériques.

## **Nous avons besoin de plans libres pour le matériel numérique**

Bien que dans la situation actuelle nous n'ayons pas besoin de rejeter le matériel numérique issu de plans non libres, nous avons besoin de développer des plans libres et nous devons les utiliser quand c'est faisable. Ils procurent des avantages aujourd'hui, et à l'avenir ils pourraient être la seule façon d'utiliser du logiciel libre.

La disponibilité de plans libres pour le matériel offre des avantages pratiques. De nombreuses entreprises peuvent fabriquer le même, ce qui réduit la dépendance à un fournisseur unique. Des groupes peuvent s'organiser pour les

fabriquer en grande quantité. Lorsqu'on possède les schémas ou le code HDL des circuits, on peut étudier les plans des composants pour y déceler d'éventuelles erreurs ou fonctionnalités malveillantes (il est connu que la NSA a introduit des faiblesses intentionnelles dans certains matériels informatiques). En outre, ces plans libres pourraient servir de modules élémentaires dans la conception d'ordinateurs et autres appareils complexes, dont les spécifications seraient publiées et dont moins de composants pourraient être utilisés contre nous.

Il nous sera peut-être possible d'utiliser des schémas libres pour certains composants de nos ordinateurs et de nos réseaux, ainsi que pour les systèmes embarqués, avant de pouvoir nous en servir pour fabriquer des ordinateurs complets.

Les plans libres pour le matériel pourraient même devenir essentiels avant que nous puissions fabriquer le matériel nous-mêmes, s'ils deviennent la seule façon d'éviter le logiciel non libre. Comme le matériel commercial le plus courant est de plus en plus conçu pour assujettir les utilisateurs, il devient de moins en moins compatible avec le logiciel libre, car ses spécifications sont confidentielles et il oblige le code à être certifié par quelqu'un d'autre que vous. Les puces des modems de téléphones mobiles et même certains accélérateurs graphiques exigent déjà un micrologiciel certifié par le fabricant. Tout programme, tournant sur votre ordinateur, que quelqu'un d'autre a le droit de modifier mais pas vous, est un instrument de pouvoir injuste envers vous ; du matériel qui impose cette exigence est du matériel malveillant. En ce qui concerne les puces des modems de téléphones mobiles, tous les modèles actuellement disponibles sont malveillants.

Un jour, le matériel numérique de conception libre pourrait être l'unique plateforme permettant de faire tourner du logiciel libre. Ayons pour but d'avoir à notre disposition d'ici là les plans numériques libres requis, et espérons que

nous aurons les moyens de fabriquer le matériel correspondant à des coûts suffisamment bas pour tous les utilisateurs.

Si vous concevez du matériel, veuillez s'il vous plaît libérer vos plans. Si vous utilisez du matériel, rejoignez notre effort en insistant auprès des entreprises, en faisant pression sur elles, pour qu'elles rendent libres les plans de leur matériel.

## Niveaux de conception

Le logiciel a plusieurs niveaux de conception ; par exemple, un paquet peut inclure des bibliothèques, des commandes et des scripts. Mais du point de vue de la liberté du logiciel, ces niveaux ne présentent pas de différence significative parce qu'il est possible de rendre libre chacun d'entre eux. La conception des composants d'un programme est de même nature que la conception du code qui les combine ; de même, compiler les composants à partir de leur code source est de même nature que compiler le programme à partir de son code source. Rendre l'ensemble libre nécessite simplement de poursuivre le travail jusqu'à ce qu'il soit terminé.

Par conséquent, nous insistons pour que les programmes soient libres à tous les niveaux. Pour qu'un programme puisse être considéré comme libre, chaque ligne de code qui le compose doit être libre, de sorte qu'on puisse le compiler à partir du seul code source libre.

Les objets physiques, en revanche, sont souvent construits à partir de composants qui eux-mêmes sont conçus et construits dans différentes sortes d'usines. Par exemple, un ordinateur est constitué de puces, mais la conception (ou la fabrication) des puces est très différente de la conception (ou de la fabrication) de l'ordinateur à partir de ces puces.

Ainsi, nous avons besoin de distinguer des *niveaux* dans les plans d'un produit numérique (et peut-être de certains autres

types de produits). Le circuit qui connecte les puces est l'un de ces niveaux ; le plan de chaque puce en est un autre. Dans un FPGA, l'interconnexion des cellules primitives constitue un niveau, tandis que les cellules primitives elles-mêmes en sont un autre. Dans un avenir idéal, nous voudrions que le plan soit libre à tous les niveaux. Dans les circonstances actuelles, le simple fait de rendre libre un niveau est une amélioration significative.

Cependant, si à l'un des niveaux le plan fait appel à des parties libres et à des parties non libres – par exemple un circuit HDL « libre » qui incorpore des processeurs *softcore* privés – nous devons conclure que le plan dans son ensemble est non libre à ce niveau. De même pour les « assistants » ou les « macros » non libres, s'ils définissent une partie des interconnexions des puces ou bien les parties des puces dont les connexions sont programmables. Les parties libres peuvent constituer une étape vers notre objectif futur de liberté des plans, mais pour atteindre cet objectif il faudra remplacer les parties non libres. Elles ne pourront jamais être acceptables dans le monde du libre.

## **Licences et copyright adaptés aux plans libres pour le matériel**

Pour rendre libre le plan d'un matériel, il faut le publier sous une licence libre. Nous recommandons la licence publique générale GNU, version 3 ou ultérieure. Nous avons conçu la version 3 de la GPL en envisageant cet usage.

Placer des circuits ou des formes d'objets non décoratifs sous copyleft ne va pas aussi loin que l'on pourrait le supposer. Le copyright sur ces plans s'applique uniquement à la manière dont le plan est dessiné ou écrit. Le copyleft étant une façon d'utiliser le droit du copyright, son effet ne se fait sentir que dans la mesure où le copyright s'applique.

Par exemple, un circuit, en tant que topologie, ne peut pas

faire l'objet d'un copyright (et par conséquent ne peut pas être placé sous copyleft). Les définitions de circuits écrit en HDL peuvent être placées sous copyright (et donc sous copyleft), mais le copyleft ne régit que les détails de l'expression du code HDL, pas la topologie du circuit qu'il génère. De même, un dessin ou le diagramme d'un circuit peut être placé sous copyright, donc sous copyleft, mais cela ne couvre que le dessin ou le diagramme, pas la topologie du circuit. N'importe qui peut légalement dessiner le même circuit d'une façon apparemment différente, ou écrire une définition différente en code HDL qui produise le même circuit.

Étant donné que le copyright ne régit pas les circuits physiques, lorsque des gens construiront des exemplaires du circuit, la licence des plans n'aura aucune incidence juridique sur ce qu'ils feront avec les appareils qu'ils auront construits.

En ce qui concerne les dessins d'objets et les modèles pour imprimantes 3D, le copyright ne s'applique pas à un dessin différent de la forme du même objet purement fonctionnel. Il ne couvre pas non plus les objets physiques utilitaires fabriqués à partir du dessin. Sous le régime du copyright, chacun est libre de les fabriquer et de les utiliser (et c'est une liberté dont nous avons grand besoin). Aux États-Unis, le copyright ne régit pas les aspects fonctionnels de ce que décrit le plan, mais en revanche [il couvre les aspects décoratifs](#). Quand un objet a des aspects décoratifs et des aspects fonctionnels, on se trouve en situation délicate (\*).

Tout ceci est peut-être vrai également dans votre pays, ou non. Avant de produire des objets pour un usage commercial ou en grande quantité, vous devriez consulter un juriste local. Le copyright n'est pas le seul problème qu'il vous soit nécessaire de prendre en compte. Vous pourriez être attaqué au plan des brevets, très probablement détenus par des entités qui n'avaient rien à voir avec l'élaboration des plans que

vous utilisez ; et d'autres problèmes juridiques peuvent aussi se présenter.

Gardez à l'esprit que le droit du copyright et le droit des brevets sont radicalement différents. C'est une erreur de supposer qu'ils aient quoi que ce soit en commun. C'est pourquoi le terme « [propriété intellectuelle](#) » est source de pure confusion et doit être totalement rejeté.

---

\* Un article de Public Knowledge donne des renseignements utiles sur cette [complexité](#) (pour ce qui est des États-Unis) bien qu'il tombe dans l'erreur commune consistant à utiliser l'expression fallacieuse « propriété intellectuelle » et le terme de propagande « [protection](#) ».

## **Promotion du matériel libre par le biais des dépôts**

Pour favoriser la liberté des plans de matériel, le moyen le plus efficace est d'édicter des règles dans les dépôts où ils sont publiés. Les opérateurs de ces dépôts doivent mettre la liberté des personnes qui vont utiliser les plans au-dessus des préférences des personnes qui les réalisent. Cela suppose d'exiger la liberté des plans d'objets utiles comme condition préalable à leur publication.

Pour les objets décoratifs, cet argument ne s'applique pas, aussi ne devons-nous pas insister pour qu'ils soient libres. Cependant, nous devons insister pour qu'ils puissent être partagés. Ainsi, un dépôt qui gère à la fois des objets décoratifs et des objets fonctionnels doit avoir une politique appropriée en matière de licence pour chaque catégorie.

Pour les plans d'appareils numériques, je suggère que le dépôt préconise instamment la GNU GPL v3 ou ultérieure, la licence Apache 2.0 ou la CC-0. Pour les modèles 3D d'objets

fonctionnels, le dépôt doit demander à l'auteur du modèle de choisir l'une des quatre licences suivantes : GNU GPL v3 ou ultérieure, Apache 2.0, CC-SA, CC-BY ou CC-0. Pour les modèles d'objets décoratifs, le choix doit être entre la GNU GPL v3 ou ultérieure, la licence Apache 2.0, la CC-0 ou n'importe laquelle des licences Creative Commons.



Le dépôt doit exiger que tous les plans soient publiés en tant que code source, étant entendu que les codes sources en formats secrets utilisables uniquement par des logiciels privateurs de conception ne sont pas vraiment adéquats. Pour les modèles 3D, le [format STL](#) n'est pas le format préféré pour les modifications et par conséquent n'est pas du code source, aussi le dépôt ne doit-il pas l'accepter, sauf peut-être s'il accompagne le vrai code source.

Il n'y a aucune raison de choisir un format unique pour le code source des plans de matériel, mais les formats sources qui ne peuvent pas être reconnus par des logiciels libres

doivent être acceptés avec réticence dans le meilleur des cas.

## Matériel libre et garanties

En général, les auteurs de plans libres pour du matériel n'ont aucune obligation morale d'offrir une garantie à ceux qui mettent en œuvre ces plans. Il s'agit d'un problème différent de celui de la vente de matériel physique, qui doit être accompagné d'une garantie du vendeur ou du fabricant.

## Conclusion

Nous avons déjà des licences appropriées pour rendre libres nos plans de matériel. Ce dont nous avons besoin est de reconnaître que c'est notre devoir en tant que communauté, et d'insister pour que nos plans soient libres lorsque nous fabriquons nous-même des objets.

---

### Note de traduction

1. En anglais, *free software*. Le mot *free* a deux significations : « libre » et « gratuit ». [↑](#)
-