**Fraiseuse Numérique Pilotée à l’aide d’un CPC**

Pour ceux que ca pourrais intéresser, voila un petit article technique pour accompagner la vidéo CNC 3 axes sur CPC.

**Présentation :**

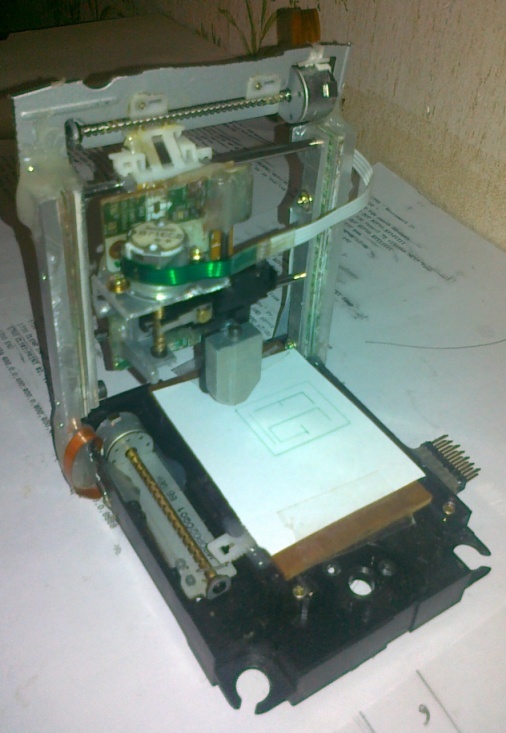
Pourquoi ce projet ?

Tout simplement par ce que j’avais envi d’avoir une fraiseuse numérique et de la piloter avec mon bon vieux CPC. Sur internet on trouve plein de vidéos de fraiseuse « Home made », mais la plus part utilise des composants standards et un PC avec un logiciel provenant du net.

C’est là qu’es l’originalité du projet, utiliser le moins de produits standard utilisé habituellement pour ce genre de machine.

Alors quoi de plus inattendu que de piloter cette machine avec un CPC !

**Description :**

Ce projet permet de piloter un robot 3 axes depuis un CPC, il met en œuvre 3 moteur pas à pas grâce à des cartes de contrôles

A l’heure actuelle le programme ne permet de piloter qu’un axe à la fois, sauf pour le plan X Y pour lequel les axes sont « synchronisés ».

Il n’y a que les axes X et Y qui sont synchronisés pour plusieurs raisons, la première est que ca demande un morceau de programme assez complexe (pour moi en tout cas), la deuxième raison vient des limites du CPC, L’Amstrad a du mal à suivre, et on vient vite a être limité au niveau des impulsions qu’il peut envoyer aux cartes de contrôle d’axes sur le port imprimante.

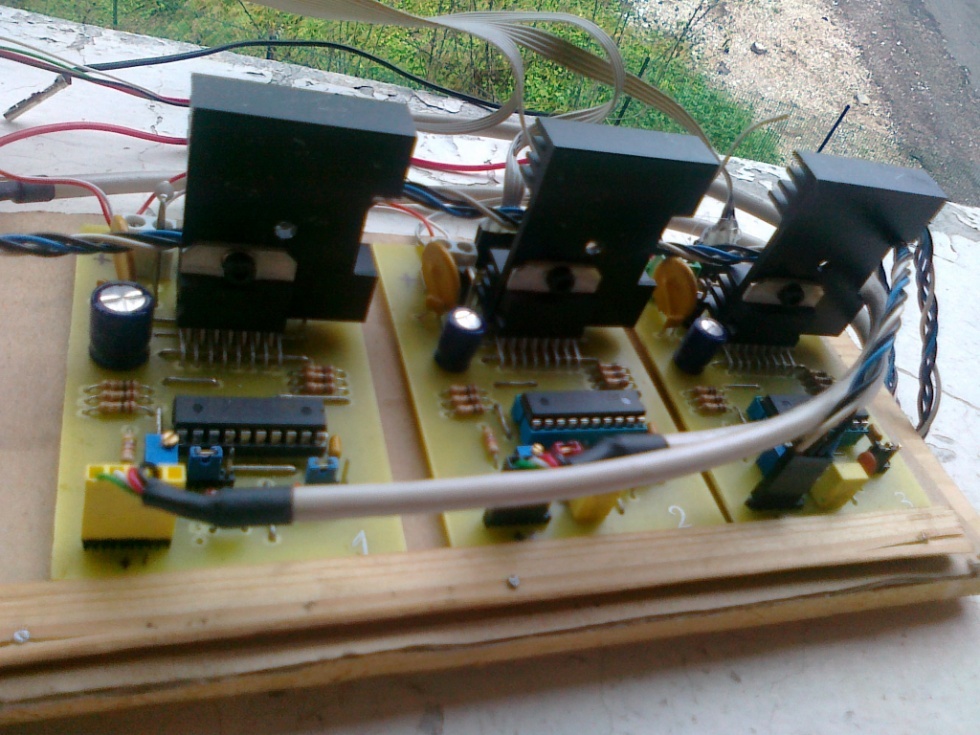
**Liste du matériel :**

* Un CPC 6128 (bin ouais normal)
* Une carte d’interface (made in by me )



* 3 cartes de contrôles de moteur pas à pas (plans provenant du site <http://otocoup.com/> )

Dossier de la carte : <http://cncloisirs.com/fichiers/Construction.Cartes%c0R%e9aliser/L298_card.zip>



* Du temps et de la patience

**Principes de déplacements**

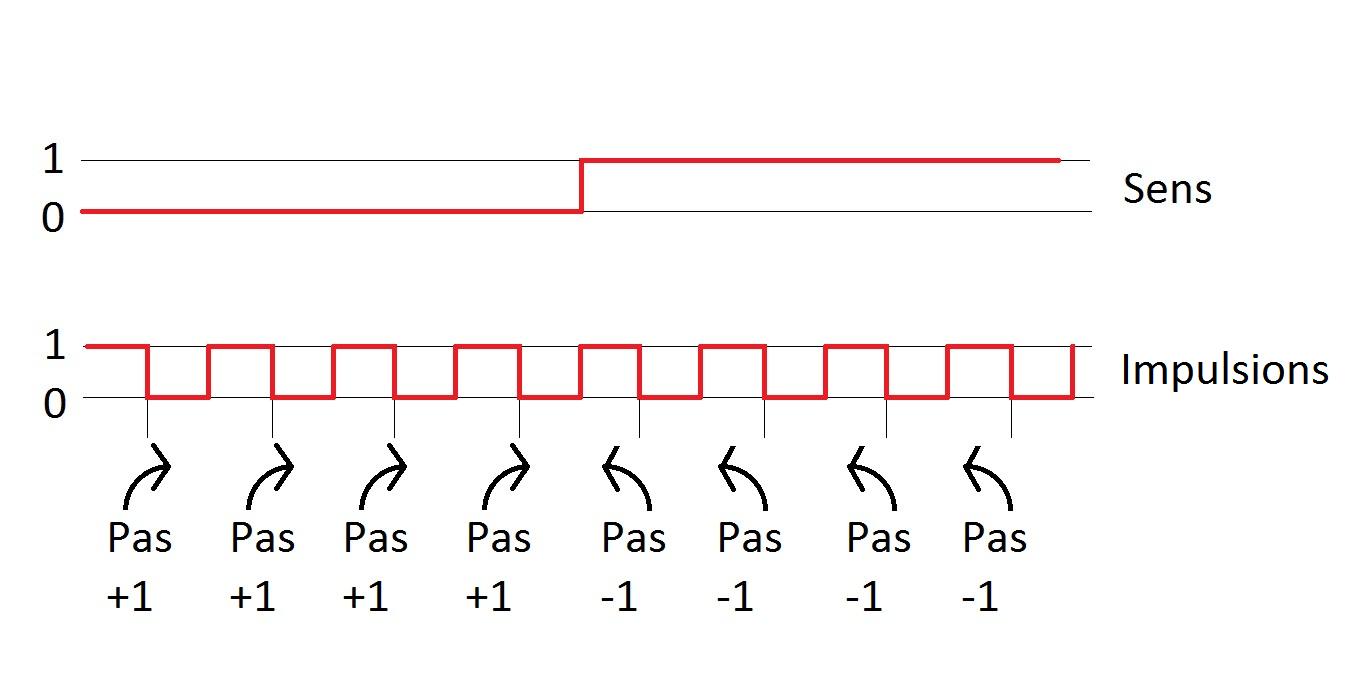
**Principe de pilotage d’une carte de contrôle de moteur pas à pas :**

Sur une carte on n’utilise que 4 des 5 pin présent pour la connectique,

* La masse
* Le +5v
* La broche « impulsions »
* La broche « sens »

Donc pas de problèmes pour les deux premier, c’est juste l’alimentation de la carte.

Les deux suivantes sont celles qui seront utilisé pour contrôler le pas et le sens d’un moteur. La carte se base sur l’état 0 des entrées, petite explication.



Pour des mouvements suivant un seul axe c’est assez simple, il suffit donc de piloter les 2 bits associés à un axe pour piloter un sont mouvement.

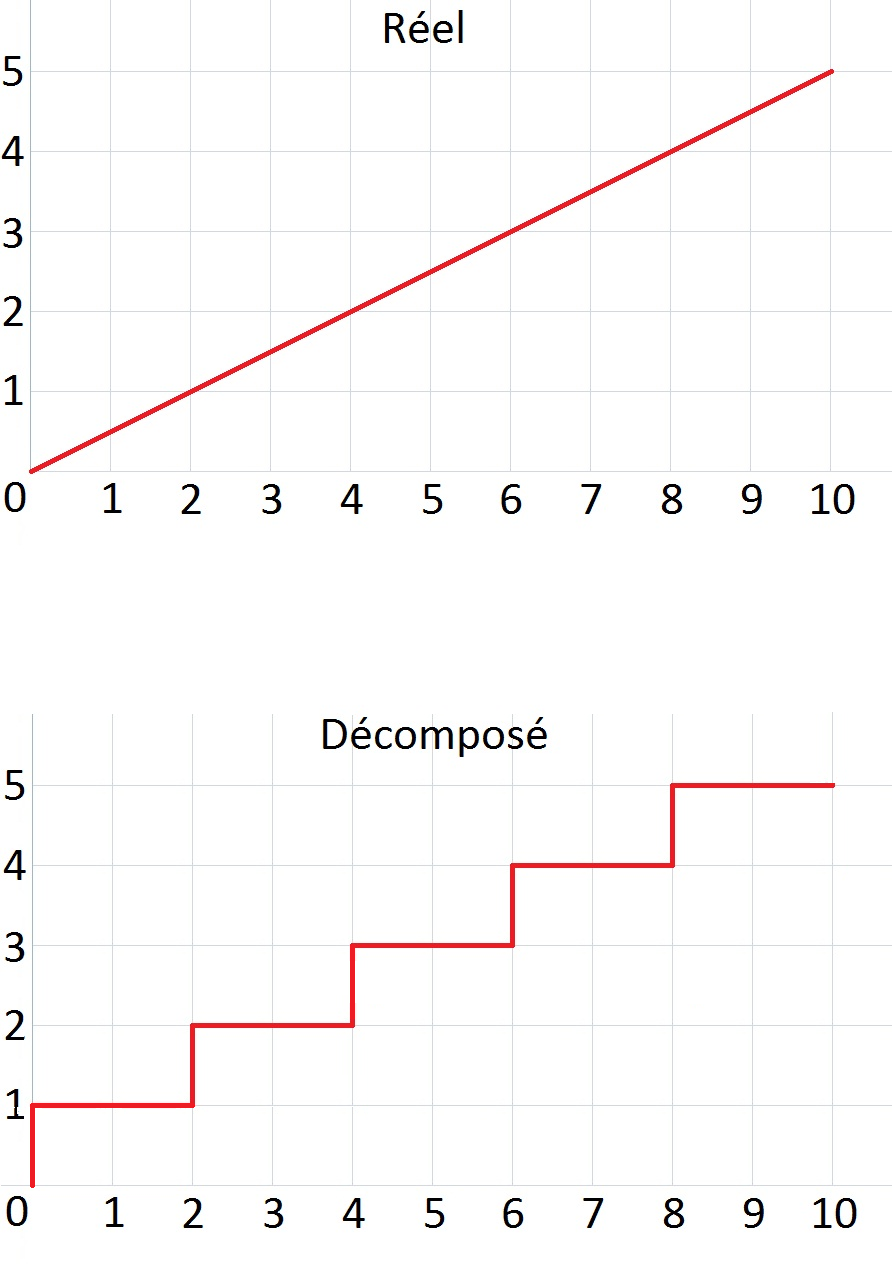
Passons maintenant à un mouvement en diagonale suivant les axes X et Y.

Etant donné que l’on utilise des cartes de contrôles basic et un Amstrad pour les piloter on ne peut pas réellement piloter 2 axes simultanément (en tout cas pour ma part)

Mon astuce a donc été de décomposer les diagonales en « marche d’escalier »

Petit exemple concret :

* Premier cas simple, imaginons que le robot se trouve à la position (0,0) suivant (X,Y) et que l’on veux une diagonale de 10 pas suivant X et de 5 suivant Y se qui donne (10,5)



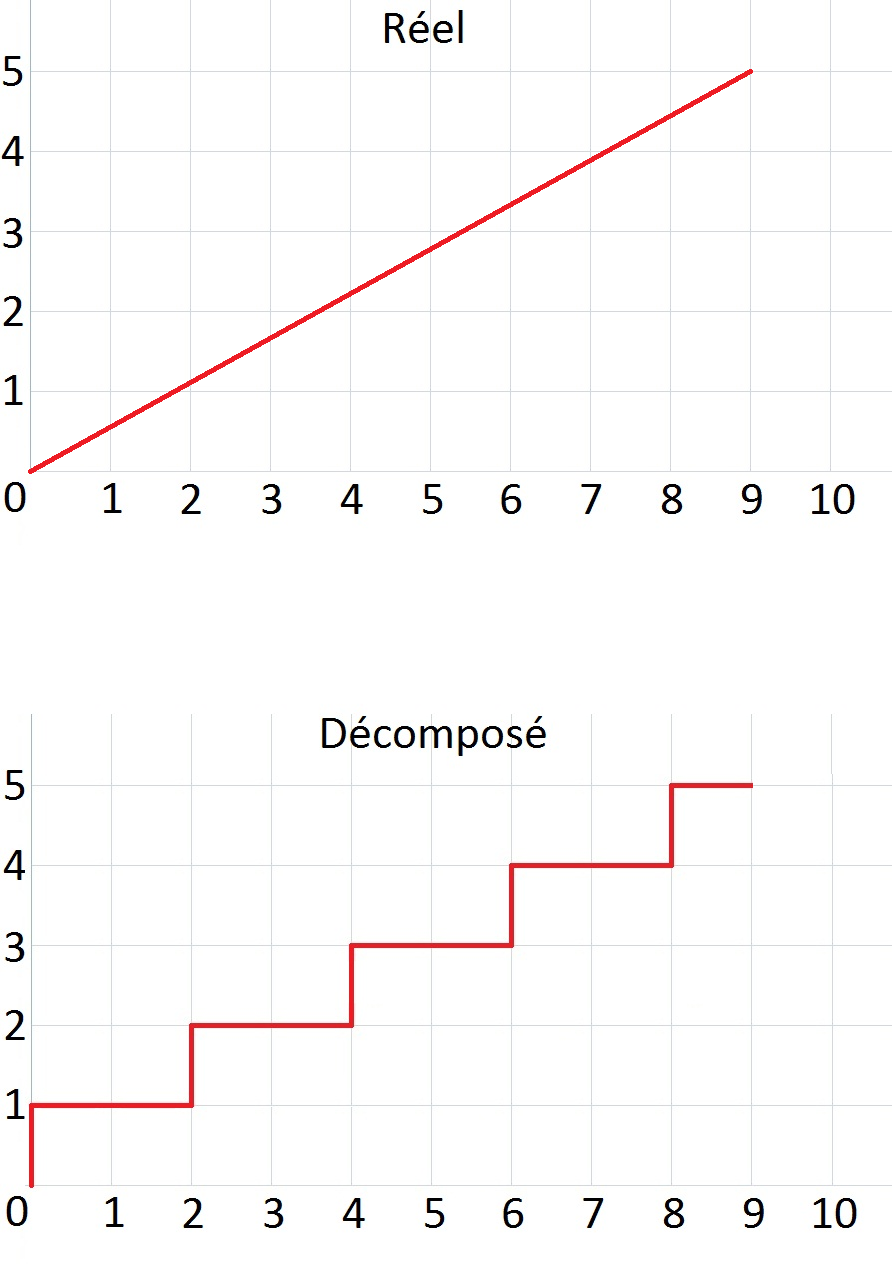
Le mouvement sera décomposer en échelons comme ci-dessus et le programme commandera 1 suivant Y puis 2 pas suivant X jusqu’a arriver à la position de destination.

La détermination des pas X et Y est ici simple, le programme prend la plus grande des valeurs (ici X avec 10) et la divise par la plus petite (ici Y avec 5)

On obtient 10/5=2

Le programme n’as plus qu’a effectué une un déplacement de 1 suivant Y puis de 2 suivant X et de renouveler ces 2 mouvements 5 fois pour se retrouver en (10,5).

* Deuxième cas plus complexe, ici on fait un déplacement de 9 suivant X et de 5 suivant Y (9,5), on part toujours de (0,0), dans ce cas la division du plus grand déplacement par le plus petit donne 9/5=1.8 on obtient un résultat à virgule. Voila comment le programme traite ce genre de cas.



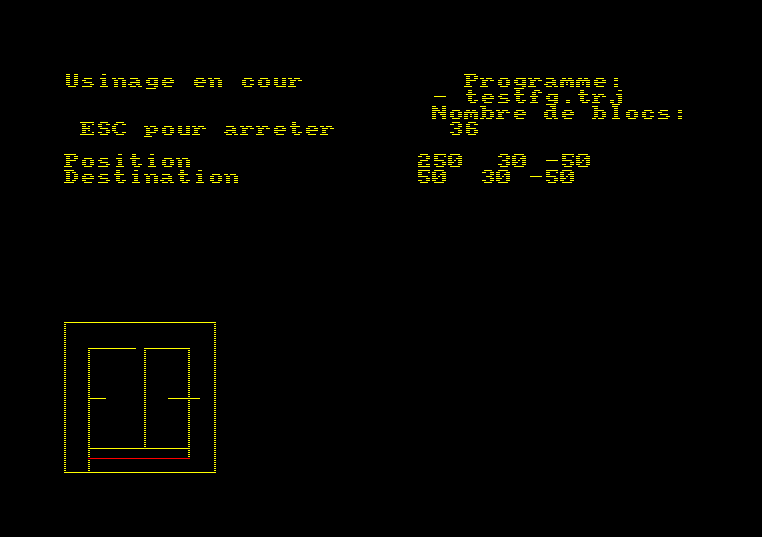
Le mouvement suit la même logique que l’exemple précédent, sauf que pour le dernier mouvement en X et le dernier en Y le programme connait la position du robot et se débrouille juste pour arriver aux valeurs de destinations (9,5)

**Présentation du programme :**

**Fonctions réalisées :**

Dans l’état actuel du programme il est possible de commander en manuel chacun des axes à tours de rôle, de charger un fichier contenant une série de coordonnées de déplacements et de lancer l’exécution des déplacements définit par le fichier trajectoire. Un dessin apparait alors, les traits jaunes représentent les déplacements déjà réalisés, le trait rouge le déplacements en cour.

Il n’est pas nécessaire d’avoir la partie hardware pour le développement du programme vu qu’il n’y a aucun retour d’information récupérer par l’Amstrad, il et donc tout a fait possible de lancer l’usinage « juste » pour vérifier que le dessin qui est tracé correspond bien a ce que l’on veut faire, un mode « simulation » est prévu par la suite.

****

Ouais je sais y a encore du boulot sur l’interface

**Adressage du port imprimante :**

Adressage du port imprimante sur l’Amstrad se fait grâce à la commande &EF00,valeur.

Dans mon programme pour visualiser l’état du port j’utilise « valeur » en binaire, pour cela on rajoute &X devant « valeur » pour que le basic sache que l’on « parle » en binaire.

Petit exemple de programme pour un mouvement (théorique)

10 For x=0 to 200

20 out &EF00,&X10

30 out &EF00,&X11

40 next

Ce petit programme correspond à une rotation dans le sens négatif ci on regarde sur le premier dessin.

Pour les 2 premiers bits je n’ai pas rencontré de trop gros problèmes, c’est en rajoutant les bits des autres axes, autrement dis en agissent sur un mot « valeur » de 6 bits que quelques problèmes apparaissent, je suis pars moment obligé de mettre une remarque « ‘ » entre 2 ligne d’adressage du port, ici par exemple : 25 ‘

**Format des fichiers trajectoires :**

Le programme d’usinage ne se déverrouille qu’une fois qu’un fichier de trajectoire a été chargé à l’aide du menu prévu pour ca (version beta)

Les fichiers trajectoires sont générer à partir d’un autre programme plutôt orienté FAO/CAO (encore en développement)

Chaque fichier a l’extension .TRJ

Possède un bit de test pour vérifier lors du chargement que le fichier et bien compatible,

le nombre de ligne de trajectoire que contient le fichier

Et enfin la série de coordonnées au format PosX(l),PosY(l),PosY(l)

Exemple de format d’enregistrement : (programme croix.trj)

(Récupéré à l’aide de ManageDsk)

1

8

100,0,0

100,100,0

0,100,0

0,0,0

100,100,0

0,100,0

100,0,0

0,0,0

1111,0,0

**Inventaire du fichier cpcnc.dsk :**

L’image disque qui va avec cet article contient 5 fichiers :

-CPCNC.bas

Qui est le programme principal pour piloter la fraiseuse

- testfg.trj

Qui est le fichier trajectoire de test (celui utilisé sur la vidéo)

- croix.trj

Qui est un autre fichier trajectoire pour mètre en évidence le problème de temps du au calcul et à l’exécution pour les diagonales

- gentrj.bas

Qui est un programme basic pour enregistrer une liste de coordonnées dans un fichier au format.TRJ

- gencercl.bas

Qui sert a générer des trajectoires circulaires a partir d’un point de départ, une arrivée, un rayon et un sens. C’est un morceau que je dois encore travailler pour l’intégrer dans le programme principal. Une fois le programme lancé, les flèches déplace un point, « ENTER » pour valider, taper « c » pour le menu cercle…

**Conclusion**

Bon, à l’heure actuel les bases sont opérationnelles, pour la maquette de fraiseuse réalisé a partir de moteurs pas a pas de lecteurs DVD et de lecteur disquette le fonctionnement en ligne droite est correcte je dirais (comme sur la vidéo), par contre dé que l’on commence à faire des diagonales on réduit considérablement la vitesse d’avance de le machine (charger le fichier croix.trj pour avoir une idée), et c’est bien là qu’ai le problème, en utilisant des moteurs plus gros, donc plus adapté a faire de l’usinage j’obtiens des vitesses d’avances du robot de 1.4mm/min… faut pas être pressé, par contre je peux positionné mon axe à 1/400 mm près avec mes moteur de 400 pas et ma réduction vis sans fin de 1 tour/mm soit une tige fileté M6 :D



(Test de déplacements avec moteur plus costaud mais pas pressé)

Me voila arrivé à la fin de mon exposé, j’espère avoir été claire et avoir écrit un sujet intéressant. Si vous êtes intéressé je reste à vôtres disposition pour plus de détails.

Amicalement

James