Cédric DUVAL

Antonin CLAUS

IMA4

# Rapport de Projet

# P28 : Imprimante 3D multi-fils

Encadrants : Alexandre BOE

Antoine URQUIZAR

Année 2015/2016

# Introduction

Les imprimantes 3D ne permettent généralement que d'imprimer une seule couleur à la fois, ce qui limite le potentiel créatif des designers ainsi que la qualité des pièces imprimées.

Il existe plusieurs parades à ce problème, comme par exemple ajouter d'autres buses d'extrusion à son imprimante. Cette solution est assez couteuse puisqu'il faut ajouter une buse, un dispositif de chauffe et un moteur pour apporter du fil pour chaque couleur souhaitée. De plus, comme toutes les buses ne peuvent physiquement pas déposer de la matière au même endroit, cette solution génère de l'encombrement, ce qui limite la taille des pièces qu'on peut imprimer.

Une autre solution consiste à interrompre l'impression en cours de route pour changer le fil. Cette méthode ne permet généralement que de changer de fil entre chaque couche. De plus, cette méthode nécessite à l'utilisateur d'être présent au moment du ou des changement(s) de couleur.

Notre projet consiste donc à réaliser une machine permettant à une imprimante 3D classique d'imprimer des pièces multicolores. Notre machine s'inspire de la deuxième solution, en réalisant une bobine de fil spécialement conçue pour la pièce qu'on souhaite imprimer. Le principe est de concaténer des bouts de fils dont la longueur a été préalablement calculée pour qu'au moment où la buse imprime à un endroit, ce soit la couleur souhaitée qui soit extrudée.

Ainsi, la réalisation de notre machine se distingue en deux parties distinctes : la partie software avec l'interprétation du fichier 3D, le calcul de la longueur de chaque segment de fil et la gestion des moteurs, et la partie hardware avec le déplacement, la sélection, la découpe et la soudure des fils en dur.

# Partie logicielle

## Lecture du fichier gcode

Un fichier gcode est le résultat du slicing de la pièce voulue par un logiciel appelée slicer (Cura, Slc3r, Meshmixer...). Celui-ci coupe la pièce en couches et les met dans le gcode afin que l'imprimante connaisse les mouvements à faire pendant l'impression. On analyse ensuite ce gcode afin de récupérer les longueurs de fil et les couleurs correspondantes. Le slicer doit être configuré en multi-extrudeurs afin de simuler les différents couleurs et les changements de fils pendant l'impression.

Comme nous voulons une pièce de plusieurs couleurs, nous devons charger une pièce constituée de plusieurs parties, une pour chaque couleur. On spécifie ensuite dans le slicer quelle partie correspond à quelle couleur pour retrouver dans le gcode les bonnes parties.

Le gcode est constitué de plusieurs commandes comme par exemple bouger la buse, changer la température ou le fil, ou arrêter l'impression en cours.

Les commandes qui vont nous intéresser ici sont :

- TX avec X le numéro de l'extrudeur. Cette commande indique le changement d'extrudeur et donc le changement de fil pour notre machine. Nous devons donc trouver tous les changements de fil du gcode afin de mettre le fil correspondant pour la suite de l'impression.

- EYY avec YY la longueur de fil demandée pour l'impression. Cette ligne de code commence avec YY très petit puis augmente au fur et à mesure de l'impression. Nous devons donc garder seulement la dernière ligne de code contenant E avant un changement d'extrudeur. Pour se faire, on repère le changement d'extrudeur dans le code, et si la valeur lue après ce changement est inférieure à la valeur avant le changement, alors la valeur avant le changement est la longueur associée au fil précédent.

Avec le programme conçu pour extraire ces informations du gcode, nous obtenons un fichier texte contenant exclusivement le changement d'extrudeur

et sa longueur associée. Nous chargeons ensuite ce fichier dans la carte SD qui sera connectée à l'Arduino.

# Arduino

## Shield Arduino

Le shield de la carte SD est connecté à l'Arduino afin qu'il puisse interpréter le fichier texte. L'arduino va lire les données du fichier et les interpréter afin de régler les moteurs sur les bonnes configurations et obtenir les bonnes longueurs pour chaque fil.

On recherche en parcourant le fichier les lettres clés T et E comme pour le gcode et on range les valeurs qui suivent ces lettres dans des variables afin de pouvoir les utiliser dans le code arduino. On parcoure donc le fichier au fur et à mesure que le fil est créé.

## Drivers et paramétrage des moteurs

Les moteurs sont connectés à des drivers. Ceux-ci ont 3 fils connectés à l'arduino pour la direction, les pas, et l'activation. Il y a donc 15 sorties nécessaires sur l'arduino pour commander les 5 moteurs ainsi que l'alimentation 5V et la masse pour les alimenter.

Pour commander un moteur, il faut régler les 3 paramètres du driver : enable, direction et step.

- La broche enable sert à activer le moteur. Si enable est sur 1, le moteur est à l'arrêt et 0 pour qu'il soit en marche.

- La broche direction sert à changer le sens de déplacement du moteur.

- La broche step sert à faire avancer pas par pas le moteur. Par exemple, pour qu'un moteur avance d'un pas, on met la broche step à 1 puis à 0, avec une pause entre chaque changement pour ne pas surcharger l'arduino de commandes et régler la vitesse du moteur. Pour faire avancer le moteur d'un nombre x de pas, on crée une boucle for avec comme paramètre x qui incrémente.

Pour faire avancer plusieurs moteurs en même temps, il faut écrire sur les broches step de chaque moteur les uns après les autres. On crée donc une boucle tant que qui vérifie si les moteurs ont vérifié leur condition, puis à l'intérieur, une

boucle if pour chaque moteur qui écrit sur la broche step afin que le moteur bouge d'un pas si le moteur n'a pas encore atteint son objectif.

## Sélection du fil

On crée 2 variables dans l'arduino pour le fil sélectionné et sa longueur. Une fois qu'on sait quel fil est à extruder en lisant le fichier issu du gcode, on tourne l'arbre à cames de façon à ce que ce fil puisse passer. Les fils sont numérotés de 0 à 3. Les moteurs sont réglés sur 400 pas, et les cames sont séparés de 90°. Il faut donc demander au moteur d'avancer de 100 pas si l'arbre est actuellement sur le fil 0 et que la variable fil est à 1.

Une fois le bon fil sélectionné, on le fait avancer avec le moteur relié au rouleau qui entraine tous les fils, ainsi que ceux des extrudeurs à la sortie de l'entonnoir. Avec la valeur issue du gcode qui nous donne la longueur de fil voulue, on fait avancer les 3 moteurs de la bonne distance. La distance change avec le firmware de l'imprimante pendant le slicing, il faut donc calculer la distance exacte que représente cette valeur, puis la comparer avec les distances des moteurs afin de prendre une valeur exacte.

## Découpe du fil

On coupe ensuite le fil avec le moteur commandant la découpeuse. On déplace celle-ci jusqu'à ce qu'elle soit au-dessus du fil, puis on désactive le moteur, entrainant la chute de la découpeuse et la section du fil. On sépare ensuite les 2 fils découpés afin qu'il ne fonde pas et qu'ils ne se ressoudent pas entre eux après la relève de la découpeuse.

On peut ensuite ramener un autre fil afin de le souder au premier.

## Soudure des fils

La phase de soudure est similaire à la phase découpe. La découpeuse se positionne entre les 2 fils puis ceux-ci sont pressés contre le fil chauffant afin qu'ils soient tous les 2 à la même température pour une meilleure soudure. On relève ensuite la découpeuse puis les 2 fils sont cette fois pressés l'un contre l'autre puis tout de suite emmenés vers l'extrudeur de sortie afin que la soudure soit cylindrique pour qu'elle puisse passer dans l'imprimante 3D.

# Partie mécanique

Pour satisfaire la mention du cahier des charges qui spécifiait un prix de fabrication faible et notamment un nombre de moteur non linéaire au nombre de fils utilisés, nous avons dû concevoir un système de sélection et de déplacement des fils assez complexe.

## Déplacements des fils

En ce qui concerne le déplacement, nous entrainons tous les fils en même temps, par friction, à l'aide d'un rouleau en caoutchouc dur récupéré dans une imprimante papier. Le rouleau est pressé contre les fils avec des pièces imprimées en 3D et des ressorts et il est mis en rotation grâce à un moteur pas à pas. (Voir annexe1)

## Sélection du fil

Comme on ne souhaite acheminer qu'un seul fil à la fois, il faut bloquer tous les fils sauf un. Pour cela, nous avons réalisé un arbre à cames qui permet, grâce à un système de piston, de presser trois des quatre fils. L'arbre à cames transforme le mouvement de rotation du moteur pas à pas en un mouvement de translation alterné. En positionnant le moteur correctement, on parvient à freiner les fils que l'on ne veut pas entrainer. Ainsi, nous somme parvenu à gérer quatre fils avec seulement deux moteurs. De plus, on pourrait étendre ce procédé à davantage de fils en ajoutant plus de pistons et en décalant les cames du bon angle. (Voir annexe2)

## Orientation des fils

Pour que les fils, insérés à des endroits différents débouchent à la même zone de découpe/ soudure, nous avons réalisé un « entonnoir à fil ». Nous avions initialement prévu de le réaliser en impression 3D, seulement, du fait de sa taille importante (20cm/20cm) et de sa complexité (« tunnel » à l'intérieur) (Voir

annexe 3), son impression n'a pas été possible avec les moyens du Fabricarium. Nous l'avons donc réalisé avec une découpeuse laser. Pour cela nous avons gravé les chemins qu'emprunteront les fils en suivant des courbes de Bézier pour éviter que les fils ne soient trop tordus, ne se cassent ou se plantent dans le bois (Voir annexe 4).

## Découpe

La découpe du fil est réalisée à l'aide d'une lame résistive tendue sur une fourche en bois. En appliquant une tension à cette lame, elle chauffe, ce qui permet de couper facilement le plastique. La découpe est gérée grâce à un moteur pas à pas positionné à la sortie de l'entonnoir. (Voir annexe 5)

## Soudure

La soudure est effectuée avec ce même matériel : la lame chauffée est maintenue précisément entre les deux fils à souder. La chaleur fait fondre les deux fils qui se soudent l'un à l'autre lorsqu'un vient les presser l'un sur l'autre. C'est pour pouvoir positionner et maintenir la lame bien entre les deux fils qu'il est nécessaire d'utiliser un moteur pas à pas.

# Conclusion

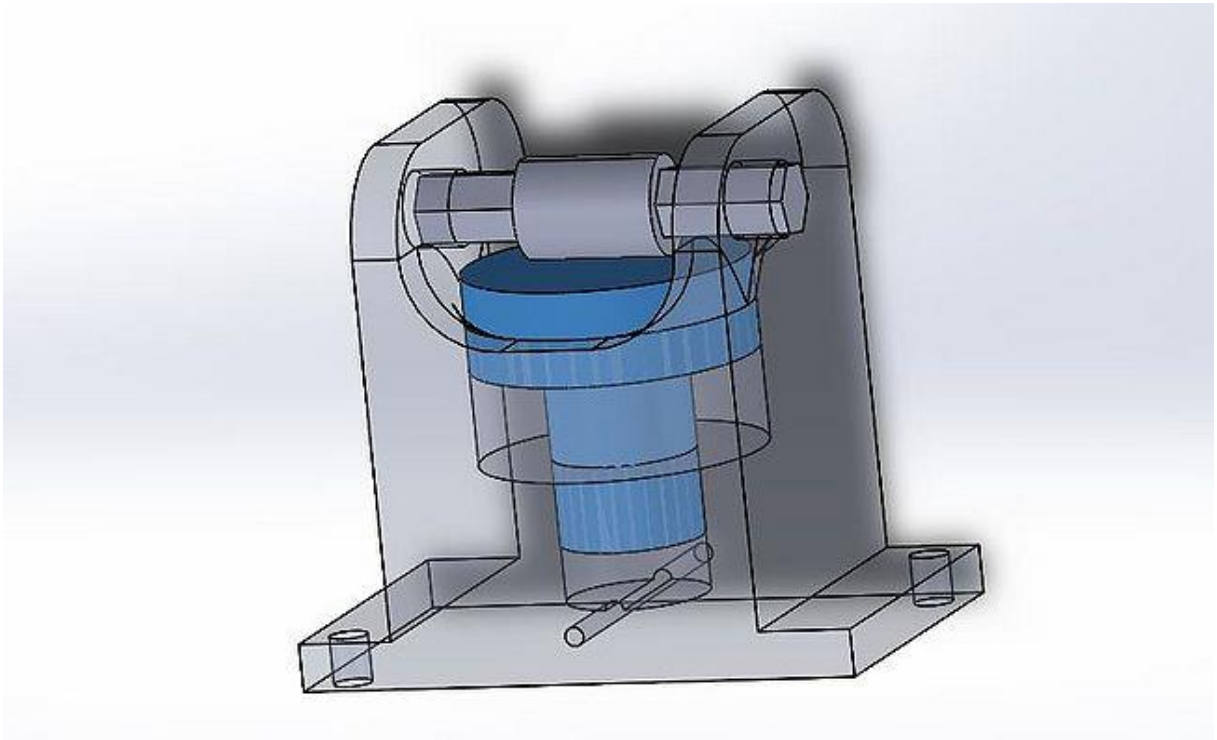
Ce projet nous a permis de réaliser un prototype qui prouve la faisabilité de notre méthode. Cependant, il nous a manqué un peu de temps pour finaliser l'automatisation du procédé. Pour ce faire, nous aurions implémentés des capteurs de présence au niveau de l'entrée de chaque fils dans l'entonnoir et au niveau de la zone de soudure (optique TOR, galet, ...).

Pour augmenter la fiabilité du procédé, on pourra ajouter un capteur de couleur au niveau de la buse d'impression de l'imprimante 3D, on modifiera ensuite le programme pour 'surdoser' la quantité de fil à chaque changement de couleur. Il suffira alors d'extruder l'excédent de plastique jusqu'à obtenir la couleur souhaitée et on évite ainsi tout risque de décalage des couleurs ainsi que les problèmes de couleur mélangée lors des changements.

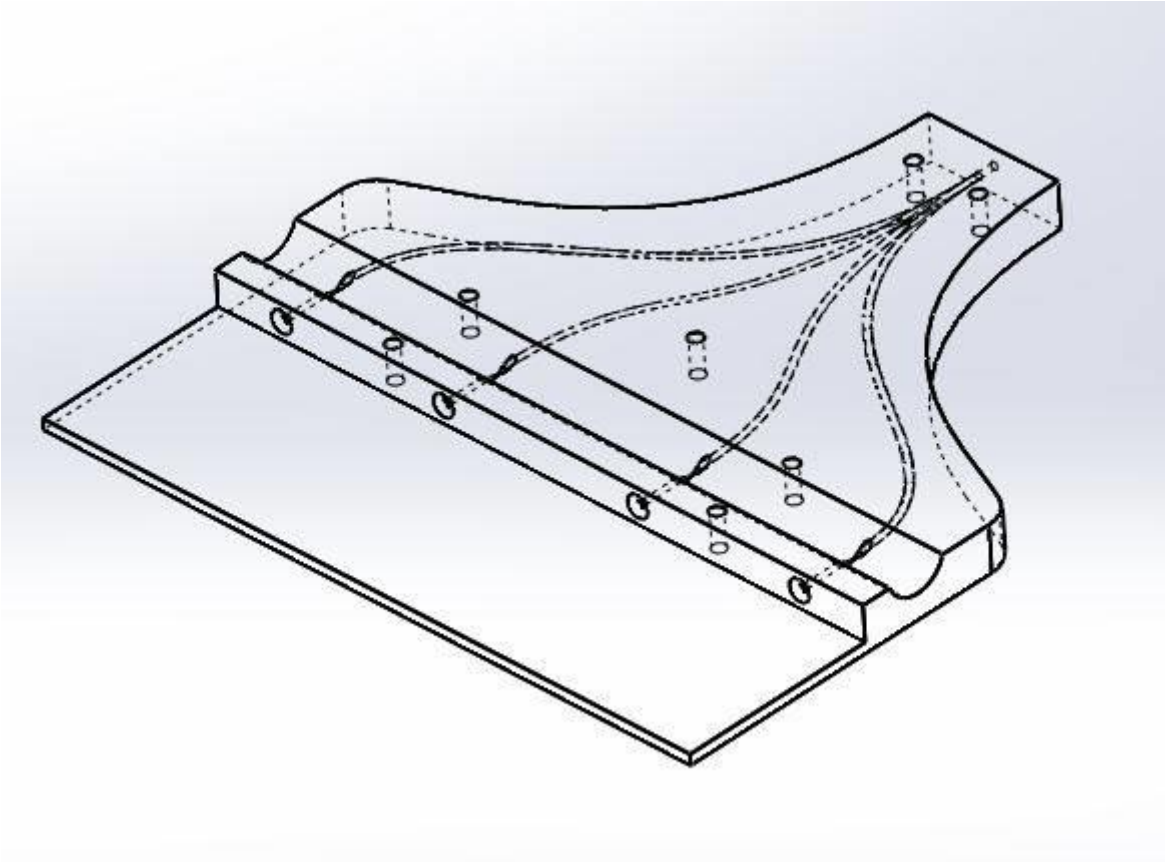
# Annexes:

Annexe 1 :

Annexe 2 :

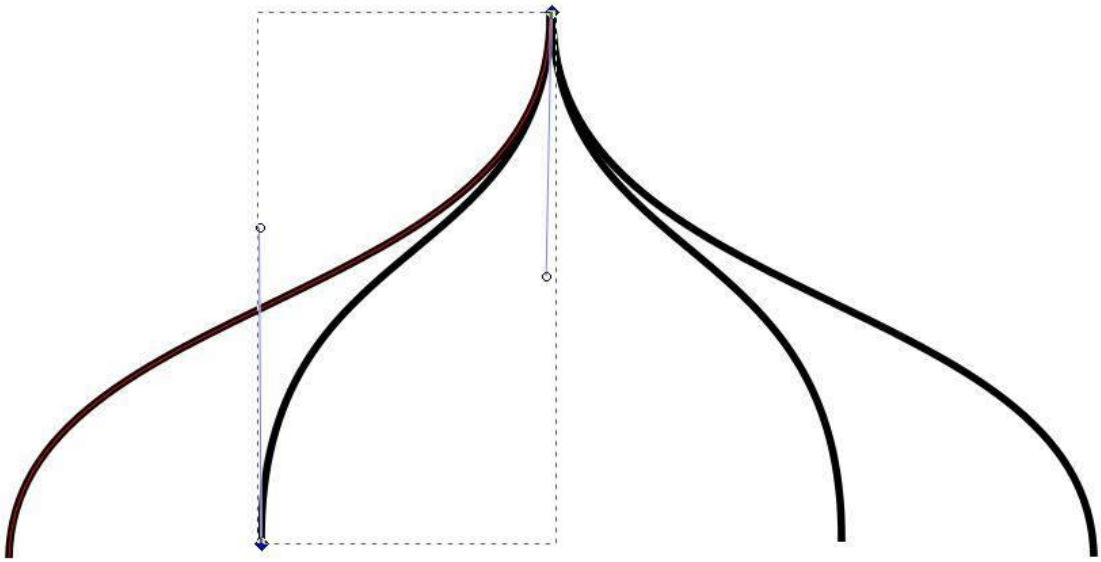


Annexe 3 :

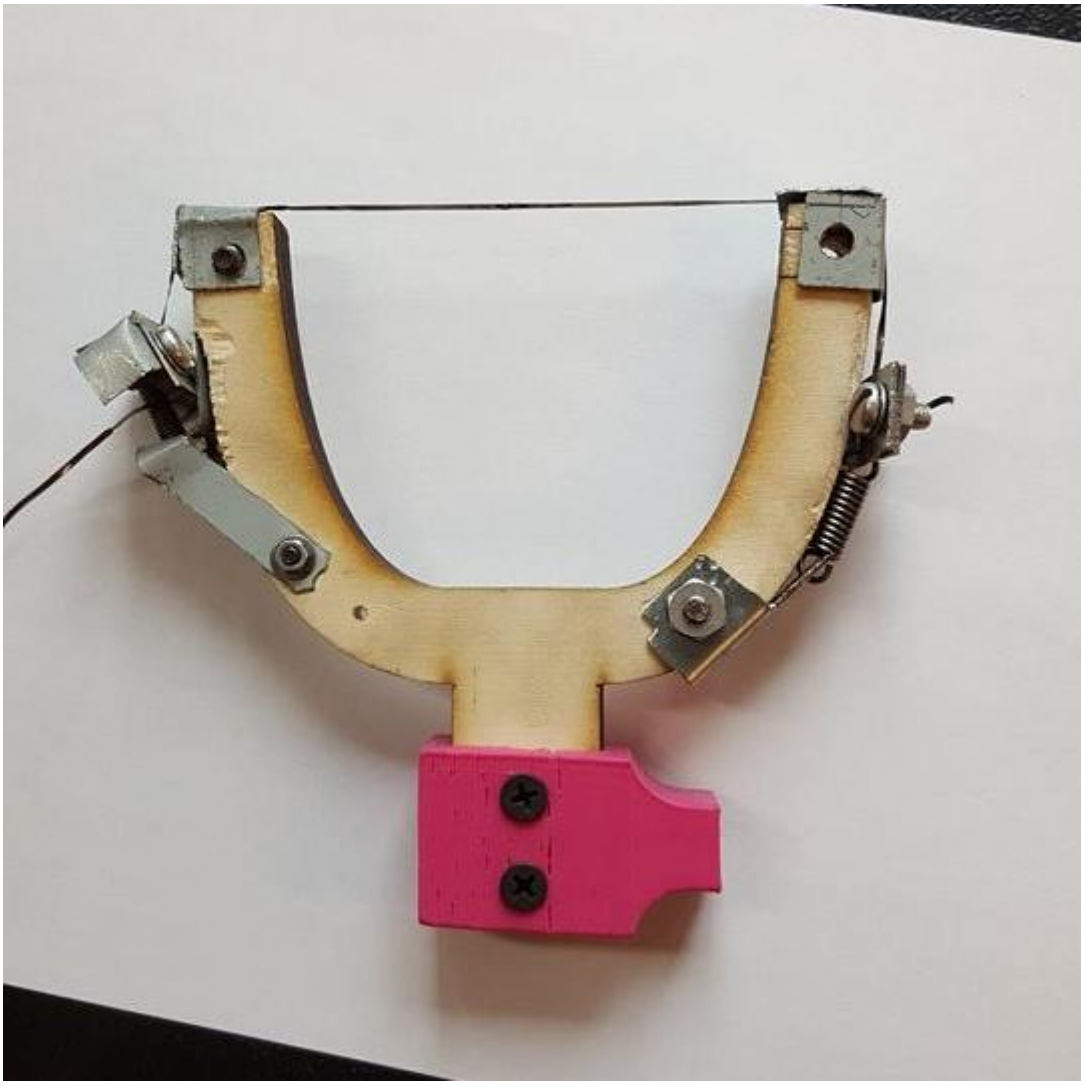


Annexe 4 :

Les deux Vecteur aux extrémités permettent de régler le rayon de courbure et l'angle d'incidence du trait. Les deux autres courbes sont créés par symétrie.



Annexe 5 :



gcode.c :

```
#include<stdio.h>
#include<stdbool.h>
#include<stdlib.h>
#include<string.h>
#define TAILLE 50

double conversion(char tab []){
return atof(tab);
}

int main(int argc, char *argv[])
{
  FILE* fichierR = NULL;
  FILE* fichierW = NULL;
  int caractereActuel = 0;
  int changement =0;
  char ligne1[TAILLE]="1";
  char ligne0[TAILLE]="1";
  char tab_char_nb[10];
  double longueur_fil1=0;
```

```
double longueur_fil0=0;
double longueur_fil_finale=0;
int new_extrudeur=0;
int old_extrudeur=0;
int fil_trouve=0;
char cmd[2];
int i=0,k=0,j=0;
fichierR = fopen("frog2.gcode", "r");
fichierW = fopen("gcodeinterprete.txt", "w");
if (fichierR != NULL)
  {
    if (fichierW != NULL)
   {

    while (caractereActuel != EOF && (ligne1[5]!='G'))
    {
      i=0;

      do{
        caractereActuel = fgetc(fichierR);
        ligne1[i]=caractereActuel;
        i++;
      }while ( caractereActuel != EOF && caractereActuel != '\n' );

      ligne1[i]='\0';

      if (ligne1[0]=='T' && ligne1[1]=='0')
            {
             old_extrudeur=new_extrudeur;
             new_extrudeur=0;
            }
      if (ligne1[0]=='T' && ligne1[1]=='1')
            {
             old_extrudeur=new_extrudeur;
             new_extrudeur=1;
            }
      if (ligne1[0]=='T' && ligne1[1]=='2')
            {
             old_extrudeur=new_extrudeur;
             new_extrudeur=2;
            }
      if (ligne1[0]=='T' && ligne1[1]=='3')
            {
             old_extrudeur=new_extrudeur;
             new_extrudeur=3;
            }
      if(ligne1[0] == 'G' && ligne1[1] == '1')
      {

        for (j=12; j < sizeof(ligne1);j++)
        {
           if(ligne1[j]== 'E' && ligne1[j+1]!='-'){
            j++;
            k=0;
            for (j; j < sizeof(ligne1); j++)
            {
              tab_char_nb[k]=ligne1[j];
              k++;
            }
           }
        }
```

```
        }
        longueur_fil1=atof (tab_char_nb);
         if(longueur_fil1<longueur_fil0)
         {
          printf("T%d\n",old_extrudeur );
          printf("E%f\n",longueur_fil0 );
          fprintf(fichierW,"T%d\n",old_extrudeur );
          fprintf(fichierW,"E%f\n",longueur_fil0 );
         }
         if(ligne1[0]==';' &&ligne1[1]=='E' &&ligne1[2]=='n'
&&ligne1[3]=='d')
         {
          printf("T%d\n",new_extrudeur);
          printf("E%f\n",longueur_fil0);
          fprintf(fichierW,"T%d\n",new_extrudeur);
          fprintf(fichierW,"E%f",longueur_fil0);
          longueur_fil0=0;
         }
         longueur_fil0=longueur_fil1;
      }

    fclose(fichierW);

    }
    fclose(fichierR);
  }




return 0;

}
```

## Code arduino :

```
#include <SPI.h>
#include <SD.h>

//variables globales
int caractereActuel = 0; //caractere lu
char tab_char_nb[15]; // tableau longueur fil
File fichierSD; // fichier lu

double longueur_fil; // longueur du fil

//moteur(H,400,Came_pinEnable,Came_pinDir,Came_pinStep);
#define Came_pinEnable 34 // Activation du driver/pilote
#define Came_pinStep   32 // Signal de PAS (avancement)
#define Came_pinDir    30// Direction

//moteur(H,400,Decoupe_pinEnable,Decoupe_pinDir,Decoupe_pinStep);
#define Decoupe_pinEnable 50// Activation du driver/pilote
#define Decoupe_pinStep   48// Signal de PAS (avancement)
#define Decoupe_pinDir    46// Direction

//moteur(H,400,Entrainement_pinEnable,Entrainement_pinDir,Entrainement_pinS
tep);
#define Entrainement_pinEnable 22// Activation du driver/pilote
#define Entrainement_pinStep    24 // Signal de PAS (avancement)
#define Entrainement_pinDir     26// Direction

//moteur(H,400,Extruentree_pinEnable,Extruentree_pinDir,Extruentree_pinStep
);
#define Extruentree_pinEnable 7 // Activation du driver/pilote
#define Extruentree_pinStep    6// Signal de PAS (avancement)
#define Extruentree_pinDir     5 // Direction


//moteur(H,400,Extrusortie_pinEnable,Extrusortie_pinDir,Extrusortie_pinStep
);
#define Extrusortie_pinEnable 10 // Activation du driver/pilote
#define Extrusortie_pinStep    9 // Signal de PAS (avancement)
#define Extrusortie_pinDir     8 // Direction

#define H HIGH
#define L LOW
#define D 20


////////////////////////////////////////////////////////////////
//////////////FONCTION LECTURE CARTE SD/////////////////////////
////////////////////////////////////////////////////////////////

int fonction_numero_fil(int fil) {
  fil = fichierSD.read();
  delay( 10 );
  fil = fil - 48;
  Serial.print(F("fil n "));
  Serial.println(fil);
  return fil;
}

void fonction_longueur_fil() {
  int k;
```

```
  k = 0;
  while (caractereActuel != 13) {
    caractereActuel = fichierSD.read();
    delay( 10 );
    tab_char_nb[k] = caractereActuel;
    k++;
  }
  longueur_fil = atof (tab_char_nb);
  Serial.print(F("longueur fil :" ));
  Serial.println(longueur_fil);
}
////////////////////////////////////////////////////////////////
////////////////////FONCTION MOTEUR ///////////////////////////
////////////////////////////////////////////////////////////////

void moteur(char dir, int longueur, int pinEnable, int pinDir, int pinStep)
{
  digitalWrite(pinDir , dir);
  digitalWrite( pinStep  , LOW);  // Initialisation de la broche step
  // Avance de longueur pas
  for (int i = 0; i < longueur; i++)
  {
    digitalWrite( pinStep, HIGH );
    delay( 8 );
    digitalWrite( pinStep, LOW );
    delay( 8 );
  }

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable);
  Serial.print(" de ");
  Serial.print(longueur);
  Serial.println("cm");
  delay( 1000 );

}

void moteurlent(char dir, int longueur, int pinEnable, int pinDir, int
pinStep) {
  digitalWrite(pinDir , dir);
  digitalWrite( pinStep  , LOW);  // Initialisation de la broche step
  // Avance de longueur pas
  for (int i = 0; i < longueur; i++)
  {
    digitalWrite( pinStep, HIGH );
    delay( 10 );
    digitalWrite( pinStep, LOW );
    delay( 10 );
    delay(250);

  }

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable);
  Serial.print(" de ");
  Serial.print(longueur);
  Serial.println("cm");
  delay( 1000 );

}
```

```
void deuxmoteurs(char dir1, int longueur1, int pinEnable1, int pinDir1, int
pinStep1, char dir2, int longueur2, int pinEnable2, int pinDir2, int pinStep2)
{
  int i1 = 0, i2 = 0;
  digitalWrite(pinDir1 , dir1);
  digitalWrite(pinDir2 , dir2);
  digitalWrite( pinStep1  , LOW);
  digitalWrite( pinStep2  , LOW);// Initialisation de la broche step
  // Avance de longueur pas
  while (i1 < longueur1 && i2 < longueur2)
  {
    if (i1 < longueur1) {
      digitalWrite( pinStep1, HIGH );
      delay( 1 );
      digitalWrite( pinStep1, LOW );
      delay( 1 );
      i1++;
    }
    if (i2 < longueur2) {
      digitalWrite( pinStep2, HIGH );
      delay( 1 );
      digitalWrite( pinStep2, LOW );
      delay( 1 );
      i2++;
    }
  }

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable1);
  Serial.print(" de ");
  Serial.print(longueur1);
  Serial.println("cm");

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable2);
  Serial.print(" de ");
  Serial.print(longueur2);
  Serial.println("cm");
}

void troismoteurs(char dir1, int longueur1, int pinEnable1, int pinDir1, int
pinStep1, char dir2, int longueur2, int pinEnable2, int pinDir2, int pinStep2,
char dir3, int longueur3, int pinEnable3, int pinDir3, int pinStep3)
{
  int i1 = 0, i2 = 0, i3 = 0;
  digitalWrite(pinDir1 , dir1);
  digitalWrite(pinDir2 , dir2);
  digitalWrite(pinDir3 , H);
  digitalWrite( pinStep1  , LOW);
  digitalWrite( pinStep2  , LOW);// Initialisation de la broche step
  digitalWrite( pinStep3  , LOW);
  // Avance de longueur pas
  while (i1 < longueur1 && i2 < longueur2 && i3 < longueur3)
  {
    if (i1 < longueur1) {
      digitalWrite( pinStep1, HIGH );
      delay( 1 );
      digitalWrite( pinStep1, LOW );
      delay( 1 );
      i1++;
    }
```

```
    delay( 1 );
    if (i2 < longueur2) {
      digitalWrite( pinStep2, HIGH );
      delay( 1);
      digitalWrite( pinStep2, LOW );
      delay( 1 );
      i2++;
    }
    delay( 1 );
    if (i3 < longueur3) {
      digitalWrite( pinStep3, HIGH );
      delay( 1);
      digitalWrite( pinStep3, LOW );
      delay( 1);
      i3++;
    }
    delay( 10 );
  }

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable1);
  Serial.print(" de ");
  Serial.print(longueur1);
  Serial.println("cm");

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable2);
  Serial.print(" de ");
  Serial.print(longueur2);
  Serial.println("cm");

  Serial.print("Avancée du moteur ");
  Serial.print(pinEnable3);
  Serial.print(" de ");
  Serial.print(longueur3);
  Serial.println("cm");
}

////////////////////////////////////////////////////////////////
//////////////////////////FONCTIONS ELEMENTAIRES ///////////////
////////////////////////////////////////////////////////////////

void entrainement_fil3(int longueur) {
  Serial.println("entrainement du fil");
  troismoteurs(H, longueur, Entrainement_pinEnable, L, Entrainement_pinStep,
H, longueur, Extruentree_pinEnable, Extruentree_pinDir, Extruentree_pinStep,
H,        longueur,        Extrusortie_pinEnable,        Extrusortie_pinDir,
Extrusortie_pinStep);

}

void entrainement_fil2(int longueur) {
  Serial.println("entrainement du fil");
  deuxmoteurs(H,  longueur,  Entrainement_pinEnable,  Entrainement_pinDir,
Entrainement_pinStep,       H,       longueur,       Extruentree_pinEnable,
Extruentree_pinDir, Extruentree_pinStep);
}

void decoupe() {
  //descente de la decoupe
  Serial.println("Decoupe");
```

```
  moteur(H, 165, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);
  // Changer de direction remontée de la decoupe

  digitalWrite( Decoupe_pinEnable, HIGH );
  delay(2000);
  //for (int i = 0; i < 7; i++)
  //   moteurlent(H, 1, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);

  moteur(L,       25,       Extruentree_pinEnable,       Extruentree_pinDir,
Extruentree_pinStep);
  digitalWrite( Decoupe_pinEnable, LOW );
  moteur(L, 180, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);

  //     moteur(L,     15,     Extruentree_pinEnable,     Extruentree_pinDir,
Extruentree_pinStep);


}

void soudure() {
  Serial.println("Soudure");
  //fils s'eloignent
  moteur(L,       30,       Extruentree_pinEnable,       Extruentree_pinDir,
Extruentree_pinStep);
  moteur(H,       40,       Extrusortie_pinEnable,       Extrusortie_pinDir,
Extrusortie_pinStep);
  //descente
  moteur(H, 160, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);
  digitalWrite( Decoupe_pinEnable, HIGH );
  delay(2000);
  digitalWrite( Decoupe_pinEnable, LOW );
  // Changer de direction remontée de la decoupe
  for (int i = 0; i < 3; i++)
    moteurlent(L, 1, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);
  //moteurlent(L, 30, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);

  //fils se rapprochent et se soudent
  moteur(H,       50,       Extruentree_pinEnable,       Extruentree_pinDir,
Extruentree_pinStep);
  moteur(L,       46,       Extrusortie_pinEnable,       Extrusortie_pinDir,
Extrusortie_pinStep);
  delay(2000);


  //remontée de la soudure
  moteur(L, 30, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);
  //fil se presse contre l'autre
  moteur(H,       20,       Extruentree_pinEnable,       Extruentree_pinDir,
Extruentree_pinStep);

  entrainement_fil3(1600);

  //remontée 2 de la soudure
  moteur(L, 160, Decoupe_pinEnable, Decoupe_pinDir, Decoupe_pinStep);



}

void arbre_came(int position_actuelle, int position_voulue)
{
```

```
  Serial.print("position_actuelle : ");
  Serial.println(position_actuelle);
  Serial.print("position_voulue : ");
  Serial.println(position_voulue);
  while (position_actuelle != position_voulue)
  {


    moteur(H, 100, Came_pinEnable, Came_pinDir, Came_pinStep);
    position_actuelle++;

    if (position_actuelle == 4)
      position_actuelle = 0;
    Serial.println(position_actuelle);
    Serial.println(position_voulue);
  }
  Serial.print("Arbre came mis sur position ");
  Serial.println(position_actuelle);
}

void setup() {
  Serial.begin(9600);

  pinMode( Came_pinEnable, OUTPUT );
  pinMode( Came_pinDir   , OUTPUT );
  pinMode( Came_pinStep  , OUTPUT );

  pinMode( Decoupe_pinEnable, OUTPUT );
  pinMode( Decoupe_pinDir   , OUTPUT );
  pinMode( Decoupe_pinStep  , OUTPUT );

  pinMode( Entrainement_pinEnable, OUTPUT );
  pinMode( Entrainement_pinDir   , OUTPUT );
  pinMode( Entrainement_pinStep  , OUTPUT );

  pinMode( Extruentree_pinEnable, OUTPUT );
  pinMode( Extruentree_pinDir   , OUTPUT );
  pinMode( Extruentree_pinStep  , OUTPUT );

  pinMode( Extrusortie_pinEnable, OUTPUT );
  pinMode( Extrusortie_pinDir   , OUTPUT );
  pinMode( Extrusortie_pinStep  , OUTPUT );
   //Initialisation de l'instance
  if (!SD.begin(4)) {
    Serial.println(F("Initialisation impossible !"));
    return;
  }
  Serial.println(F("Initialisation OK"));
  //Ouverture du fichier
  fichierSD = SD.open("FILE.txt", FILE_READ);
}

void loop() {

  int fil = 0; // numero du fil
  int longueur = 0;
  int position_actuelle = 0, position_voulue;
  if (fichierSD) {
    while (caractereActuel != 'T') {
      caractereActuel = fichierSD.read();
    }
```

```
    fil = fonction_numero_fil(fil);

    arbre_came(position_actuelle, fil);

    //entrainer le fil jusqu'a la decoupe
    entrainement_fil3(D);

    while (caractereActuel != 'E') {
      caractereActuel = fichierSD.read();
    }
    fonction_longueur_fil();
    //prendre longueur voulue du fil
    entrainement_fil3(longueur_fil);
    //couper
    decoupe();
    //retour fil qui reste en arriere jusqu'a l'arbre
    moteur(L,      D,      Entrainement_pinEnable,      Entrainement_pinDir,
Entrainement_pinStep);

    while (fichierSD.available()) {
      caractereActuel = fichierSD.read();
      delay( 10 );
      if (caractereActuel == 'T') {
        fil = fonction_numero_fil(fil);
        arbre_came(position_actuelle, fil);
        //entrainer le fil jusqu'a la decoupe
        entrainement_fil2(D);
        //soudure
        soudure();
      }
      if (caractereActuel == 'E') {
        fonction_longueur_fil();

        delay( 10 );
        //avancée des fils de longueur voulue
        entrainement_fil3(longueur_fil);
        //decoupe
        decoupe();
        //retour fil qui reste en arriere jusqu'a l'arbre
        moteur(L,     D,     Entrainement_pinEnable,     Entrainement_pinDir,
Entrainement_pinStep);
      }
    }
  }
}
```