

Structure fonctionnelle d'une imprimante 3D open-source



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 !

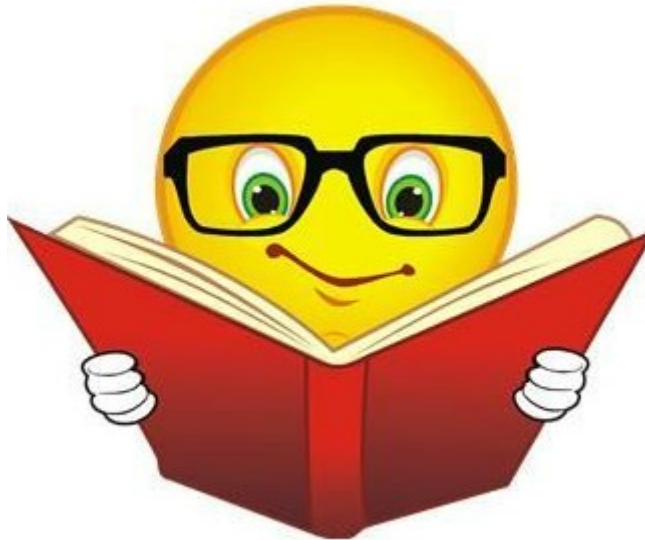
Licence de cet exemplaire accordée à Franck Ourion uniquement pour usage personnel, franck.ourion@univ-lorraine.fr # 7517226

Atelier Impression 3D open-source : Structure fonctionnelle d'une imprimante 3D open-source. p. 1/22

1. Intro

L'objectif ici est :

- de découvrir le principe fonctionnel d'une imprimante 3D
- de comprendre le rôle des différents éléments mécanique, électriques et électroniques

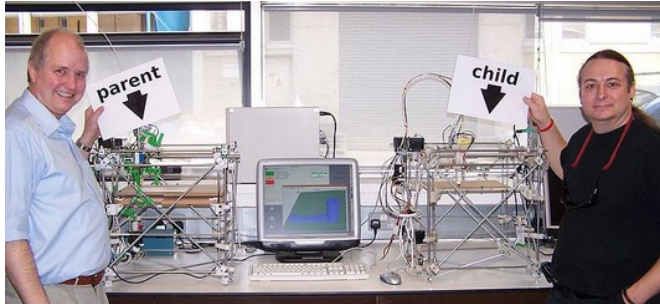


Prêt ? C'est parti !

2. Les imprimantes 3D open-source : petite histoire

Au commencement

Les imprimantes 3D opensource, dont les « reprop » sont le chef de file, ont vu le jour entre 2005 et 2008. A l'origine de ces machines opensource, un maître de conférence en ingénierie mécanique à l'Université de Bath au Royaume-Uni, **Adrian Bowyer**. Dès le départ, A. Bowyer a conceptualiser l'aspect « social » de cette machine qui avait pour but de permettre à chacun de produire les pièces dont il a besoin : le projet a d'emblée été mis sous licence libre permettant la création de machines dérivées et en obligeant à les laisser libres également !



très haute qualité, etc...

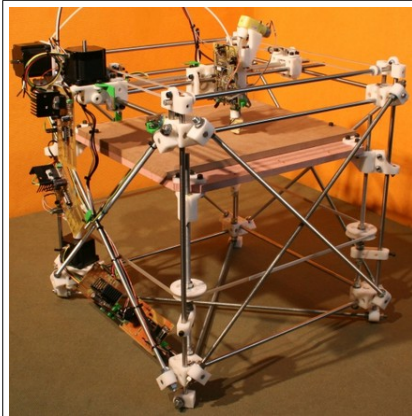
Le concept

RepRap est le diminutif de contraction de l'anglais **Replication Rapid prototyper**, pouvant se traduire par machine **de répliation rapide**

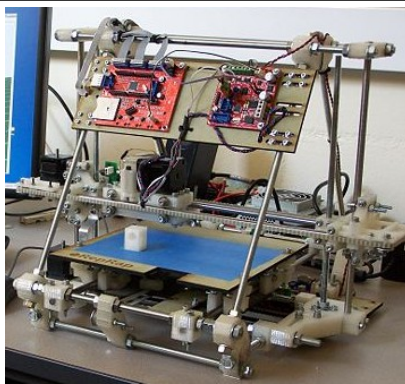
Le concept initial est le suivant : le projet RepRap a pour but de permettre la création d'une machine capable de se répliquer. De 2005 à 2008, le projet va mûrir et la première « reprop » opérationnelle est montée en février 2008 : la formidable aventure de l'impression 3D opensource pouvait commencer !

Le projet étant libre et opensource tant au niveau matériel que logiciel, c'est toute une communauté qui s'est greffée sur le projet, produisant des machines dérivées améliorées, une chaîne logicielle de

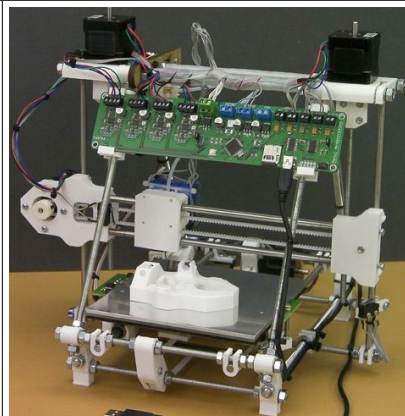
Les générations de machines



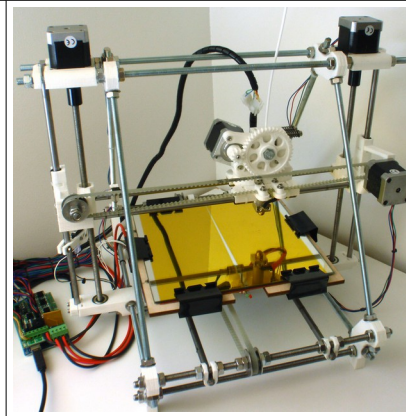
2008 : La RepRap 1.0, la Darwin, la première



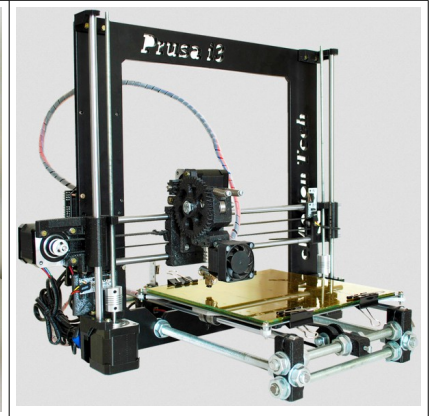
2009 : La RepRap 2.0, la Mendel



2010 : Une mini-Mendel, la Huxley



2010 : La Prusa Mendel, une évolution majeure de la Mendel



2013 : La Prusa Mendel i3 (3ème itération), la maturité !

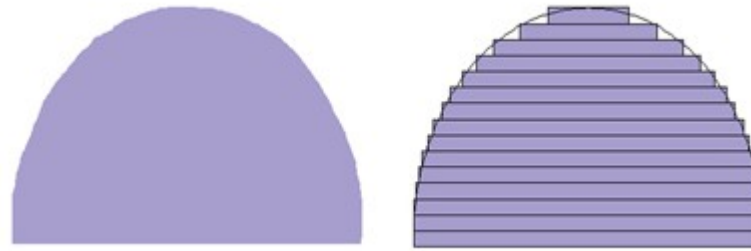
L'arbre généalogique des RepRap est ici : http://upload.wikimedia.org/wikipedia/commons/5/56/Reprap_Family_Tree.png

Et aussi...

A côté des machines opensources, des machines « commerciales », ayant un modèle semi-ouvert ou carrément fermé : Makerbot Replicator, la Up !, etc...

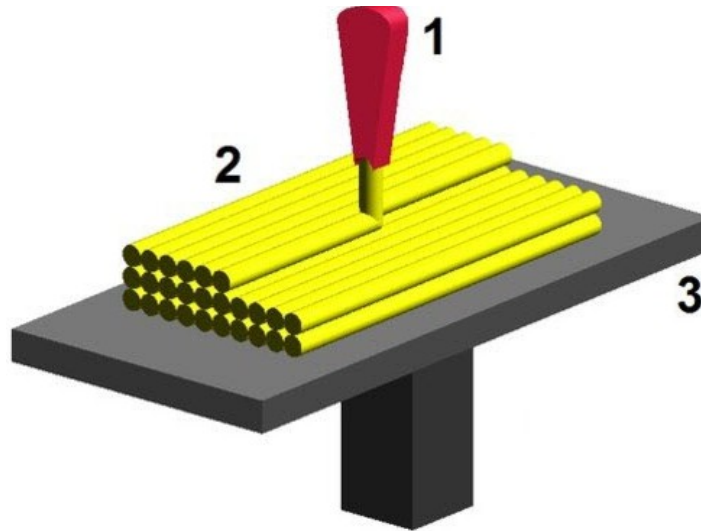
3. L'impression 3D à « plastique fondu » (ou FDM) : comment ça marche ?

Le principe général de l'impression 3D par « plastique fondu » (ou FDM « Fused deposition modeling », ou FFF « fused filament fabrication » ou PJP « Plastic Jet Printing ») 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

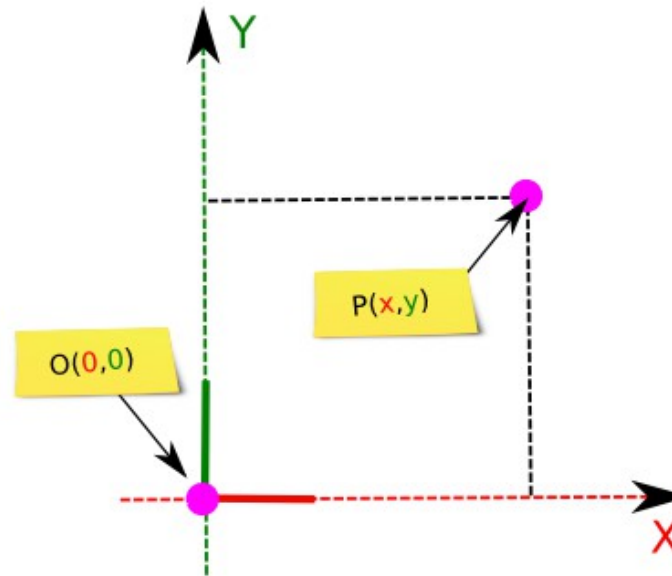
Pour réaliser cela, chaque couche va être imprimée à l'aide d'une « buse chauffante » (1) qui va déposer sur un plateau (3) chauffant (pour faciliter l'adhésion) des filaments de plastique fondu juxtaposés (2) de façon à réaliser une couche complète, avant de passer à la couche suivante par élévation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



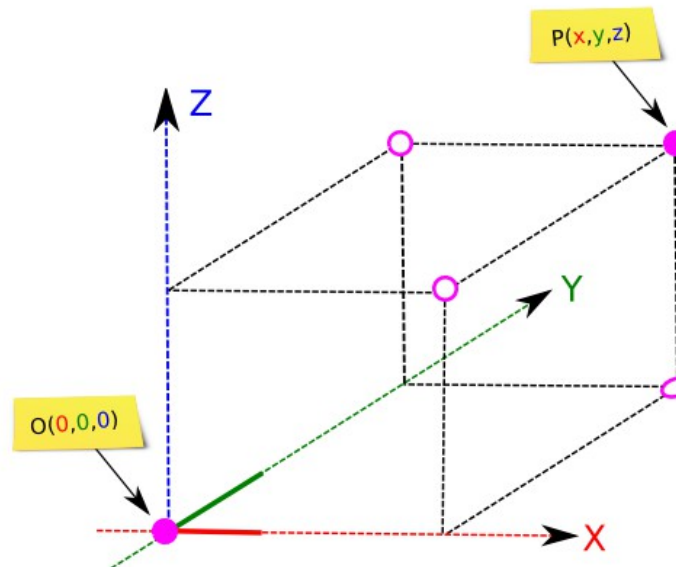
source image : <http://www.ghredcross.org/wp-content/uploads/2013/02/3dprinter.jpg>

4. Rappel : Représentation dans le plan et dans l'espace

Pour positionner un objet dans un plan, on utilise 2 axes X et Y et on repère un point avec les coordonnées (x,y)



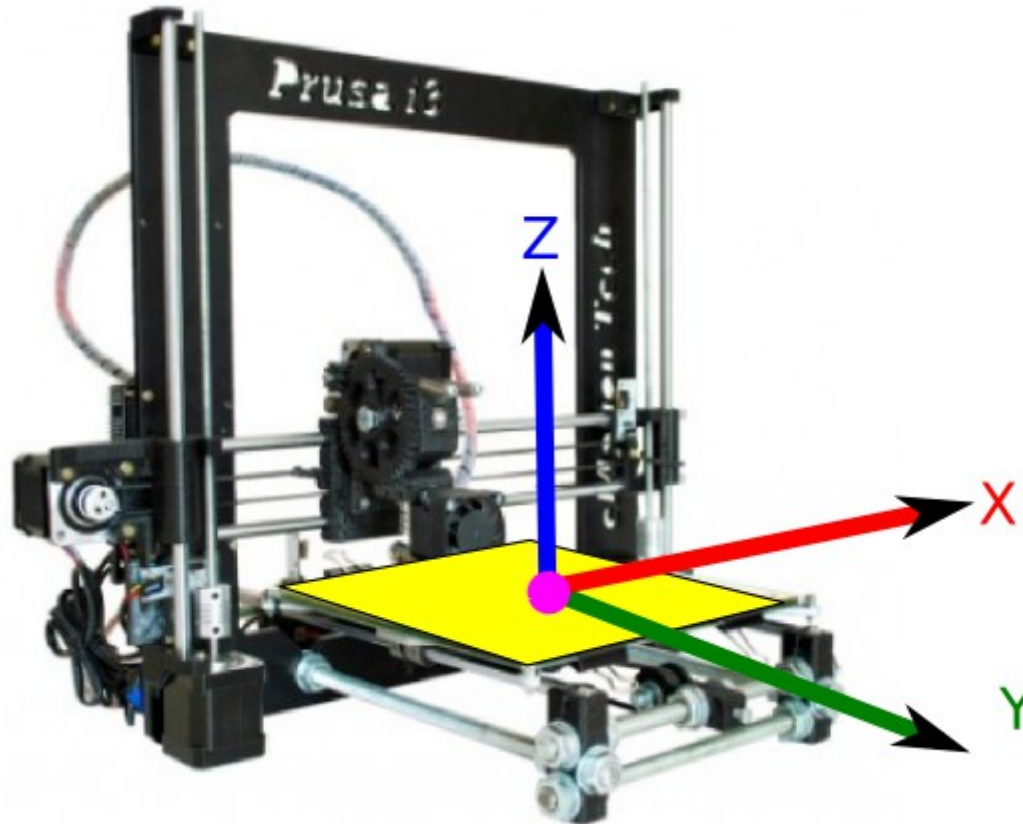
Pour positionner un objet dans l'espace, on utilise 3 axes X,Y et Z et on repère un point avec les coordonnées (x,y,z) :



5. Représentation dans l'espace : application à l'imprimante 3D

Dans le cas de l'imprimante 3D :

- le plan (X,Y) est représenté par le plateau : c'est le plan dans lequel est dessiné chaque couche de l'objet
- l'axe Z est représenté par la hauteur de la buse, autrement c'est l'axe d'empilement des couches imprimées



L'imprimante 3D va devoir être capable de réaliser un positionnement :

- pour dessiner chaque couche : dans l'axe X et dans l'axe Y
- entre chaque couche : dans l'axe Z

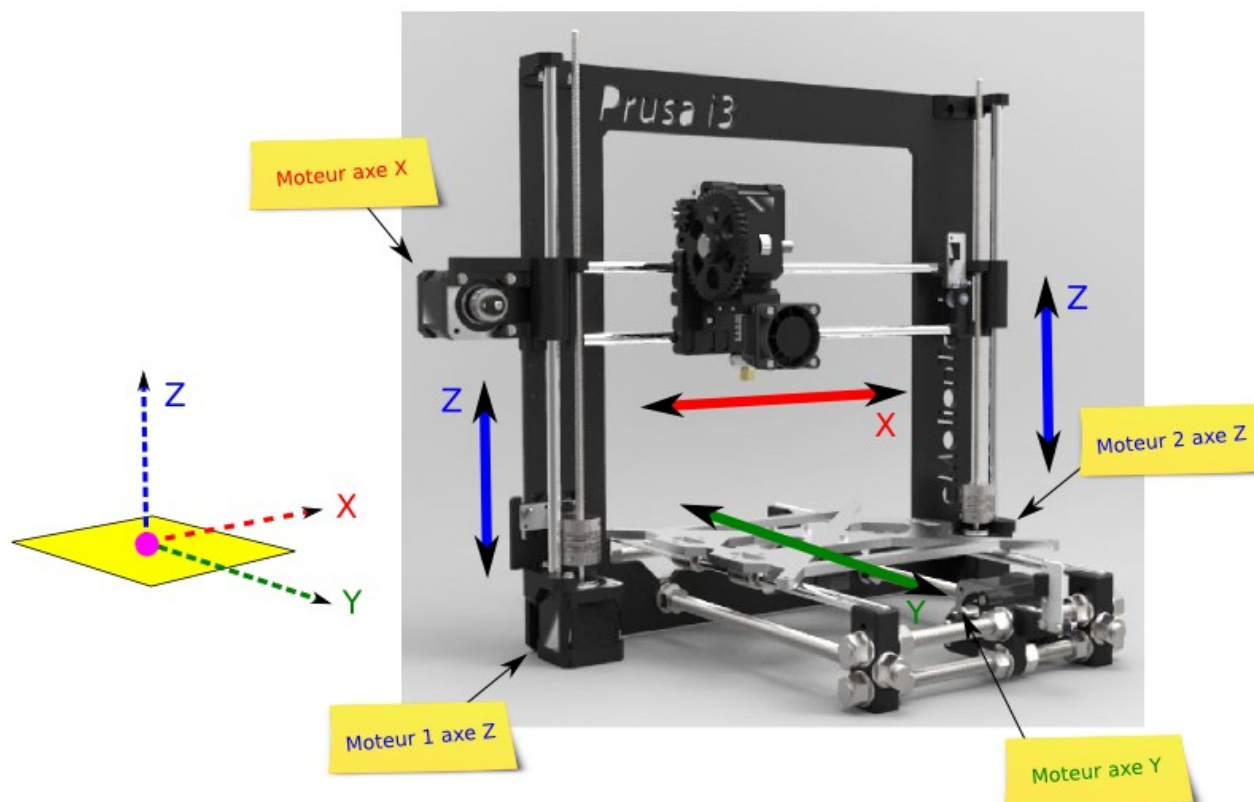
6. **Éléments fonctionnels mécaniques de positionnement d'une imprimante 3D : positionnement dans les 3 axes de l'espace X, Y et Z**

Assez logiquement, d'après ce que l'on vient de dire, on comprend qu'une imprimante 3D devra être en mesure :

- de tracer le contour d'une couche dans le plan du plateau puis de la remplir : ceci nécessite un déplacement dans le plan du plateau, c'est à dire selon l'axe Y et selon l'axe X.
- de « monter d'un cran » pour réaliser la couche suivante : ceci nécessite une mobilisation dans l'axe Z.

Ainsi très logiquement, une imprimante 3D va comporter :

- une motorisation et une mécanique de translation de l'**axe Y**
- une motorisation et une mécanique de translation de l'**axe X**
- une motorisation et une mécanique de translation de l'**axe Z, typiquement 2 moteurs**



Techniquement, chaque axe va être motorisé par un **moteur pas à pas** et disposera d'un capteur de fin de course, un **microrupteur** (capteur mécanique ON/OFF).

7. **Éléments fonctionnels d'impression d'une imprimante 3D : Chauffe de la buse et alimentation en plastique**

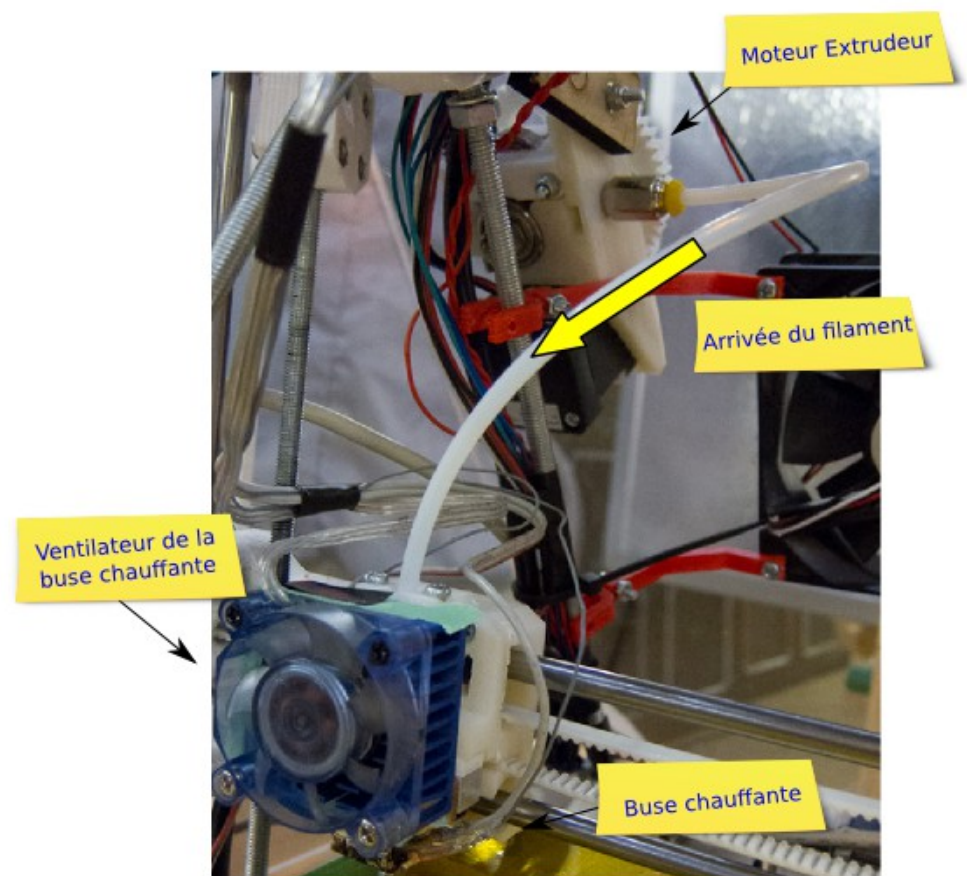
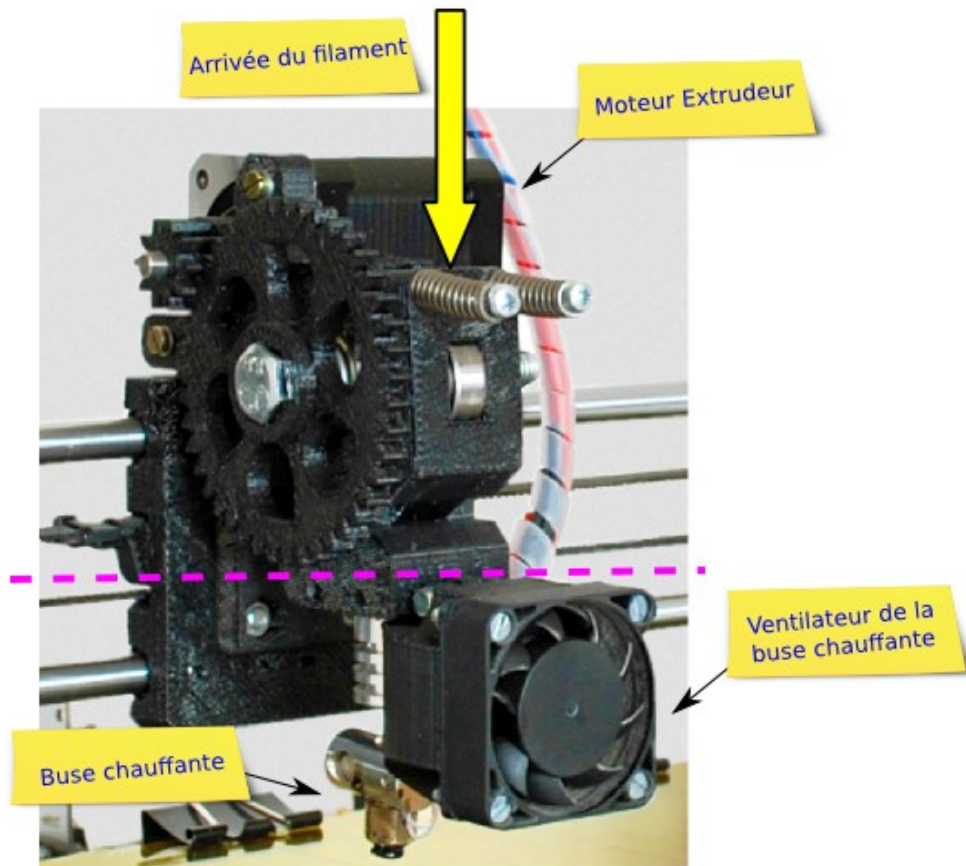
L'ensemble des motorisations que l'on vient de voir va assurer le déplacement de la buse : à son niveau devra arriver le plastique fondu.

Très logiquement :

- un ensemble mécanique, appelé « **extrudeur** » va assurer la motorisation de l'arrivée du filament du plastique non fondu
- D'autre part, **la chauffe de la buse** sera obtenue grâce à une de façon à assurer la fonte du plastique

L'ensemble extrudeur + buse pourra être :

- soit d'un **tenant** et sera placé sur un **chariot dans l'axe X**
- soit **fixé sur le châssis**, la buse étant fixée seule sur **un chariot dans l'axe X**, la connexion avec l'extrudeur se faisant par un **tube plastique**

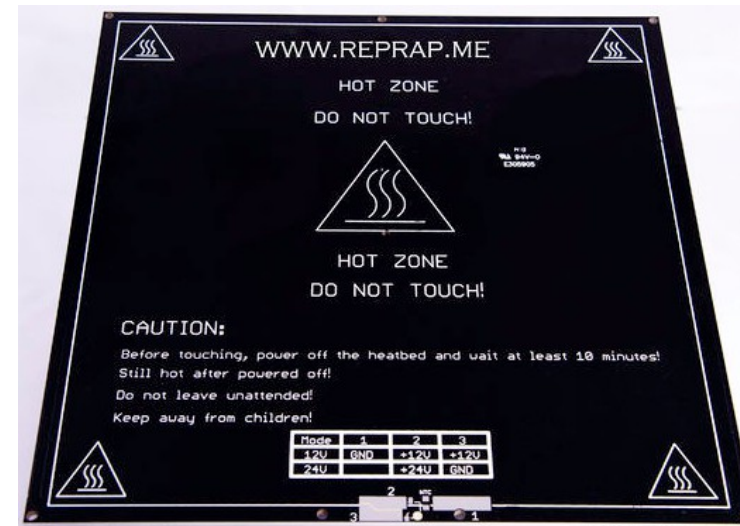
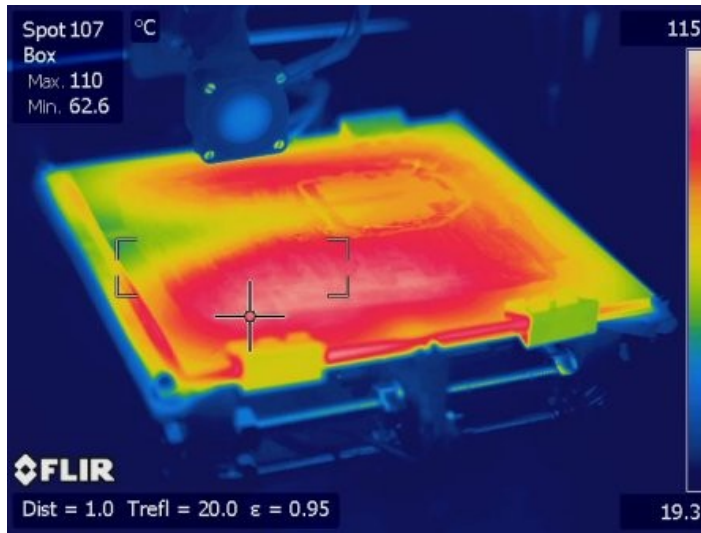


à gauche, extrudeur + buse d'un bloc (Prusa) – à droite : extrudeur et buse dissociés (Huxley +/- historique...)

8. Éléments fonctionnels d'impression d'une imprimante 3D : Plateau chauffant

L'axe Y disposera d'un **plateau chauffant** qui va permettre :

- une **adhésion optimale** du plastique d'une part
- et un **maintien en température** de la pièce plastique d'autre part



9. L'électronique d'une imprimante 3D opensource : le schéma fonctionnel

Au final, on comprend facilement qu'une imprimante 3D va intégrer une électronique permettant de contrôler :

- les 5 moteurs pas à pas bipolaires (1 axe X, 1 axe Y, 2 axes Z et 1 pour l'extrudeur)
- l'allumage/extinction du plateau chauffant et de la buse chauffante

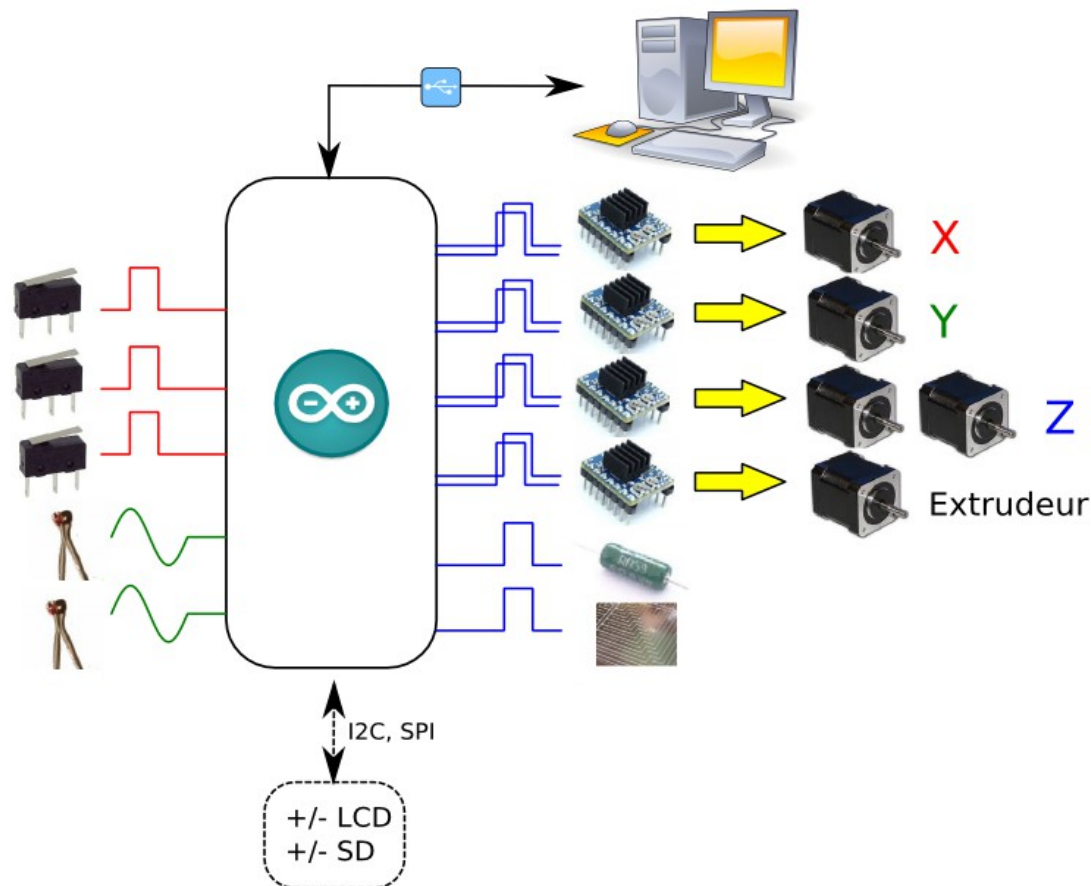
L'électronique de l'imprimante 3D va par ailleurs devoir assurer la gestion des capteurs suivants :

- lecture de l'état des microrupteurs de fin de course des axes X, Y et Z
- mesure de la température à l'aide de 2 thermistances, une pour la buse et une pour le plateau chauffant

L'électronique devra également pouvoir communiquer avec un PC de contrôle, par le port série USB.

En option, on pourra également vouloir disposer d'un afficheur LCD, d'une carte SD, etc...

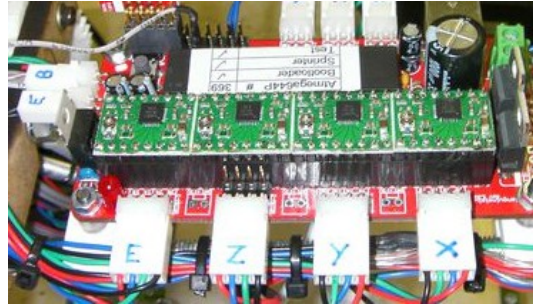
Si on fait une synthèse fonctionnelle d'une imprimante 3D, on se rend compte au final qu'il s'agit ni plus ni moins que d'un « gros montage » façon Arduino :



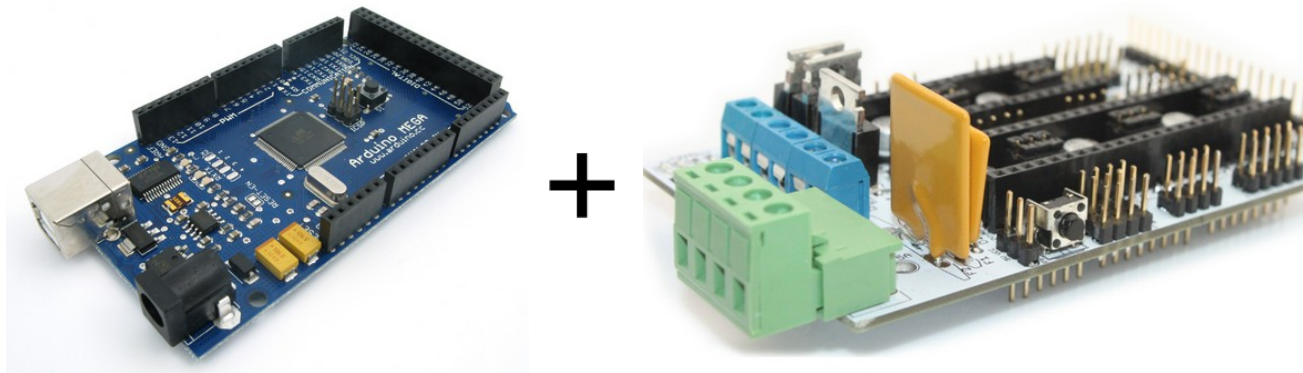
10. L'électronique d'une imprimante 3D opensource : cartes utilisables

Logiquement, **la communauté des imprimantes 3D opensource s'est tournée vers les cartes Arduino-compatibles**. Du coup, on se retrouve en terrain connu si l'on connaît Arduino. Typiquement, on peut distinguer :

- les cartes « intégrées » qui associent le micro-contrôleur et l'électronique associée pour la gestion des moteurs, etc.. (ici, l'exemple de la sanguinolulu) :



- des solutions associant une carte Arduino Mega et un shield dédié regroupant l'électronique associée, par exemple, la carte opensource Ramps :



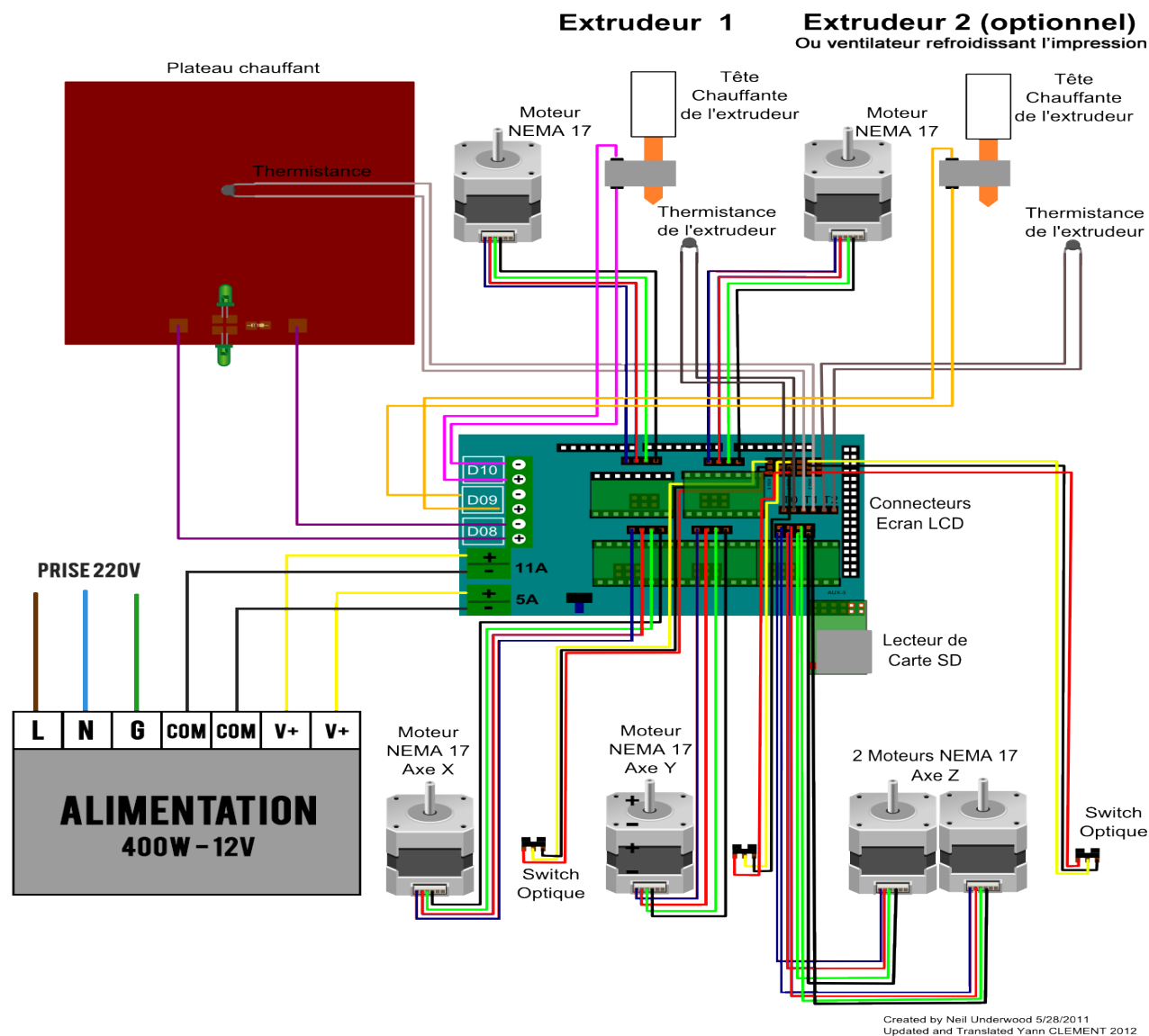
A part les étages de commande des moteurs pas à pas qui existent en « mini-shields » dédiés que l'on va enficher sur la carte électronique principale : notamment le polulu A4988



Si un driver « grille », pas besoin de tous les changer !!

11. Le câblage complet d'une imprimante 3D opensource

RepRap Arduino Mega Pololu Shield 1.4



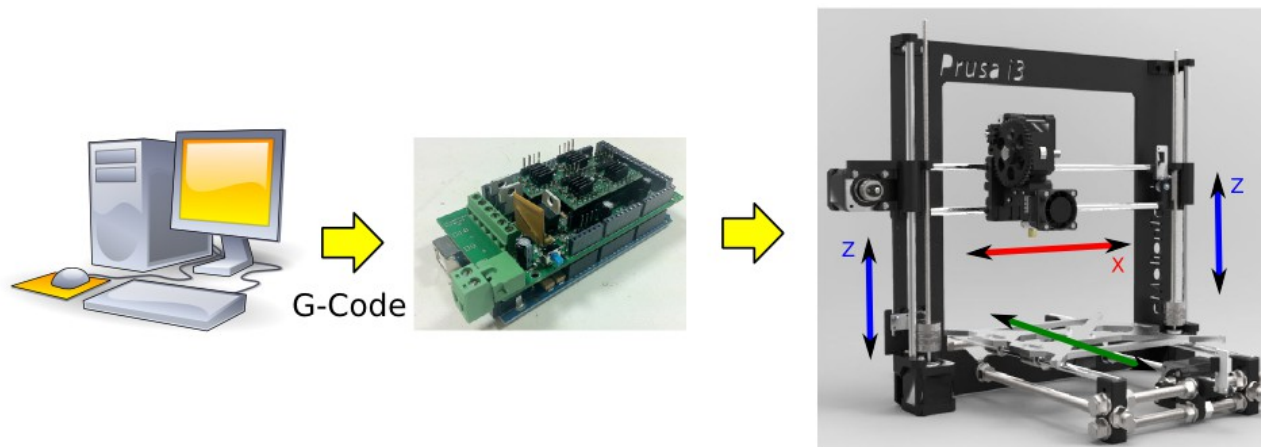
Oui, « ça fait du monde »... mais tout ça est assez logique une fois compris la fonction de chaque élément !

12. Imprimante 3D opensource : 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 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 constituées 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

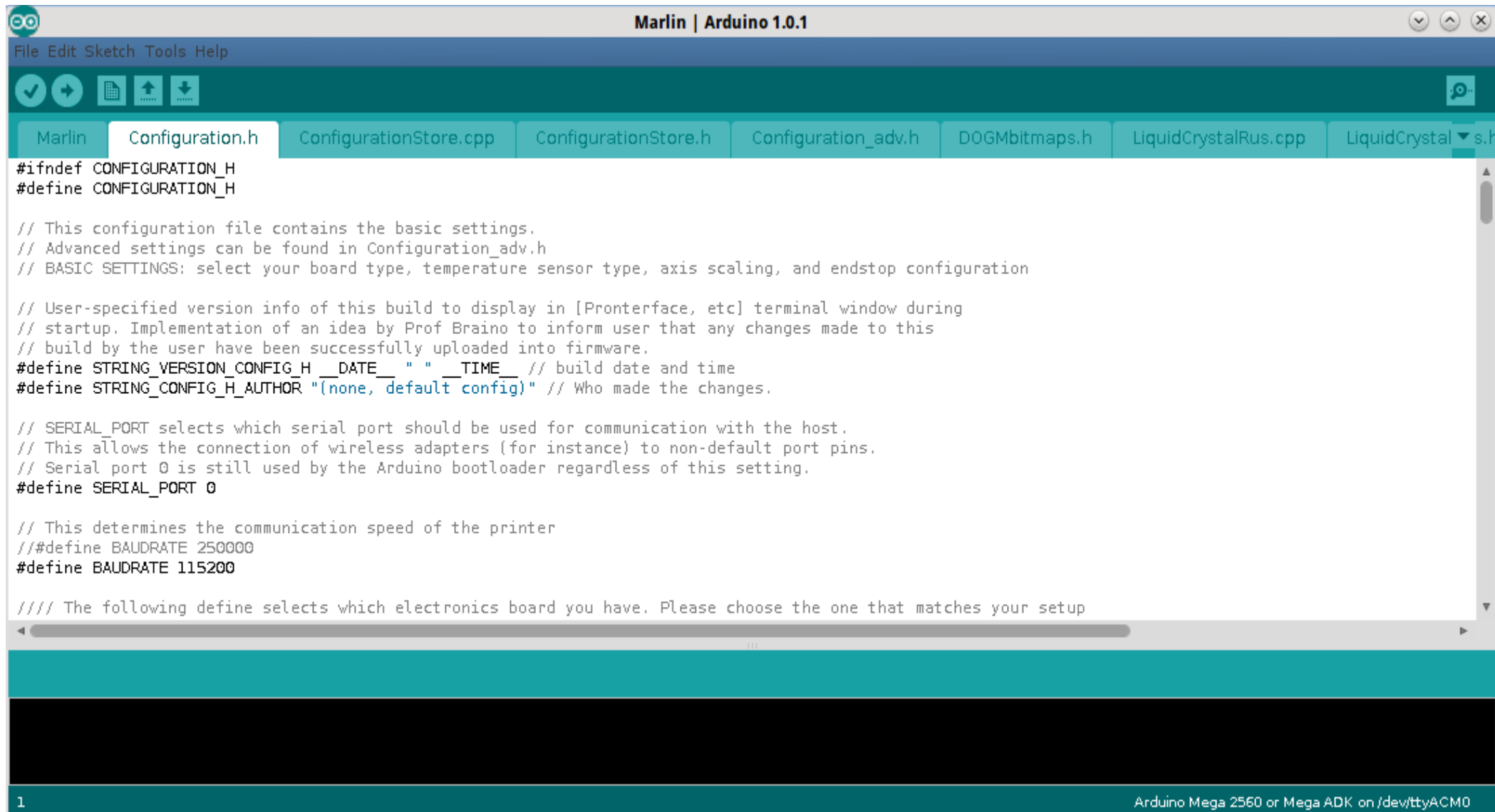
En pratique, les plus utilisés sont :

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

Principe d'installation

Le logiciel est en fait un « super-programme Arduino » et il va donc être possible de le programmer depuis l'IDE Arduino vers la carte de l'imprimante, tout simplement.

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



The screenshot shows the Arduino IDE interface with the Marlin firmware open. The title bar reads "Marlin | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, saving, and uploading. The file explorer at the top shows several files, with "Configuration.h" selected. The main editor displays the contents of "Configuration.h", which includes comments about basic settings, version information, and serial port configuration. The status bar at the bottom indicates "1" on the left and "Arduino Mega 2560 or Mega ADK on /dev/ttyACM0" on the right.

```
#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
```

Une fois programmé dans la carte de l'imprimante, on peut interagir avec l'imprimante à partir du Terminal série si on veut : comme pour un code Arduino classique ! Mais le firmware permettra surtout de contrôler l'imprimante depuis une interface graphique dédiée.

14. Vue d'ensemble d'un kit complet des pièces d'une imprimante 3D opensource

Le kit complet des pièces nécessaires pour la construction d'une imprimante 3D opensource va comprendre :

- **Les pièces du châssis X,Y,Z**
 - Pièces imprimées
 - Pièces mécaniques non imprimées :
 - Tiges lisses et filetées
 - Moteurs, roulements, courroie, poulie, coupleur d'axe...
 - Pièces découpe Laser
 - cadre châssis
 - plateau Y
- **La visserie**
- **Extrudeur et la buse chauffante**
- **Le plateau chauffant**
- **L'électronique :**
 - carte Arduino et Ramps
 - drivers moteurs
 - alimentation
 - capteurs fin de course, thermistances



15. Les grandes étapes du montage d'une imprimante 3D opensource

Les étapes du montage d'une imprimante 3D opensource sont les mêmes à quelques variantes près d'une machine à l'autre :

- Prendre le temps de lire la documentation de montage en tier avant d'attaquer le montage
- Avant de commencer : bien s'installer +++ et rassembler TOUS les outils et fournitures complémentaires qui vont être nécessaires
- **Montage du chariot Y :**
 - préparation du chariot Y
 - Assemblage des parties transversales du chariot Y
- **Assemblage de l'axe X**
 - Assemblage du X End Idler & X End Motor
 - Assemblage du chariot X (X Carriage)
 - Montage de l'axe X
- **Assemblage des axes Z et X sur le châssis**
- **Assemblage des moteurs** : axe Z, axe Y, axe X
- **Mise en place du système d'entraînement** : courroie axe X, courroie axe Y
- **Montage du plateau chauffant** :
 - préparation thermisance,
 - mise en place du plateau chauffant
- **Assemblage et montage de l'extrudeur et de la buse chauffante**
 - Assemblage de l'extrudeur
 - Fixation à l'axe X et montage du moteur
 - Réglage de la hauteur de la buse d'extrusion (réglage du Z)
- **Câblage électrique**
 - Préparation et fixation de l'électronique
 - Branchements des éléments sur l'électronique : capteurs fin de course, moteurs, thermistances, résistance chauffante, plateau chauffant, ventilateur
 - câblage alimentation
- **FINI !! Enfin, plutôt le début d'une belle aventure !**

Monter une imprimante 3D nécessite de la rigueur mais est un excellent exercice réalisant une synthèse entre plusieurs techniques : mécanique, électronique, informatique, etc... Il y a ceux qui ont construit une imprimante 3D qui imprime, et il y a les autres... (C'est pour rire...)

16. Détails mécaniques de la motorisation des axes

Axe X et Y

Moteur 200 pas

Interface 1/16ème de pas : soit 3200 « micro-pas » par tour !

GT2 = 2 mm entre chaque dent. Donc un tour de moteur pas à pas = 32mm soit une précision théorique de $32/3200 = 0,01\text{mm}$ / micropas

La documentation annonce une précision nominale de 0,015mm pour les axes X et Y

Axe Z

Moteur 200 pas

Interface 1/16ème de pas : soit 3200 pas par tour !

Entraînement direct d'une tige fileté M5 (ou M6) soit 0,8 mm / tour (ou 1mm/tour) soit une précision théorique de 0,25 µm/ micropas !

La documentation annonce une précision nominale de 0,785µm pour l'axe Z

http://fr.wikipedia.org/wiki/Calcul_de_vis_d%27assemblage#G.C3.A9om.C3.A9trie_de_la_vis

Pour plus de détails : <http://calculator.josefprusa.cz/>

17. Zoom sur les drivers de moteurs : le mini-shield A4988



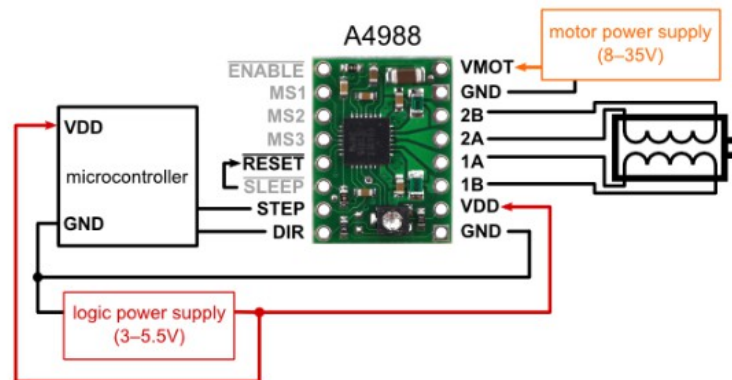
Présentation

Il s'agit d'un **mini-shield de contrôle de moteur pas à pas bipolaire**, très puissant dans ses possibilités puisque capable de générer des séquences de full step jusqu'au 16ème de pas ! Autrement dit, avec un moteur de 200 pas, vous disposez potentiellement avec ce shield d'un moteur $200 \times 16 = 3200$ pas !!

Très souple aussi d'emploi puisque peut contrôler des moteurs pas à pas de tension variée, même inférieure à la tension d'alimentation de l'étage moteur, grâce à la possibilité de fixer l'intensité maximale circulant dans les phases, indépendamment de la tension d'alimentation du moteur !

Le tout pour un pris low-cost (dispo dès 7€) !

Brochage



Configuration du mode de fonctionnement

La configuration du mode fonctionnement se fait avec les broches MS1, MS2 et MS3 : pour le mode 1/16ème de pas, elles doivent toutes être au niveau HAUT.

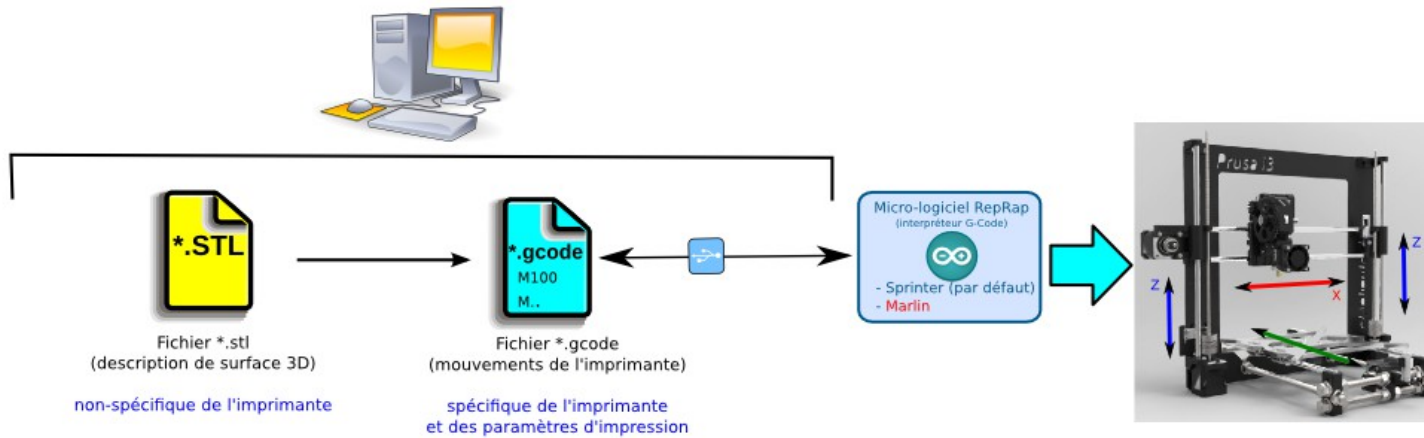
High High High Sixteenth step

Il est génial ce mini-shield !!

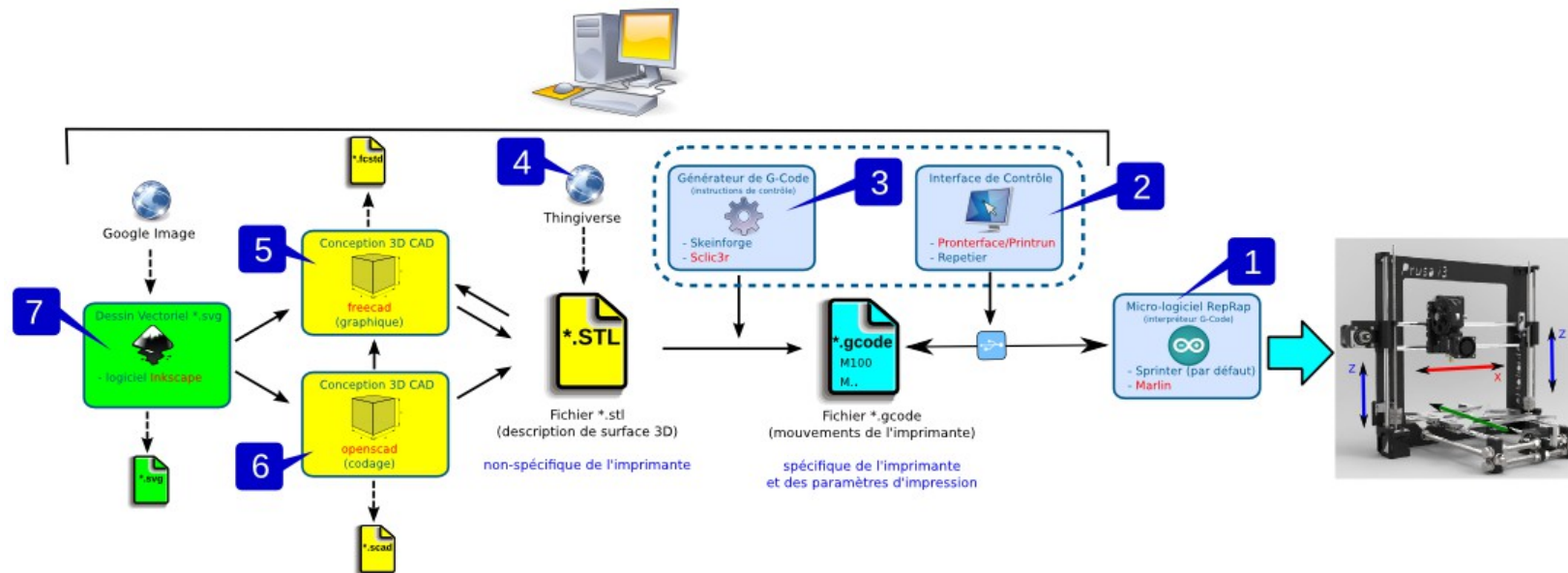
18. Conclusion

A ce stade, vous avez une idée du principe de fonctionnement d'une imprimante 3D programmée avec un décodeur de G-Code.

A présent, vous comprenez qu'il va falloir être capable de transformer un fichier décrivant une pièce 3D en un fichier de G-Code : ceci va être réalisé à l'aide d'une « chaîne logicielle » opensource sur le poste de contrôle... Ce sera l'objet d'un autre tuto.



La chaîne logicielle « courte » pour imprimer des pièces 3D



La chaîne logicielle « longue » pour imprimer des pièces 3D

19. A présent, vous devriez être capable :

- de comprendre et d'expliquer le principe fonctionnel d'une imprimante 3D opensource.

Table des matières

Structure fonctionnelle d'une imprimante 3D open-source

Intro |

Les imprimantes 3D open-source : petite histoire |

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

Rappel : Représentation dans le plan et dans l'espace |

Représentation dans l'espace : application à l'imprimante 3D |

Éléments fonctionnels mécaniques de positionnement d'une imprimante 3D : positionnement dans les 3 axes de l'espace X, Y et Z |

Éléments fonctionnels d'impression d'une imprimante 3D : Chauffe de la buse et alimentation en plastique |

Éléments fonctionnels d'impression d'une imprimante 3D : Plateau chauffant |

L'électronique d'une imprimante 3D opensource : le schéma fonctionnel |

L'électronique d'une imprimante 3D opensource : cartes utilisables |

Le câblage complet d'une imprimante 3D opensource |

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

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

Vue d'ensemble d'un kit complet des pièces d'une imprimante 3D opensource |

Les grandes étapes du montage d'une imprimante 3D opensource |

Détails mécaniques de la motorisation des axes |

Zoom sur les drivers de moteurs : le mini-shield A4988 |

Conclusion |

A présent, vous devriez être capable : |

Bravo !
vous avez terminé cet atelier !



Prêt pour la suite ? Retrouvez de nombreux autres thèmes d'ateliers « Impression 3D » ici :
http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.ATELIERSIMPRESSIION3D