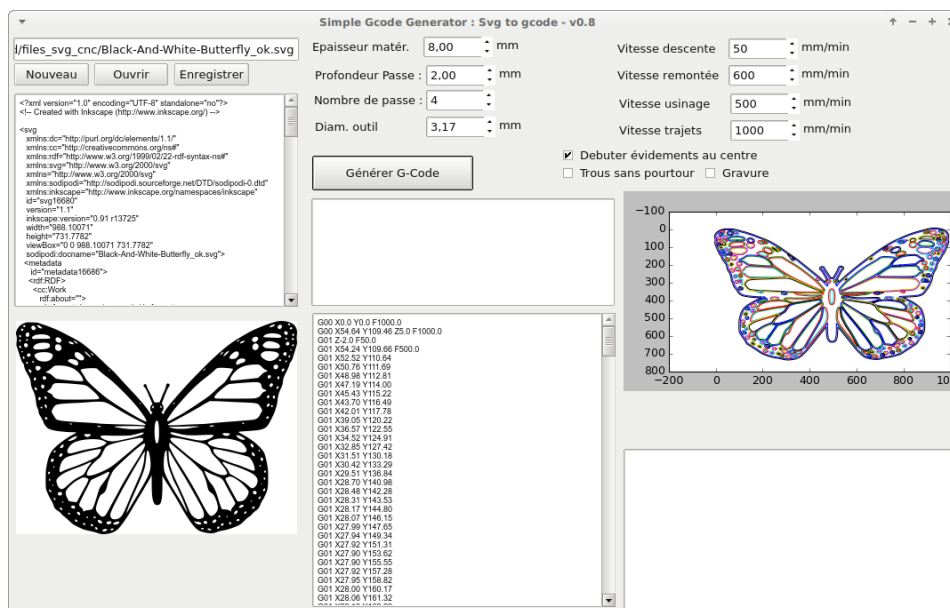


Atelier Fab'2D : Découvrir Simple G-Code Generator

Par X. HINAULT – www.mon-club-elec.fr – Juin 2016 - Tous droits réservés - Licence [Creative Commons BY NC SA](https://creativecommons.org/licenses/by-nc-sa/4.0/)



De quoi parle-t-on ?

Simple G-Code Generator est une interface graphique Python (PyQt) libre que nous avons développée (X. HINAULT) et qui permet de générer simplement du G-Code à partir d'un simple dessin SVG segmentisé.

En clair, Simple-Gcode Generator permet de générer du G-Code utilisable avec l'[Open Maker Machine](https://openmakermachine.com/) ou l'[Open Maker Machine PRO](https://openmakermachine.com/pro/) ou l'[Open Maker Machine PLUS](https://openmakermachine.com/plus/) à partir d'un simple dessin réalisé dans Inkscape.

Potentiellement le G-Code produit avec Simple G-Code Generator est utilisable avec toute machine de découpe 2D propulsée par [GRBL](https://github.com/gnea/grbl), le micrologiciel décodeur de G-Code opensource programmé dans l'Arduino.

Pour info, le Simple G-Code Generator fait partie d'une petite suite logicielle comprenant :

- **Simple G-Code Generator** : un générateur de G-Code à partir de SVG segmentisé.
- **Simple G-Code GUI** : une interface de contrôle d'une machine de fabrication numérique 2D, notamment utilisable avec nos machines l'[Open Maker Machine](https://openmakermachine.com/) ou l'[Open Maker Machine PRO](https://openmakermachine.com/pro/) ou l'[Open Maker Machine PLUS](https://openmakermachine.com/plus/)
- **Simple G-Code Decoder** : un micrologiciel Arduino décodeur de G-Code minimaliste (10Ko) mais permettant de créer facilement de petites machines numériques. Je conseille cependant plutôt l'utilisation de GRBL, notamment pour l'[Open Maker Machine PRO](https://openmakermachine.com/pro/) ou l'[Open Maker Machine PLUS](https://openmakermachine.com/plus/)

Ces logiciels constituent un ensemble cohérent opérationnel et opensource permettant de contrôler une machine de fabrication numérique 2D : ils sont cependant utilisables indépendamment les uns des autres, entre-eux ou non.

Note aux utilisateurs :

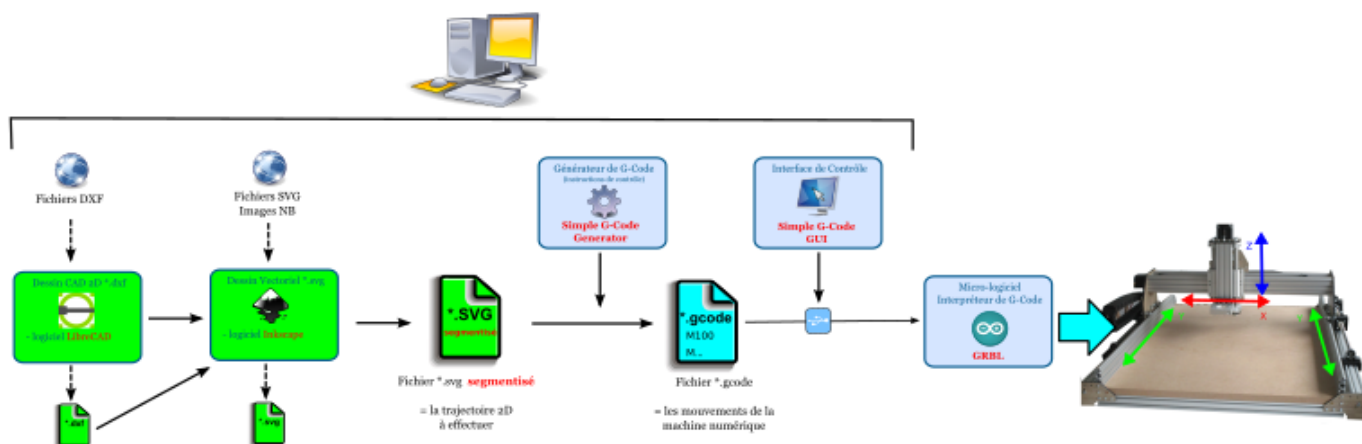
Il existe de nombreux générateurs de G-Code 2D, souvent très complets mais un peu déroutants pour des besoins simples... Plus rares sont les générateurs de G-Code 2D libres et opensource. Notre but ici est clairement de proposer un outil pragmatique, simple et rapide à utiliser par des « néophytes » à l'aide des outils habituels des « makers », tout en étant libre et opensource.

Nous utilisons Simple G-Code Generator tous les jours en production et nous l'enrichissons progressivement lorsque des besoins se font sentir.

Pour comprendre : la chaîne logicielle de fabrication numérique 2D

Pour pouvoir réaliser une pièce 2D en fabrication numérique avec Simple G-Code Generator, on devra disposer d'un fichier **SVG segmentisé** (autrement dit un fichier de dessin vectoriel 2D ne comportant que des points et des segments) : ce fichier décrit la trajectoire à effectuer.

Ce SVG segmentisé, **très simple à obtenir à l'aide d'un logiciel libre tel que [Inkscape](#)**, permettra de générer le G-Code correspondant à l'aide de Simple G-Code Generator.

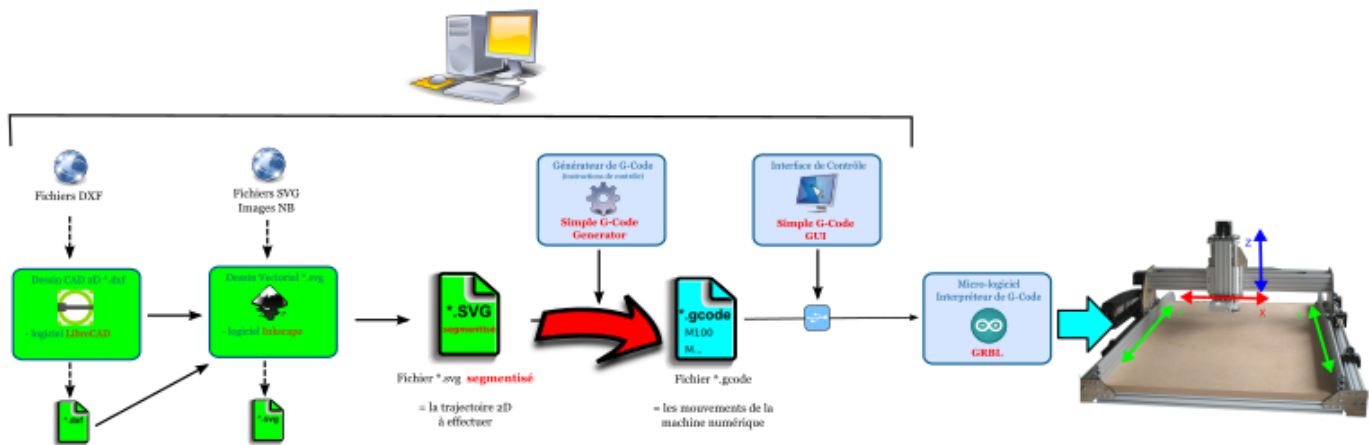


Le fichier SVG (dessin vectoriel) de départ pourra être obtenu indifféremment :

- par **dessin « from scratch »** avec le logiciel Inkscape
- par **simple téléchargement d'un fichier SVG**,
- à partir d'une **image Noir et Blanc** (détection de contour à l'aide d'Inkscape)
- à partir d'un **fichier CAD 2D** (logiciel LibreCAD) au format DXF importable dans Inkscape)
- et même **à partir d'un logiciel de conception 3D** tel qu'Openscad qui permet d'exporter en *.dxf (lui-même importable dans Inkscape) n'importe quelle face ou tranche d'un objet 3D grâce à la fonction projection() (pont entre modélisation 3D vers 2D!). Noter que Blender également permet des conversions en SVG à partir d'objet 3D.

Bref, de nombreuses possibilités qui vous permettront de faire face simplement à de nombreux scénarios.

L'étape présentée ici concerne la conversion du SVG segmentisé en G-Code (instructions de mouvements que la machine devra exécuter) :



Installation

Pré-requis

Simple G-Code Generator nécessite un système sous Gnu/Linux avec Python 2.7 installé. Personnellement, je conseille d'utiliser une Debian Testing sous XFCE mais n'importe quel Xubuntu devrait faire l'affaire.

Les dépendances utiles sont par ailleurs à installer avec la commande suivante à saisir dans un Terminal :

```
sudo apt-get update
```

```
sudo apt-get install python-shapely python-qt4 python-matplotlib
```

Note : Ce logiciel étant développé avec PyQt 4.x, il est utilisable potentiellement sur Windows et Mac mais je n'ai pas à ce jour écrit les versions pour ces plateformes.

Téléchargement

Le Simple G-Code Generator est téléchargeable ici :

http://cloud-mon-club-elec.fr:8080/pyqt_apps/simple_gcode_generator/

Choisir de préférence la version la plus récente. Dézipper l'archive obtenue. Le répertoire contient :

```

simple_gcode_generator_svg_to_gcode.pyc
simple_gcode_generator_svg_to_gcode.ui
triskel.svg
simple_gcode_generator_svg_to_gcode.py
simple_gcode_generator_svg_to_gcodeMain.py

```

Lancement

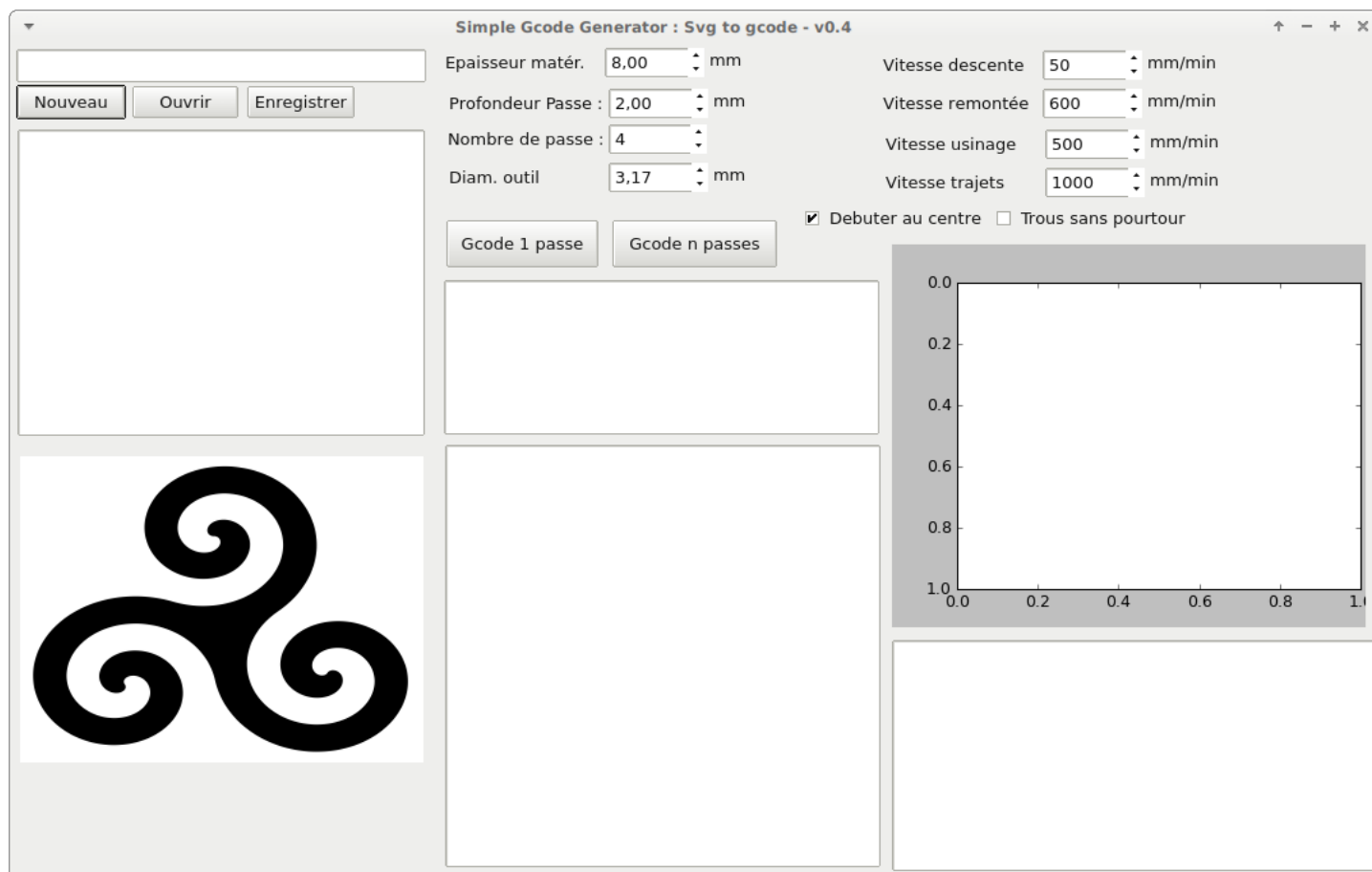
Dans le répertoire obtenu, le fichier à exécuter est le *.main :

- soit en le rendant exécutable (clic droit > propriétés > Permissions > cocher rendre exécutable)
- soit en ouvrant un terminal dans le répertoire où il se trouve et en saisissant

```
./simple_gcode_generator_svg_to_gcodeMain.py
```

- soit en l'ouvrant avec l'éditeur Geany et en l'exécutant.

Au lancement, si aucune dépendance n'est manquante, on obtient :

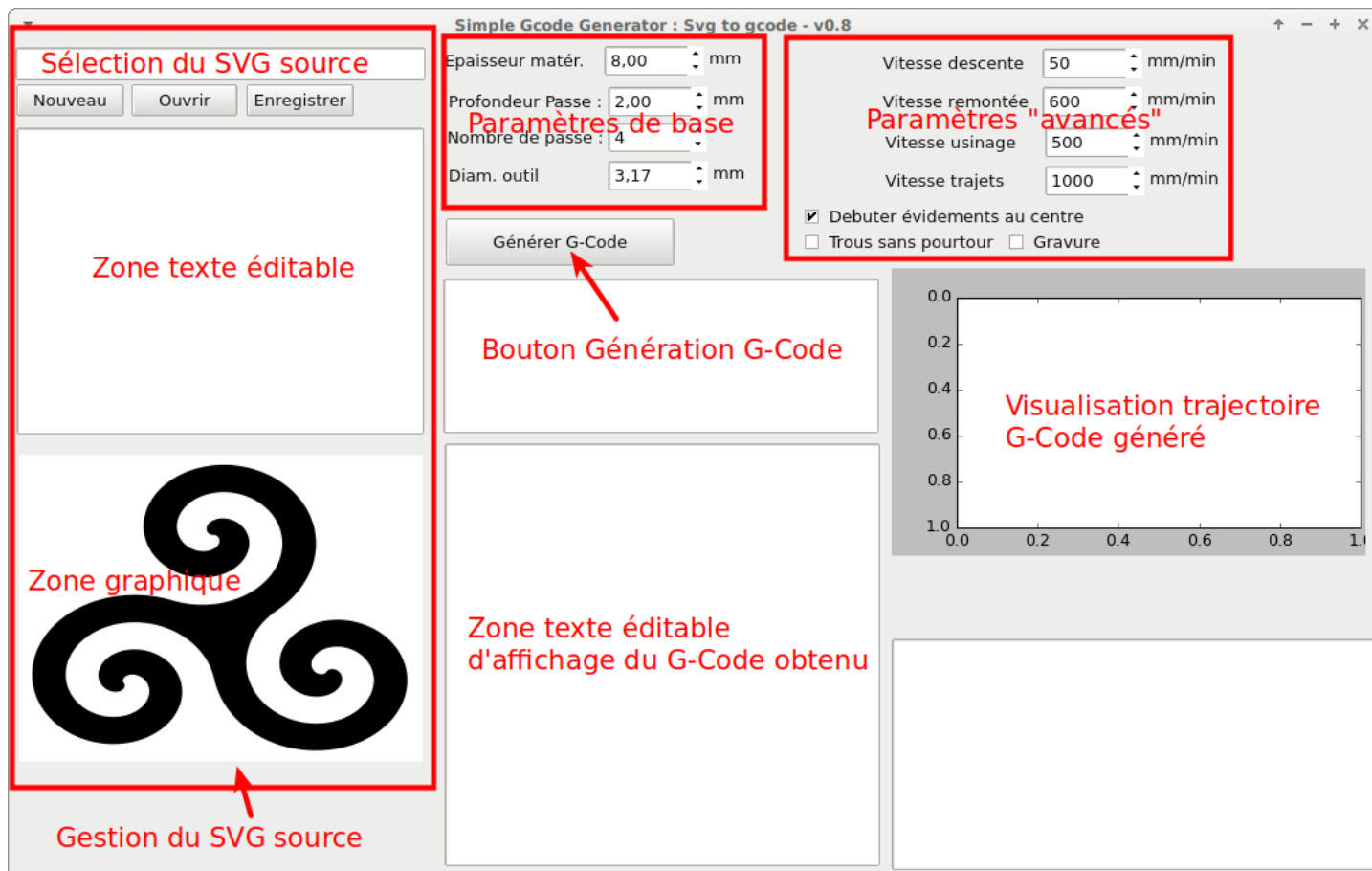


Le triskel présent dans la fenêtre est un SVG de test qui est chargé par défaut.

Présentation de l'interface

Simple G-Code Generator se veut simple et logique... On a donc :

- une zone de **chargement du fichier SVG** source qui comprend :
 - un champ + boutons d'ouverture d'un fichier SVG
 - une zone texte éditable où sera affiché le source du SVG ouvert
 - une fenêtre de visualisation du SVG ouvert. Le Triskell affiché par défaut l'est dans un simple but de test.
- D'une série de **champs de paramétrage** comprenant :
 - des paramètres de base limités à l'essentiel utile en pratique
 - des paramètres « avancés » correspondant aux vitesses de progression et à la sélection du mode de génération du G-Code. Ces paramètres peuvent être laissés en l'état par défaut.
- D'une zone de **gestion du G-Code** généré comportant :
 - un bouton à cliquer pour lancer la génération du G-Code
 - d'une zone texte d'affichage du G-Code obtenu
 - d'une visualisation de la trajectoire du G-Code généré superposée aux chemins extraits du SVG source.



Générer un G-Code à partir d'un fichier SVG segmentisé

Le point de départ : un fichier SVG segmentisé

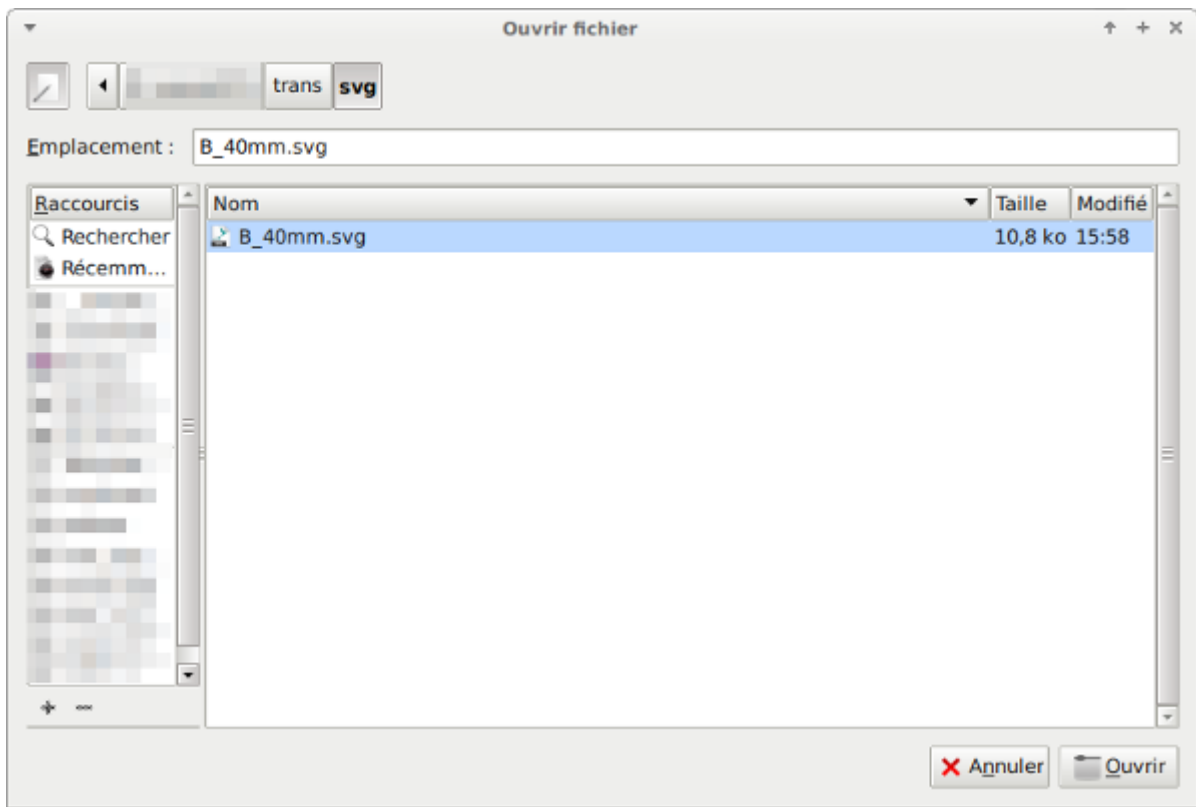
Simple G-Code Generator accepte en entrée un fichier SVG segmentisé : en clair, un SVG qui ne comporte que des chemins composés de points et de segments. La procédure (simple) à suivre dans Inkscape pour ce faire est décrite séparément.

A des fins de test, vous trouverez ici quelques fichiers SVG segmentisés prêts à l'emploi :

http://cloud-mon-club-elec.fr:8080/files_svg_cnc/

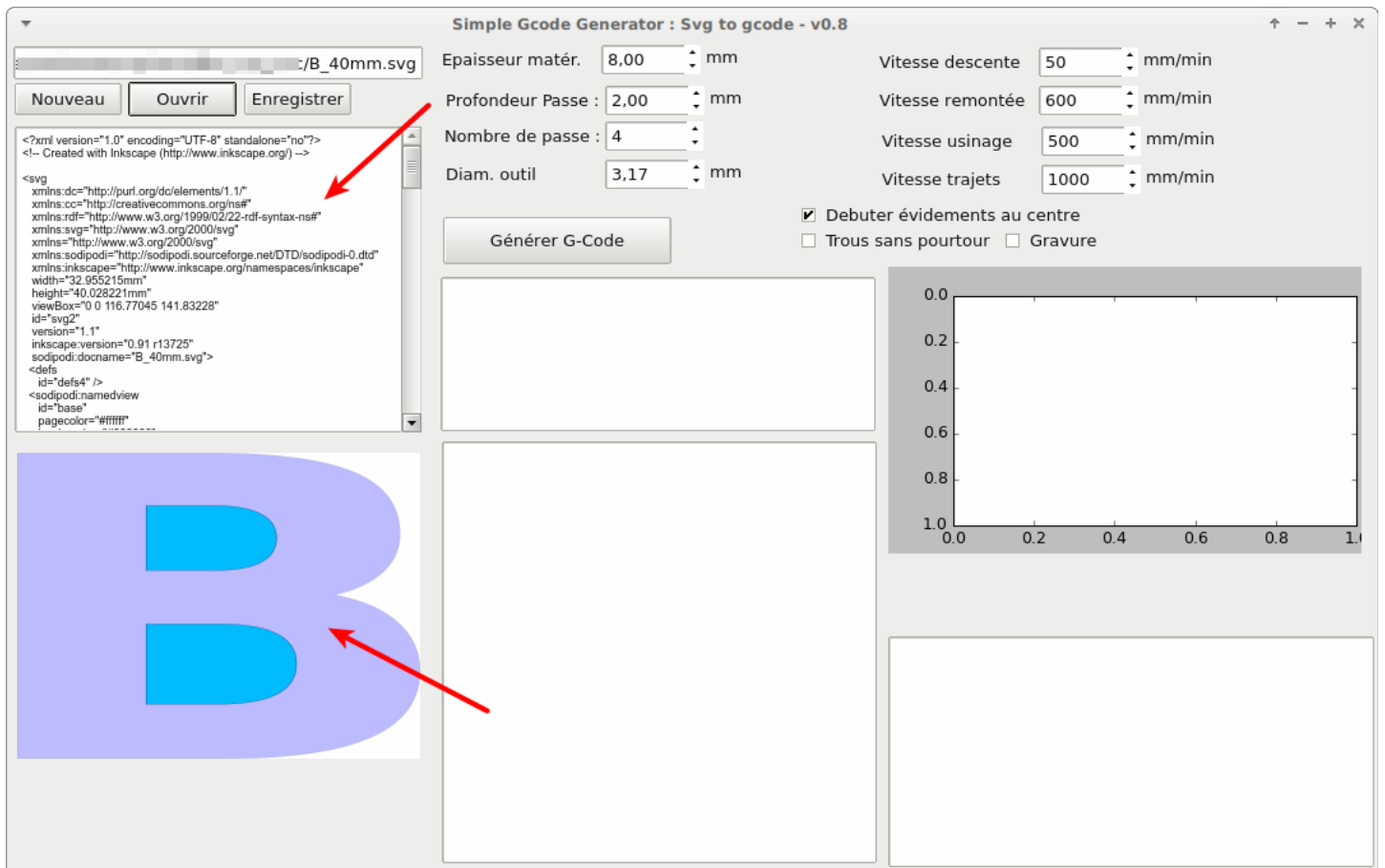
Ouvrir le fichier SVG segmentisé

Cliquer sur le bouton <OUVRIR> et sélectionner le fichier SVG voulu :



Une fois fait, vous devez voir apparaître :

- le source du fichier SVG dans le champ prévu à cet effet (ce qui peut être utile en cas de problème de génération...)
- mais surtout le visuel du SVG chargé



Fixer les paramètres de base à utiliser

Vous devez fixer de base 4 paramètres « complètement logiques » et simples :

- l'épaisseur du matériau à découper en mm
- la profondeur de la passe en mm
- le nombre de passe (qui vaut logiquement épaisseur / profondeur passe) – sans unité
- le diamètre d'outil en mm : le diamètre de la fraise utilisée. Mettre 0 si vous utiliser la découpe vinyle ou le dessin.

Simple Gcode Generator : Svg to gcode - v0.8

Epaisseur matér.	8,00	mm	Vitesse descente	50	mm/min
Profondeur Passe :	2,00	mm	Vitesse remontée	600	mm/min
Nombre de passe :	4		Vitesse usinage	500	mm/min
Diam. outil	3,17	mm	Vitesse trajets	1000	mm/min

Debuter évidements au centre
 Trous sans pourtour Gravure

Générer G-Code

Note : Les valeurs par défaut sont celles pour une découpe de 8mm en 4 passes de 2mm avec une fraise de 3.17, ce qui correspond à mon scénario de découpe le plus fréquent...

± Fixer les paramètres « avancés »

Optionnellement, vous pouvez modifier les paramètres « avancés » qui concernent essentiellement les vitesses de progression utilisées pour générer le G-Code ainsi que le choix du mode de génération du G-Code. **Dans une première approche, vous pouvez laisser en l'état les valeurs utilisées.**

Simple Gcode Generator : Svg to gcode - v0.8

Epaisseur matér.	8,00	mm	Vitesse descente	50	mm/min
Profondeur Passe :	2,00	mm	Vitesse remontée	600	mm/min
Nombre de passe :	4		Vitesse usinage	500	mm/min
Diam. outil	3,17	mm	Vitesse trajets	1000	mm/min

Debuter évidements au centre
 Trous sans pourtour Gravure

Générer G-Code

Générer le G-Code

Le plus dur reste à faire : cliquer sur le bouton <Générer le G-Code> !

Simple Gcode Generator : Svg to gcode - v0.8

Epaisseur matér. : <input type="text" value="8,00"/> mm	Vitesse descente : <input type="text" value="50"/> mm/min
Profondeur Passe : <input type="text" value="2,00"/> mm	Vitesse remontée : <input type="text" value="600"/> mm/min
Nombre de passe : <input type="text" value="4"/>	Vitesse usinage : <input type="text" value="500"/> mm/min
Diam. outil : <input type="text" value="3,17"/> mm	Vitesse trajets : <input type="text" value="1000"/> mm/min

Debuter évidements au centre
 Trous sans pourtour Gravure

Une fois fait, vous devez voir apparaître le G-Code dans la zone texte dédiée et le visuel des trajectoires calculées dans la fenêtre graphique prévue pour cela. **Un fichier de G-Code portant exactement le même nom que le fichier SVG source est automatiquement créé dans le même répertoire que le fichier SVG source.**

Debuter évidements au centre
 Trous sans pourtour Gravure

Le G-Code

```
G00 X0.0 Y0.0 F1000.0
G00 X15.66 Y11.17 Z5.0 F1000.0
G01 Z-2.0 F50.0
G01 X12.14 Y8.52 F500.0
G01 X12.14 Y13.82
G01 X16.25 Y13.82
G01 X16.52 Y13.82
G01 X16.76 Y13.81
G01 X17.00 Y13.79
G01 X17.22 Y13.77
G01 X17.42 Y13.74
G01 X17.62 Y13.70
G01 X17.80 Y13.66
G01 X17.97 Y13.62
G01 X18.13 Y13.57
G01 X18.27 Y13.51
G01 X18.41 Y13.46
G01 X18.53 Y13.40
G01 X18.65 Y13.33
G01 X18.75 Y13.27
G01 X18.85 Y13.20
G01 X18.94 Y13.13
G01 X19.02 Y13.05
G01 X19.09 Y12.98
G01 X19.16 Y12.90
G01 X19.23 Y12.81
G01 X19.29 Y12.73
G01 X19.34 Y12.63
G01 X19.39 Y12.53
G01 X19.44 Y12.42
G01 X19.48 Y12.30
G01 X19.52 Y12.17
```

Les trajectoires

Si par hasard vous n'obtenez rien ou bien si le résultat est « aberrant », c'est que votre SVG segmentisé n'est pas correctement préparé.

Noter que dans la fenêtre graphique, les trajectoires semblent « dédoublées ». En fait, il y a un double tracé qui est effectué :

- D'une part le tracé du chemin « brut » tel qu'il est fourni par le SVG
- et d'autre part le tracé des trajectoires calculées.

Ceci permet de vérifier la cohérence du G-Code obtenu, à savoir :

- que les trajectoires des contenants sont bien dilatées,
- et celles des contenus sont bien érodées.

C'est fini !

Voilà, vous avez généré votre premier G-Code ! Comme vous pouvez le constater, c'est simple, logique et intuitif.

A présent, il ne vous reste plus qu'à ouvrir le fichier de G-Code obtenu avec l'interface de contrôle de votre machine, par exemple [Simple G-Code GUI](#), à caler votre Z et à lancer la découpe.

The screenshot shows the 'Simple G-Code GUI for GRBL v0.6.16.30' interface. It includes a configuration section for the serial port (Port Série: /dev/ttyACM0, Débit Série: 115200 bauds) and a 'Réception sur le port série' area. Below this is a grid of buttons for G-code commands: Home X, Goto X0, Home Y, Goto Y0, Home Z, Goto Z0, RAZ Z, Home XY, RAZ X, RAZ Y, and RAZ Y. A speed control slider is at the bottom left. On the right, the generated G-code is displayed in a text area, starting with 'G00 X0.0 Y0.0 F1000.0' and ending with 'G01 X18.27 Y13.57'. The 'Envoyer GCode' button is highlighted in yellow.

Utiliser les paramètres « avancés »

Vous pouvez également fixer d'autres paramètres optionnels que je détaille ici :

Sélectionner les vitesses de travail à utiliser

Vous pouvez définir les vitesses suivantes, en mm/min :

- la vitesse de descente : c'est la vitesse d'entrée dans la matière. Personnellement, j'ai tendance à la laisser assez basse de façon à éviter les déformations des trous fraiser et limiter leur ovalisation.
- La vitesse de remontée : c'est la vitesse de sortie de la matière.
- La vitesse de découpe : là, ça dépend du matériau utilisé, de l'outil utilisé, de la vitesse de rotation de l'outil, etc... Sur des matériaux durs, plutôt lent et plutôt élevé sur des matériaux tendres. La valeur de 500mm/min est plutôt moyenne à basse. Se référer aux tableaux de synthèse des scénarios « type ».
- La vitesse des trajets : c'est la vitesse hors matière. Plutôt haute sans difficulté.

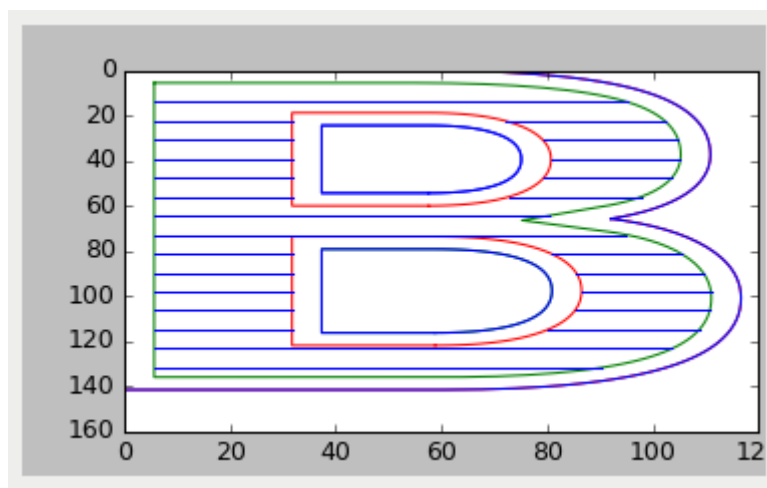
Vitesse descente	50	mm/min
Vitesse remontée	600	mm/min
Vitesse usinage	500	mm/min
Vitesse trajets	1000	mm/min

Sélectionner le mode de génération du G-Code à utiliser

Plusieurs options fixant la façon dont le G-Code est généré sont disponibles :

- Débuter évidements au centre
 Trous sans pourtour Gravure

- « **Débuter les évidements au centre** » : si cochée (par défaut), les trous et évidement sont commencés au centre, ce qui donne une meilleure circularité des trous de perçage notamment usinés en fraisage. A décocher en cas de résultat aberrant sur des évidement non-symétrique
- « **Trous sans pourtour** » : si cochée, cette option permet de découpage des évidements dans une plaque déjà à la taille sans les dilater mais en les érodants. Intéressant pour un perçage simple par fraisage d'un plateau déjà pré-découpé.
- « **Gravure** » : si cochée, cette option génère un G-Code de gravure permettant de graver des formes et des lettres dans une plaque en les évidant : les contenants sont érodés, les contenus sont dilatés et un évidement bord à bord est réalisé par lignes successives. Dans ce scénario, une passe peu profonde suffit en général. Les autres options sont inactives si ce mode est sélectionné.



Exemple de trajectoire en mode « Gravure »

Les rouages internes de Simple G-Code Generator

Le principe général de Simple G-Code Generator est le suivant :

- **extraction des coordonnées des points des objets chemins** (Path) présents au sein du fichier SVG : ceci est réalisé en utilisant la classe `minidom` du module Python `xml.dom` qui permet de parser facilement un fichier au format xml pour en extraire des données (le fichier SVG est un fichier XML).
- Ensuite, **à partir des points, des objets polygon sont construits** à l'aide de la librairie Python `shapely`.
- Une fois les objets polygon obtenus, toujours à l'aide des fonctions fournies par la librairie `Shapely`, une **analyse géométrique permet de définir lesquels sont des contenants et lesquels sont des contenus**.
- Une fois les polygones classés, on effectue une **dilatation sur les contenants et une érosion sur les contenus**, toujours à l'aide des fonctions de la librairie Python `Shapely` décidément très

pratique ici.

- Une fois fait, on **génère le fichier texte de G-Code** à l'aide des points des nouveaux objets polygones obtenus.
- L'**affichage graphique** des trajectoires est réalisé à l'aide de la librairie Python Matplotlib. Ceci est surtout utile à des fins de vérification visuelle du résultat obtenu.

L'interface graphique est par ailleurs codée avec PyQt 4.x. L'ensemble est une synthèse intéressante qui montre ce qu'il est possible de faire en Python assez simplement.

Le code source étant directement disponible, tout à chacun peut apporter sa contribution ou faire évoluer ce code qui est sous licence GPL v3.

Conclusion

Ce soft est somme toute « basique » comparativement à d'autres logiciels tel que Slic3R mais il fait le job' et la précision est au rendez-vous. De plus, il est relativement accessible en terme de codage pour pouvoir être rapidement adapté à des besoins spécifiques.