

De la conception à l'objet « imprimé », présentation de la chaîne logicielle opensource.



Ateliers « impression 3D open-source »

par X. HINAULT

www.mon-club-elec.fr



Tous droits réservés – 2014.

Ce document légèrement payant est soumis au droit d'auteur et est réservé à l'usage personnel.

Afin d'encourager la production de supports didactiques de qualité, ce document est légèrement payant.

La licence d'utilisation est attribuée pour un usage personnel uniquement, dans le cercle familial. Mise en ligne et diffusion non autorisées.

Si vous n'êtes pas le détenteur de la licence attribuée pour l'usage de ce document, soyez sympa, merci d'acheter votre exemplaire personnel ici :

http://www.mon-club-elec.fr/pmwiki_mon-club_elec/pmwiki.php?n=MAIN.ATELIERSIMPRESSION3D

Pour tout problème lié à l'utilisation de ce document, veuillez envoyer une copie ici : support@mon-club-elec.fr

Pour obtenir tout autres types de licence d'utilisation (enseignement, commercial, etc...), veuillez contacter l'auteur ici : support@mon-club-elec.fr

Vous avez constaté une erreur ? une coquille ? N'hésitez pas à nous le signaler à cette adresse : support@mon-club-elec.fr

Truc d'utilisation : visualiser ce document en mode diaporama dans le visionneur PDF. Navigation avec les flèches HAUT / BAS ou la souris.

En mode fenêtre, activer le panneau latéral vous facilitera la navigation dans le document. Bonne lecture !

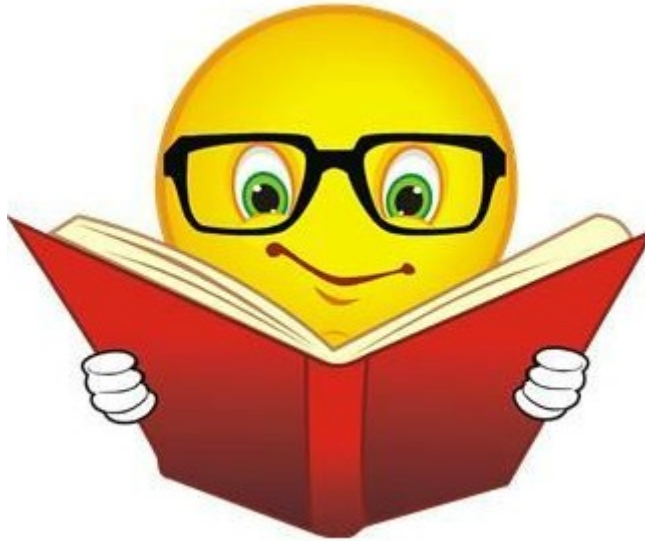
Licence de cet exemplaire accordée à Franck Ourion uniquement pour usage personnel, franck.ourion@univ-lorraine.fr # 7517226

Atelier impression 3D opensource : De la conception à l'objet « imprimé », présentation de la chaîne logicielle opensource. p. 1/19.

1. Intro :

Ce que l'on va faire ici ...

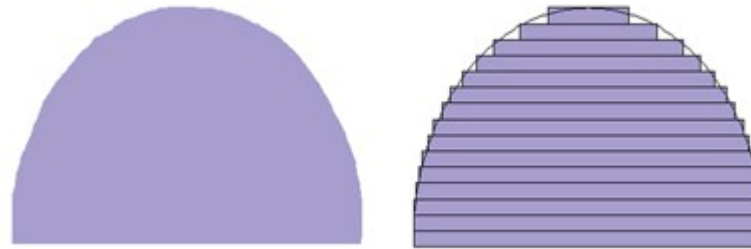
L'objectif ici est de donner une vue d'ensemble des différentes étapes de la conception d'un objet en 3D à l'objet final imprimé, en présentant les logiciels opensource disponibles constituant la chaîne logicielle utilisable en impression 3D opensource.



Prêt ? C'est parti !

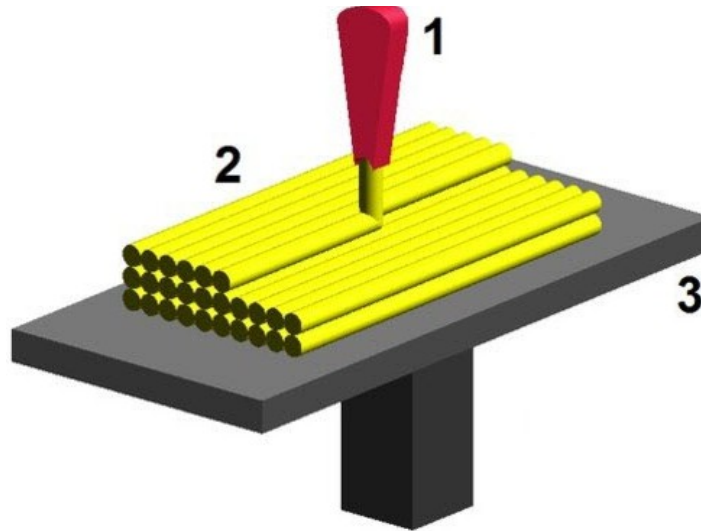
2. L'impression 3D à « plastique fondu » : comment ça marche ?

Le principe général de l'impression 3D par « plastique fondu » consiste à imprimer un objet en imprimant successivement des couches successives dont l'empilement donnera au final la forme de l'objet :



source image : <https://artsonline.uwaterloo.ca/winter2013newsletter/sites/ca.winter2013newsletter/files/diagram.jpg>

Pour réaliser cela, chaque couche va être imprimée à l'aide d'une « buse chauffante » (1) qui va déposer sur un plateau (3) chauffant (pour faciliter l'adhésion) des filaments de plastique fondu juxtaposés (2) de façon à réaliser une couche complète, avant de passer à la couche suivante par élévation de la buse.

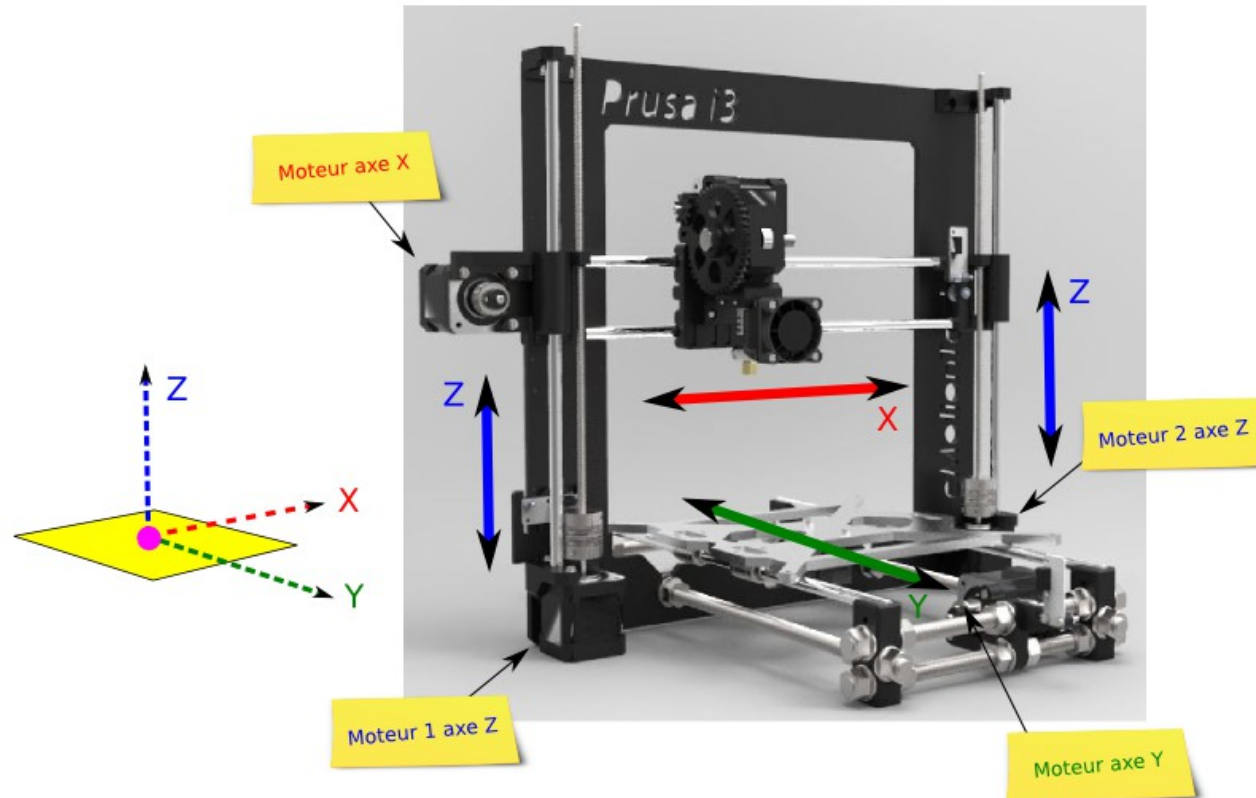


source image : <http://www.qhredcross.org/wp-content/uploads/2013/02/3dprintnter.jpg>

3. Le point de départ : une imprimante 3D opensource opérationnelle

On présuppose que l'on dispose d'une imprimante 3D opensource opérationnelle : on ne s'attardera pas ici sur les aspects fonctionnels et techniques d'une telle imprimante, présentés par ailleurs. Pour faire simple :

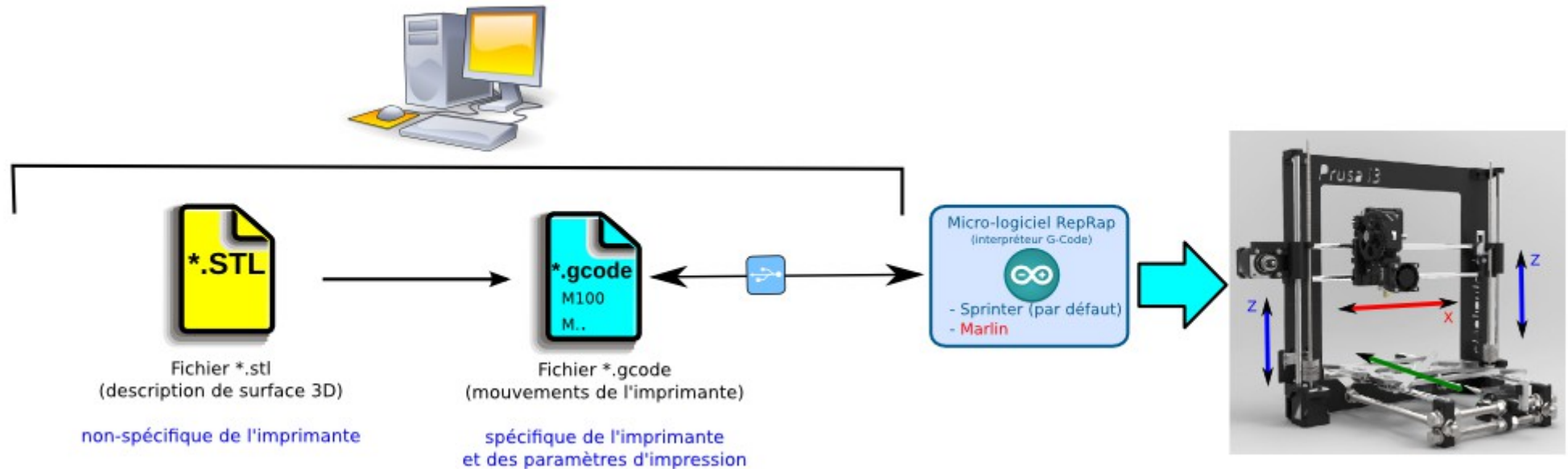
- l'imprimante assure une mobilisation dans les 3 axes X,Y et Z
- ainsi que l'arrivée et la fonte du filament plastique utilisé au niveau de la buse chauffante.



4. Les étapes fondamentales de la chaîne logicielle d'impression 3D opensource

Les étapes fondamentales de l'impression 3D d'un objet sont les suivantes :

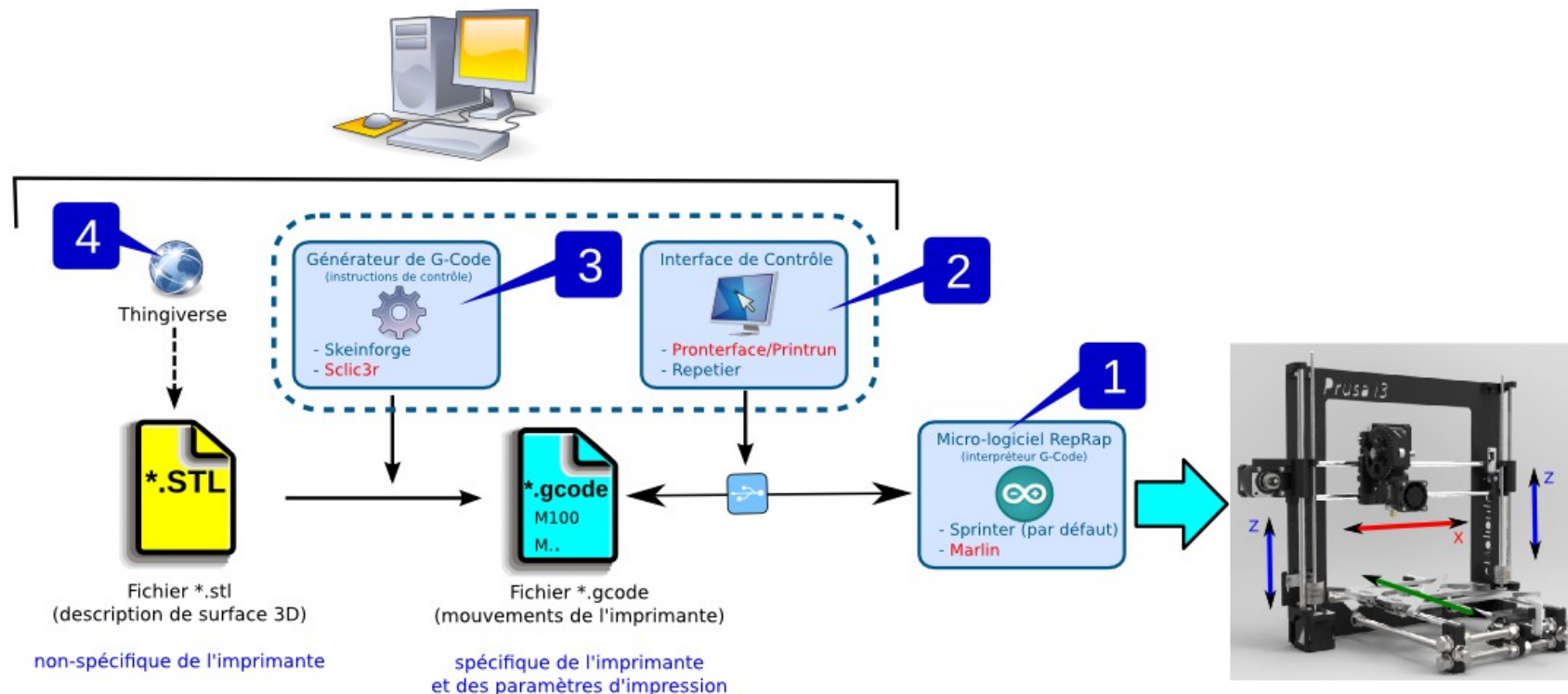
- on part d'un **fichier au format dit *.STL** qui contient la description surfacique 3D d'un objet. **Le fichier *.STL est une description de l'objet non-spécifique de l'imprimante utilisée.** C'est ce type de fichier qui sera partagé en ligne si on souhaite mettre une pièce à disposition de d'autres utilisateurs. En pratique, on pourra :
 - soit utiliser un fichier existant (dépôt de pièces en ligne)
 - soit générer ce fichier à partir d'un logiciel de conception 3D comme nous allons le voir
- le fichier *.STL va ensuite être converti en un **fichier de *.gcode** qui va contenir l'ensemble des instructions permettant les mouvements voulus de l'imprimante 3D pour imprimer la pièce. **Le fichier *.gcode est spécifique :**
 - des **paramètres d'impression choisis** (épaisseur de couche, taux de remplissage, vitesse d'impression, etc...)
 - et des caractéristiques **de l'imprimante utilisée.**
- Les instructions du fichier *.gcode seront ensuite envoyées à l'imprimante qui va exécuter les mouvements voulus grâce à son « micro-logiciel » qui est un **décodeur de G-Code.**



5. Vue d'ensemble de la chaîne logicielle d'impression 3D opensource « courte » :

Une fois comprises les étapes fondamentales, il devient possible de décrire simplement la chaîne logicielle minimale nécessaire pour imprimer un objet 3D, chaîne qui va associer, en reprenant la chaîne à l'envers (le plus simple...) :

- un **micro-logiciel** (1) implémenté dans l'imprimante 3D elle-même qui va décoder le G-Code.
- un **logiciel d'interface de contrôle de l'imprimante 3D** (2) permettant de communiquer avec l'imprimante et de lui envoyer les instructions *.gcode générée par le logiciel slicer. En pratique, il s'agit d'une interface graphique écrite en Python qui permet de contrôler manuellement ou automatiquement tous les éléments fonctionnels de l'imprimante via le port USB.
- un **logiciel de conversion d'un fichier *.STL en fichier *.gcode** (3), logiciel appelé également slicer. Ce logiciel est très puissant et va calculer tous les mouvements nécessaires de l'imprimante pour obtenir l'objet 3D. Il permet de moduler de nombreux paramètres de réglages, notamment l'épaisseur de la couche, le taux de remplissage voulu, la géométrie de remplissage à utiliser, la vitesse des mouvements, etc...
- on devra disposer au minimum d'un **fichier *.STL tout prêt** (4), obtenu dans un dépôt en ligne de pièces 3D « prêtes à imprimer » : typiquement sur le site <http://www.thingiverse.com/>

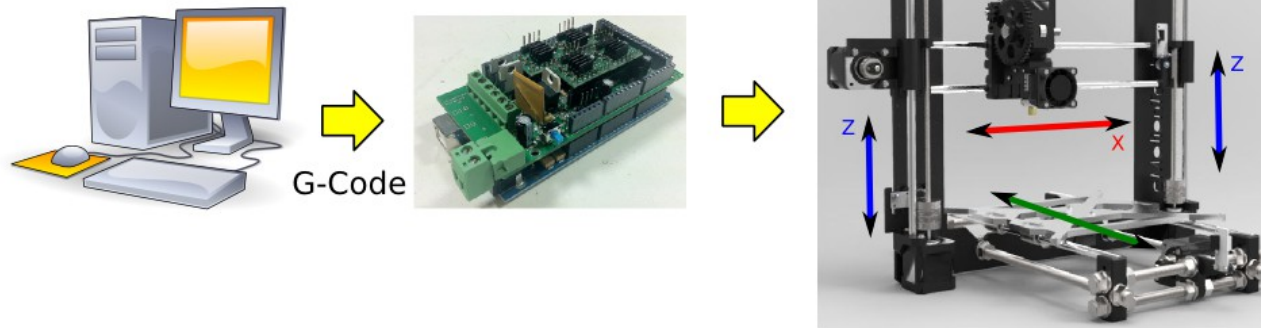


6. Impression 3D opensource : 1. le micro-logiciel (ou Firmware) de contrôle, un décodeur de G-Code !

Pour comprendre

Le principe général du contrôle d'une imprimante 3D va passer par **une série d'instructions envoyée par le port série par le poste de contrôle** (ou bien stocké sur carte SD). Les instructions reçues par l'imprimante via le port USB vont permettre de **définir les mouvements des axes et de l'extrudeur**. Le code d'instruction utilisé s'appelle le **G-Code : c'est un langage de programmation des machines numériques**, utilisé dans l'industrie notamment.

Assez logiquement, on va donc programmer la carte électronique de l'imprimante avec un programme qui va assurer le rôle de « **décodeur de G-Code** », transformant l'imprimante 3D en un automate « intelligent » capable d'exécuter les instructions reçues par le port série.



Exemple d'instructions G-Code

Les instructions G-Code sont constituée d'une lettre suivie d'un numéro, chaque code ayant une signification précise(Plus de détails ici : <https://github.com/ErikZalm/Marlin>) :

- M105 : lire la température courante
- M400 : stopper tous les mouvements

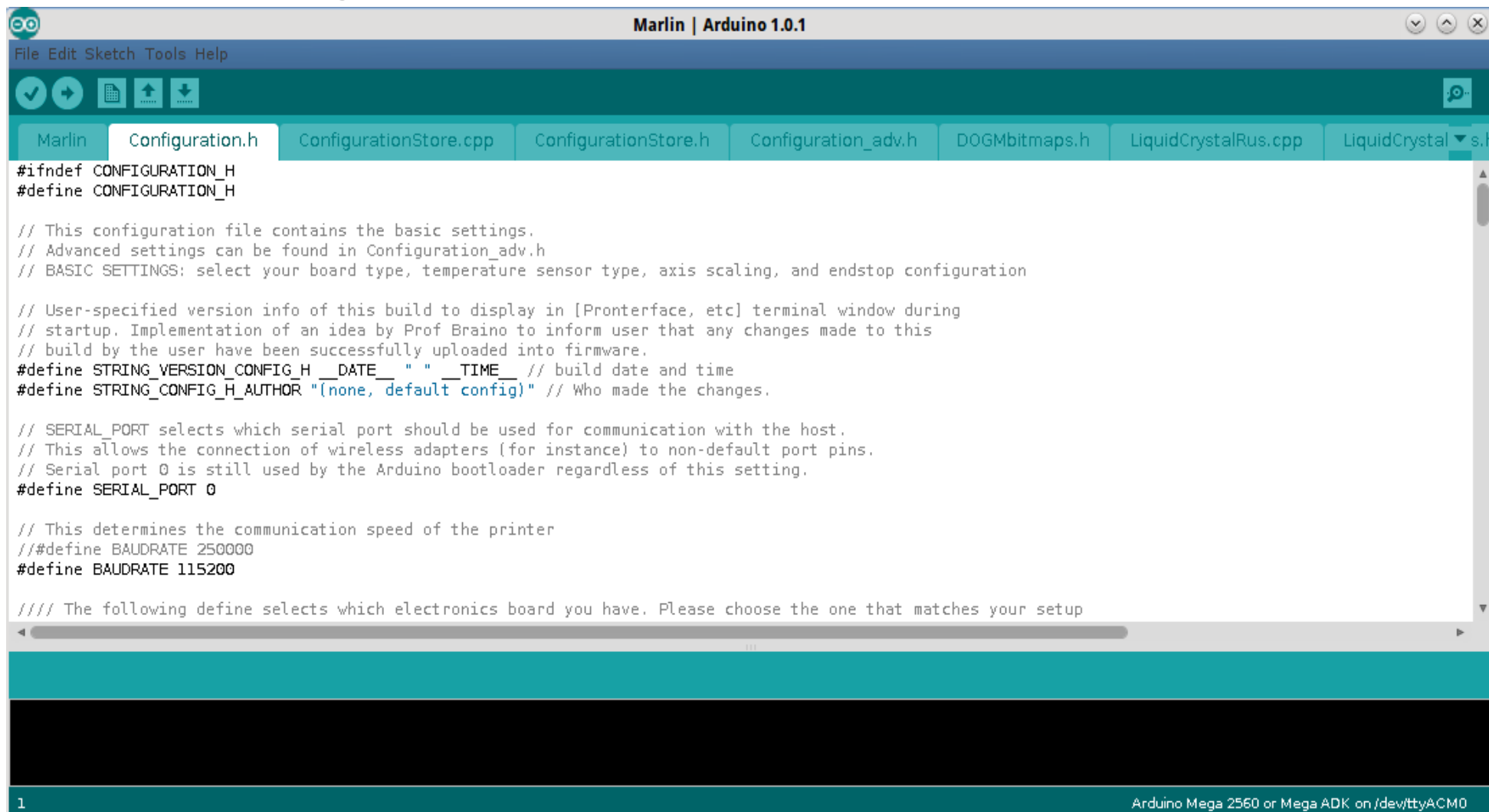
Les micro-logiciels disponibles

Comme pour toute solution de logiciel libre, il existe ici plusieurs solutions de micro-logiciel de contrôle de l'imprimante 3D. On pourra trouver un panorama ici : http://reprap.org/wiki/List_of_Firmware

En pratique, les plus utilisés sont :

- Marlin : <https://github.com/ErikZalm/Marlin>
- Repetier, Sprinter...

7. Exemple du micro-logiciel Marlin ouvert dans l'IDE Arduino



```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H

// This configuration file contains the basic settings.
// Advanced settings can be found in Configuration_adv.h
// BASIC SETTINGS: select your board type, temperature sensor type, axis scaling, and endstop configuration

// User-specified version info of this build to display in [Pronterface, etc] terminal window during
// startup. Implementation of an idea by Prof Braino to inform user that any changes made to this
// build by the user have been successfully uploaded into firmware.
#define STRING_VERSION_CONFIG_H __DATE__ " " __TIME__ // build date and time
#define STRING_CONFIG_H_AUTHOR "(none, default config)" // Who made the changes.

// SERIAL_PORT selects which serial port should be used for communication with the host.
// This allows the connection of wireless adapters (for instance) to non-default port pins.
// Serial port 0 is still used by the Arduino bootloader regardless of this setting.
#define SERIAL_PORT 0

// This determines the communication speed of the printer
// #define BAUDRATE 250000
#define BAUDRATE 115200

//// The following define selects which electronics board you have. Please choose the one that matches your setup
```

Principe d'installation

Le logiciel est en fait une « super-programme Arduino » et il va donc être possible de le programmer simplement depuis l'IDE Arduino vers la carte de l'imprimante, tout simplement. Une fois programmé dans la carte de l'imprimante, on peut potentiellement interagir avec l'imprimante à partir du Terminal série si on veut : comme pour un code Arduino classique !

8. Impression 3D opensource : 2. le logiciel d'interface graphique de contrôle de l'imprimante 3D.

Pour comprendre

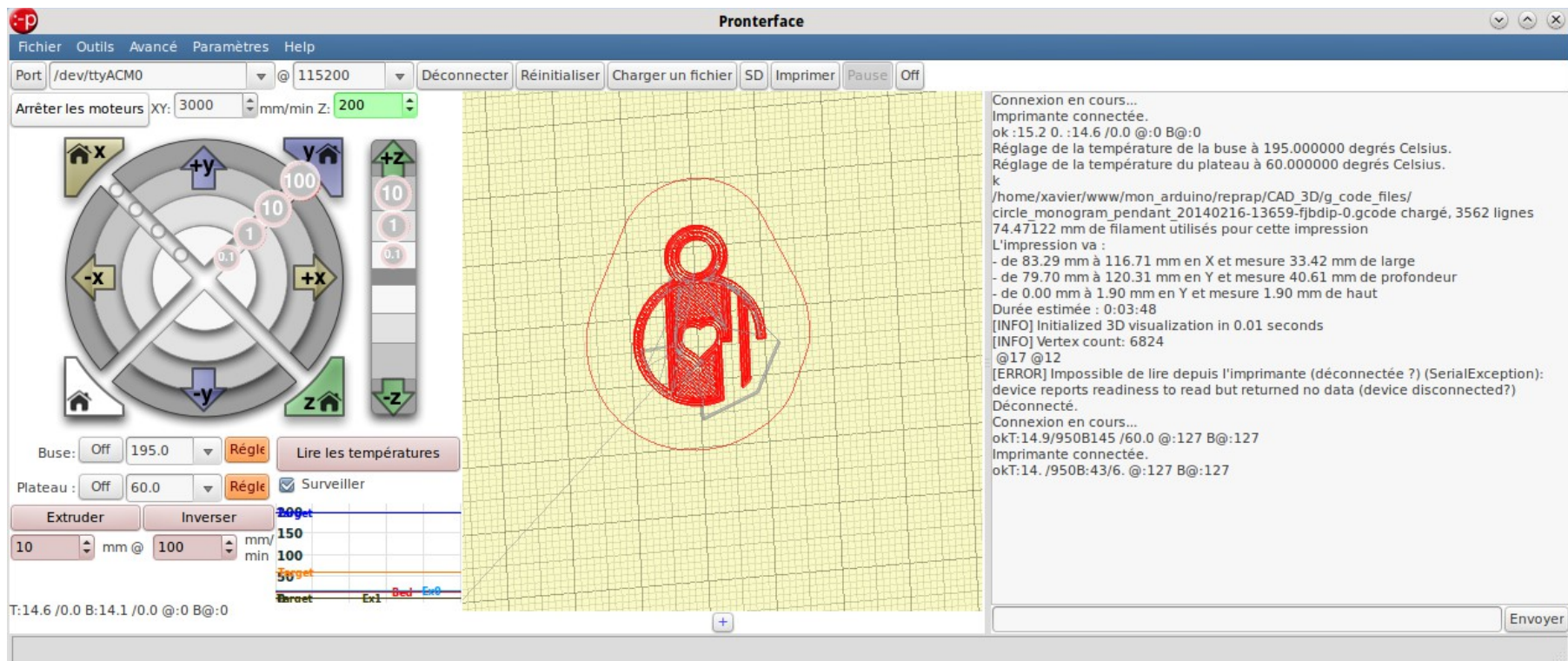
L'interface graphique permet le contrôle complet de l'imprimante :

- contrôler la **communication série** avec l'imprimante via le port USB
- contrôler et paramétrer la **chauffe de la buse, du plateau, et visualiser les températures**
- contrôler l'**extrusion** (arrivée du fil plastique)
- contrôler les **mouvements des différentes axes X, Y et Z**
- **visualiser la pièce** à imprimer en 2D ou 3D
- bien sûr **lancer une impression** à partir d'un fichier G-Code...
- et même **visualiser graphiquement la progression** de l'impression 3D !!

Noter que **toute la communication entre l'interface graphique et le PC sera visualisée dans une fenêtre**, équivalent du terminal Série Arduino.

Exemple : l'interface Pronterface

Une des interfaces les plus utilisées est l'interface **pronterface** :



9. Impression 3D opensource : 3. le logiciel générateur de G-Code ou « slicer »

Pour comprendre

Le format de fichier utilisé pour la description d'une pièce 3D en vue d'une impression 3D est un format dit *.stl. Ce type de fichier correspond à une description surfacique de la pièce 3D. Pour plus de détails, voir : http://en.wikipedia.org/wiki/STL_%28file_format%29

Une fois que l'on dispose du fichier *.stl, on va le convertir en G-code, seul « langage » que l'imprimante 3D est capable de comprendre. Le G-Code va décrire tous les mouvements que l'imprimante devra effectuer.

Pour réaliser cette conversion, on va donc utiliser un logiciel appelé « Générateur de G-Code »

Le logiciel « générateur de G-Code » va calculer toutes les trajectoires et mouvements que devra effectuer l'imprimante 3D en se basant :

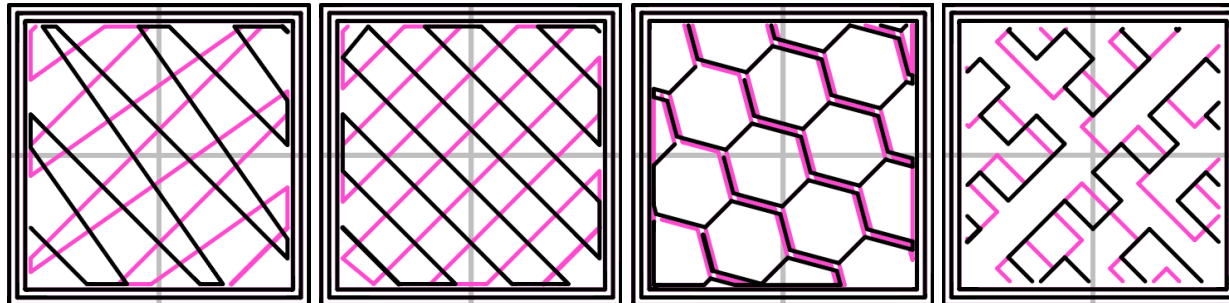
- sur les caractéristiques techniques de l'imprimante d'une part : diamètre de la buse, caractéristiques mécaniques des axes, taille du plateau, etc..
- sur les paramètres souhaités pour l'impression d'autre part : taux de remplissage de la pièce, géométrie de remplissage, vitesse de déplacement, épaisseur de couche, nombre de bords, etc..

L'exemple du logiciel Slic3R

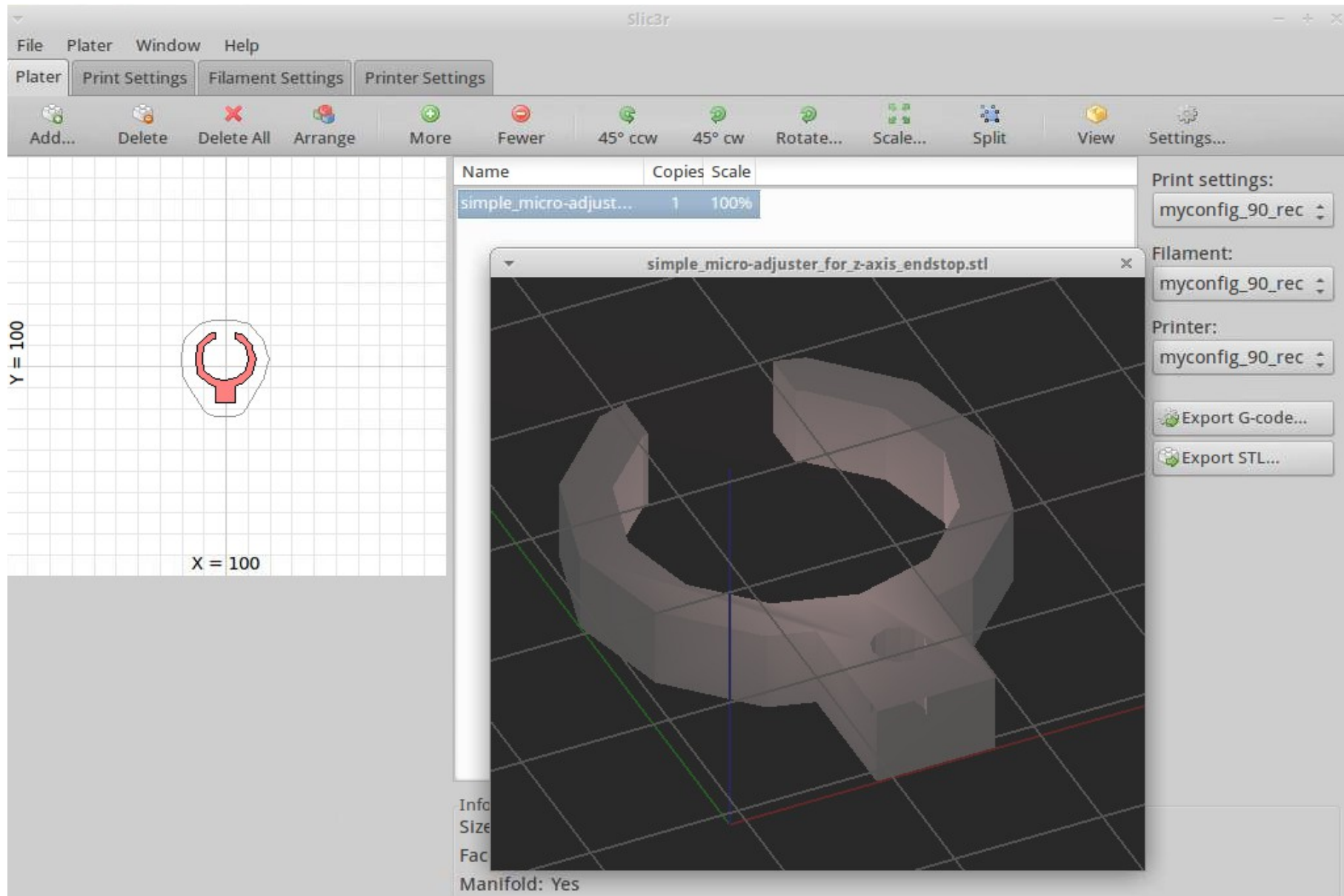
L'un des logiciels « générateur de G-Code » les plus utilisés est Slic3R : il s'agit d'un logiciel très avancé qui reste simple d'utilisation pour un débutant. Il en existe beaucoup d'autres. Site : <http://slic3r.org/>

Ce logiciel permet de :

- charger un ou plusieurs fichiers *.stl
- de pré-visualiser les pièces en 3D
- de placer/modifier la/les pièces sur le plateau de l'imprimante
- de définir de très nombreux paramètres d'impression, notamment :
 - l'épaisseur de couche
 - le nombre de périmètres (bords)
 - le taux de remplissage de la pièce
 - et même la géométrie de remplissage !



10. Slic3R : l'interface

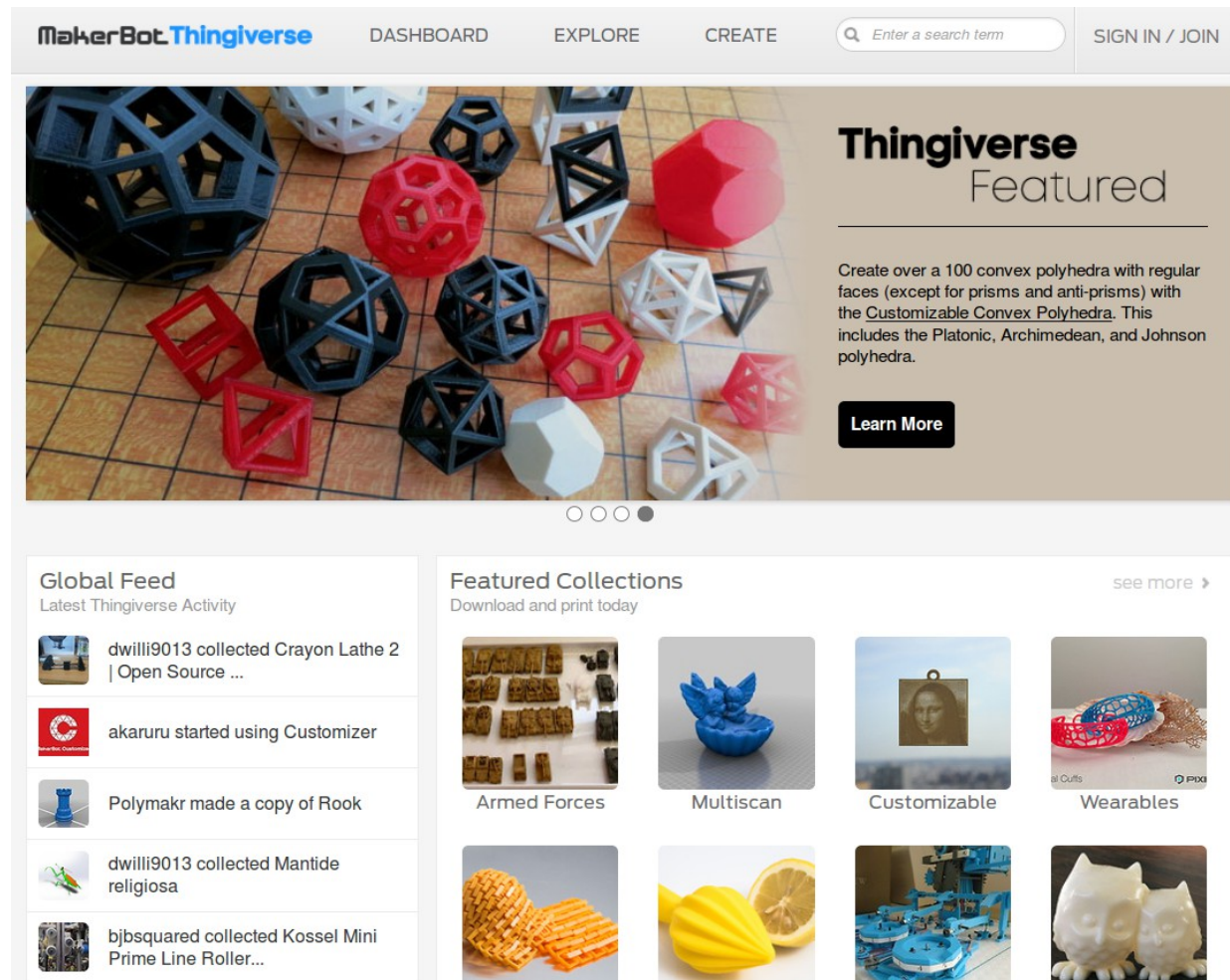


11. Impression 3D opensource : 4. dépôt en ligne de pièces « prêtes à l'emploi »

Une des grandes forces de l'impression 3D opensource est de permettre la circulation de l'information nécessaire pour réaliser une pièce mécanique plutôt que de faire circuler les pièces elles-mêmes. Il s'agit là d'un renversement à la fois social, écologique, économique, etc... dont la portée peut potentiellement devenir un « changement de paradigme » sociétal à moyen terme. Mais c'est un autre sujet...

Concrètement, il existe déjà des dépôts en ligne offrant des milliers de fichiers *.STL « prêt à l'emploi » et surtout libres d'usage. Le dépôt le plus en vogue est notamment Thingiverse : <http://www.thingiverse.com/>

Pour faire simple, il n'y a qu'à « faire son marché » : des bijoux, aux pièces de robotique en passant par le modélisme ou des pièces de bricolage, c'est une vraie caverne d'ali-baba.... où il est vite possible de se noyer si l'on n'a pas défini son besoin au préalable d'ailleurs !



12. Vue d'ensemble de la chaîne logicielle d'impression 3D opensource « longue »

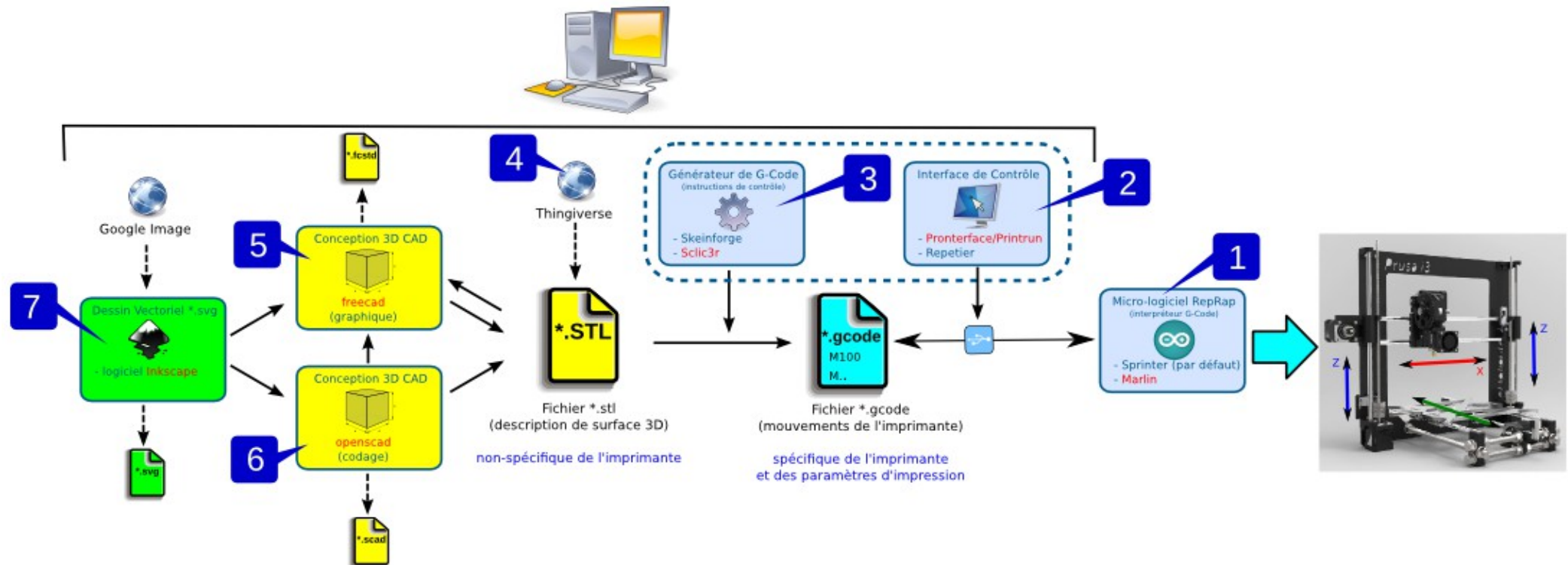
La chaîne logicielle impression 3D opensource « courte » consiste comme nous l'avons vu à partir d'un fichier *.STL prêt à l'emploi. C'est très pratique en phase de découverte, mais très vite limité dès lors que l'on souhaite concevoir et réaliser ses propres pièces. Heureusement, il est facile de compléter la chaîne logicielle minimale par des logiciels opensource de conception 3D, ce qui nous donne la chaîne logicielle « longue » :

Une fois encore, le monde du logiciel libre et de l'opensource fournit un ensemble de logiciels de conception 3D et 2D performants notamment :

- Freecad, un logiciel de conception 3D en mode graphique de niveau industriel,
- Openscad, un logiciel de conception 3D paramétrique par codage, très pratique et performant,
- Inkscape, un logiciel de dessin vectoriel qui permettra notamment d'utiliser n'importe quel fichier 2D vectoriel existant pour créer une pièce 3D

Ces différents outils vont ensuite pouvoir se combiner entre-eux permettant de créer des stratégies de conception simples : créer du texte en 2D avant de l'extruder en 3D, créer un profil d'engrenage 2D avant d'en réaliser une roue dentée, créer des pièces paramétriques, etc...

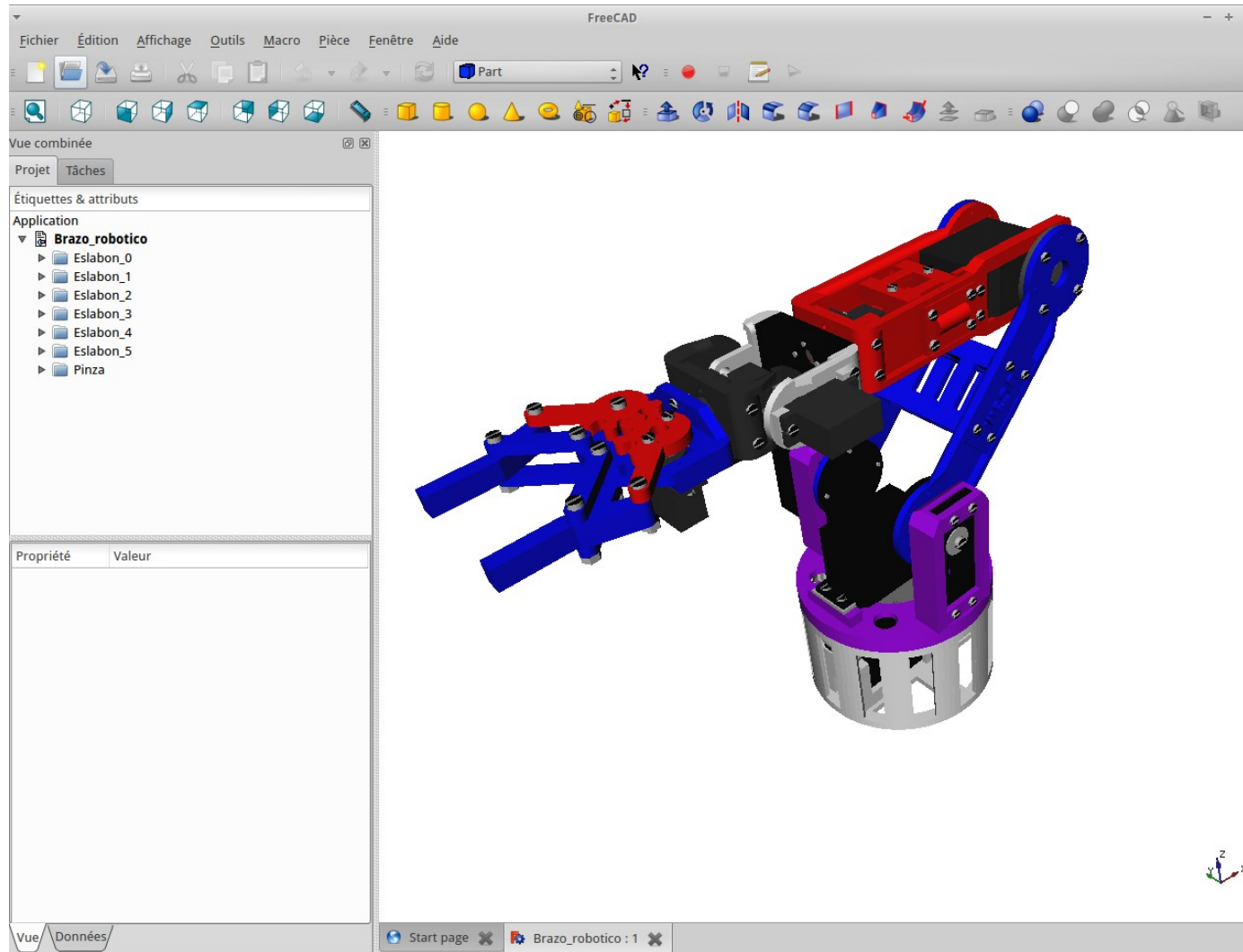
Tous ces logiciels vont évidemment permettre de générer un fichier *.STL de la pièce qui pourra ensuite être converti en *.gcode puis être imprimé !



13. Impression 3D opensource : 5. un logiciel de conception 3D en mode graphique, Freecad

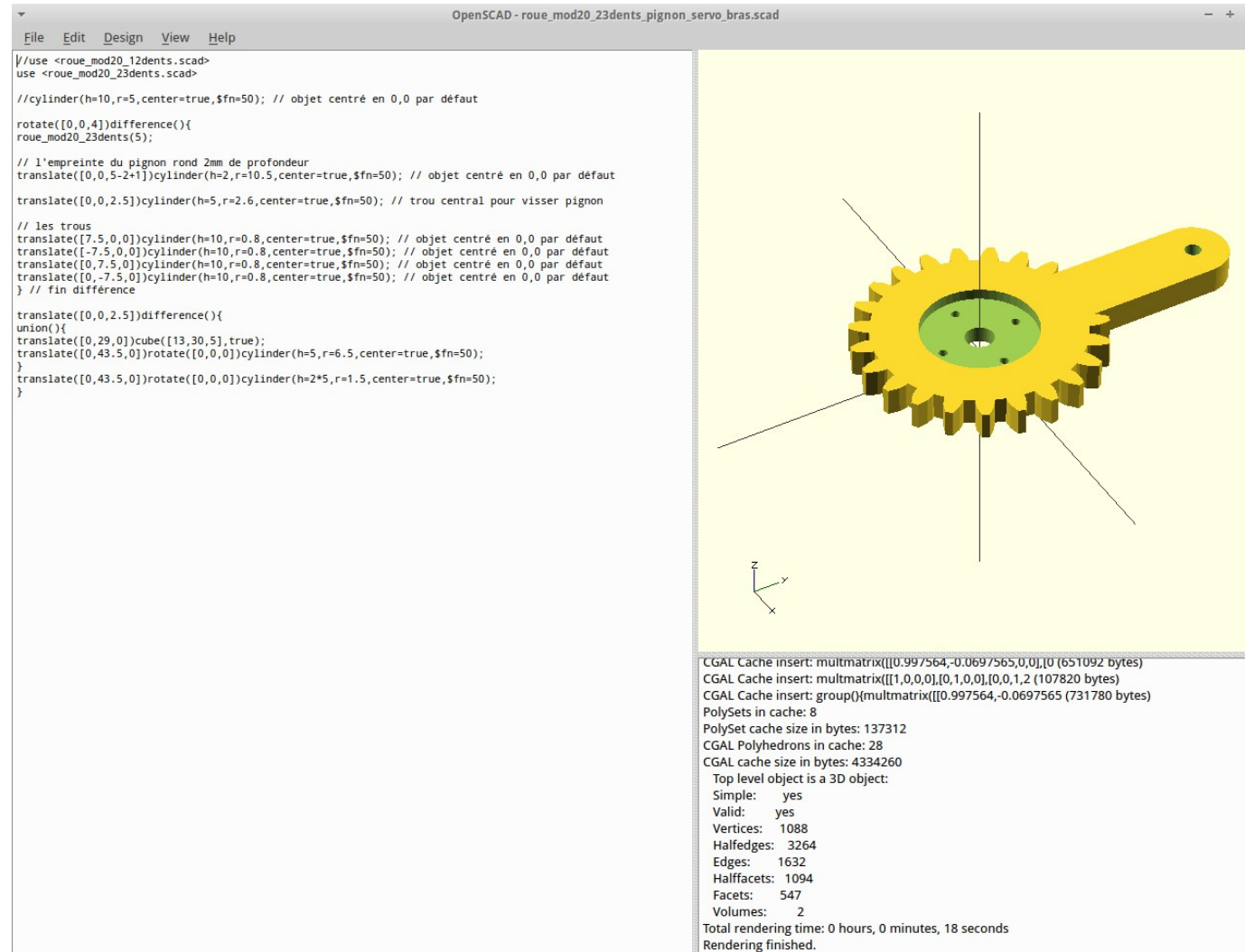
Pour créer ses pièces de zéro (ou « from scratch »), on va utiliser un logiciel de « Conception Assistée par Ordinateur en 3 dimensions » ou logiciel CAO 3D. Il existe plusieurs possibilités.

Freecad (<http://www.freecadweb.org/>) est celui que je conseille : il s'agit d'un logiciel de conception 3D de niveau professionnel, mais assez simple à prendre en main, capable d'exporter/importer toutes sortes de fichiers. Logiciel basé sur opencascade, un moteur 3D puissant, de niveau professionnel : voir <http://www.opencascade.org/> (utilisé notamment par des industriels, notamment ALCA TEL SPACE, BMW, RINA, CEA, EDF, EADS, MITUTOYO, ARCELOR ..)



14. Impression 3D opensource : 6. un logiciel de conception 3D paramétrique par codage, Openscad

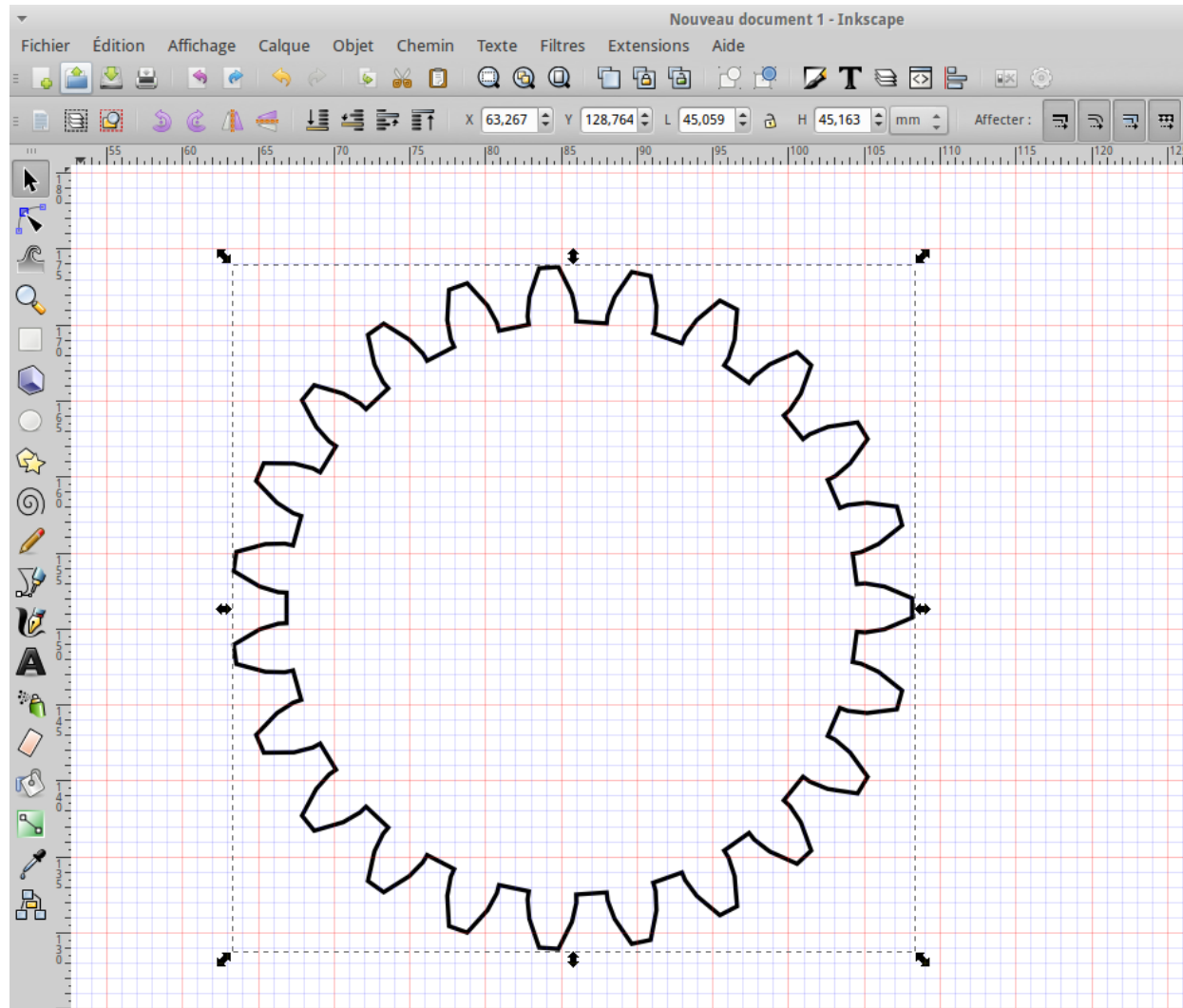
Openscad est un logiciel de CAD 3D contrôlé par codage, ayant une syntaxe proche du C. C'est un logiciel openSource, compatible Mac, Linux, Windows. L'intérêt de ce logiciel est de pouvoir paramétrer facilement des pièces, notamment répétitives ou d'intégrer des éléments d'une pièce dans une autre par simple copier/coller. Liens utiles : <http://www.openscad.org/>



15. Impression 3D opensource : 7. un logiciel de conception 2D vectoriel polyvalent, Inkscape

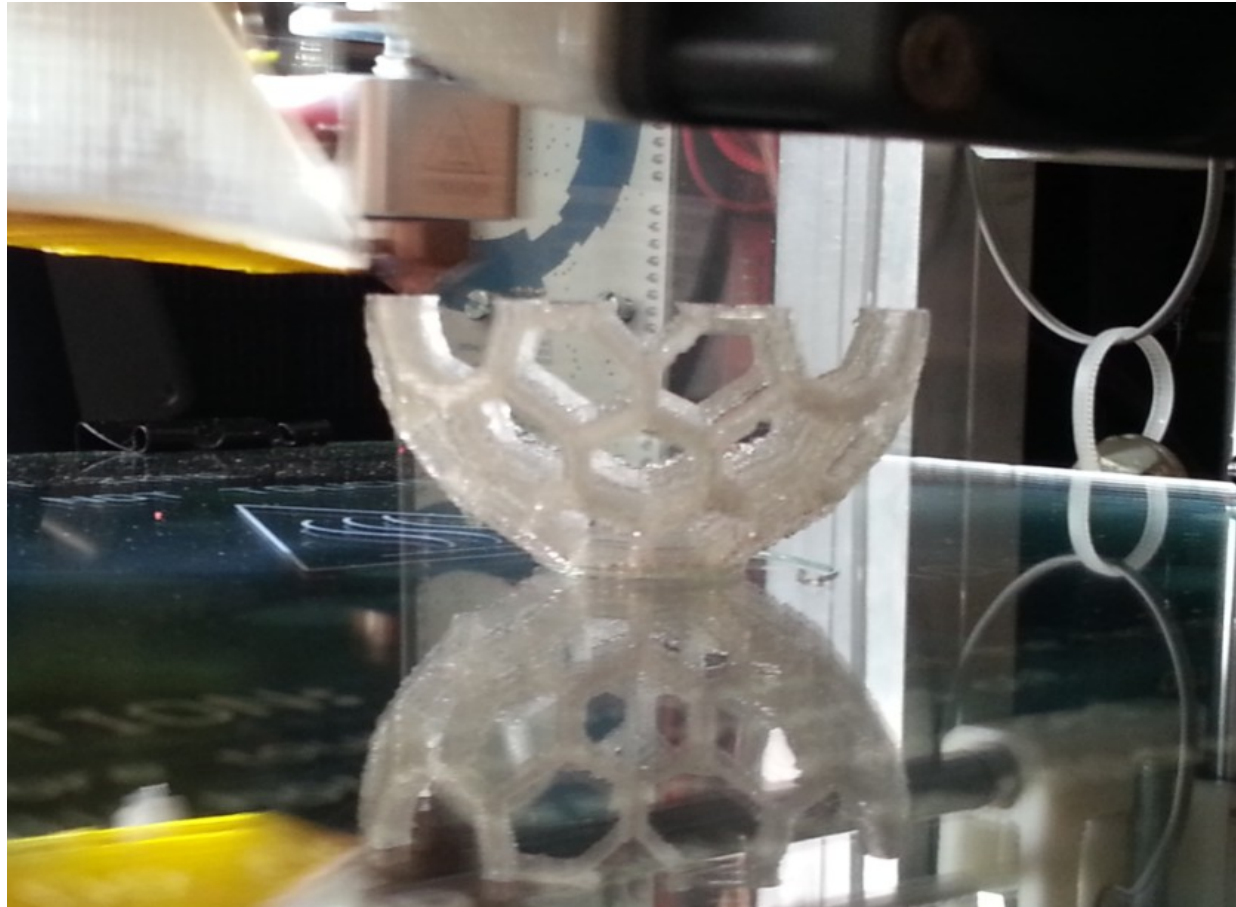
Inkscape est un logiciel opensource de dessin vectoriel de qualité polyvalent. Son intérêt en conception de pièce 3D est multiple :

- permet l'export de fichier au format *.svg utilisables avec FreeCAD ou exportable vers Openscad
- dispose d'utilitaires de dessin automatisés, notamment pour la création d'engrenage, etc..
- permet d'exporter des polices de caractères au format *.SVG et donc d'en faire des pièces 3D,
- etc...



16. Conclusion

Le monde de l'opensource offre tout l'arsenal logiciel nécessaire, de niveau (quasi-)professionnel, pour contrôler, créer et imprimer des pièces 3D à l'aide d'imprimantes 3D elle-mêmes open-hardware. Un simple comparatif des coûts permettra de se rendre compte de tout l'intérêt de cette solution, l'économie étant de -75 % environ sur les machines d'impression et de -100 % sur les licences logicielles... tout en obtenant une qualité comparable et en contrôlant totalement l'ensemble des aspects techniques utilisés. Pourquoi s'en priver ?



Une impression en cours sur une imprimante opensource et open-hardware !

Table des matières

Structure fonctionnelle d'une imprimante 3D open-source

Intro : |

L'impression 3D à « plastique fondu » : comment ça marche ? |

Le point de départ : une imprimante 3D opensource opérationnelle |

Les étapes fondamentales de la chaîne logicielle d'impression 3D opensource |

Vue d'ensemble de la chaîne logicielle d'impression 3D opensource « courte » : |

Impression 3D opensource : 1. le micro-logiciel (ou Firmware) de contrôle, un décodeur de G-Code ! |

Exemple du micro-logiciel Marlin ouvert dans l'IDE Arduino |

Impression 3D opensource : 2. le logiciel d'interface graphique de contrôle de l'imprimante 3D. |

Impression 3D opensource : 3. le logiciel générateur de G-Code ou « slicer » |

Slic3R : l'interface |

Impression 3D opensource : 4. dépôt en ligne de pièces « prêtes à l'emploi » |

Vue d'ensemble de la chaîne logicielle d'impression 3D opensource « longue » |

Impression 3D opensource : 5. un logiciel de conception 3D en mode graphique, FreeCAD |

Impression 3D opensource : 6. un logiciel de conception 3D paramétrique par codage, OpenSCAD |

Impression 3D opensource : 7. un logiciel de conception 2D vectoriel polyvalent, Inkscape |

Conclusion |