```
/*Antoine Hebert
Janvier-Fevrier 2014
Ce programme génere des signaux de type sinus, triangle, sinus amorti
en vue de commander une table tracante */
//Version 6 17/02/2014

#include <math.h>

//Déclaration des pins
const int VY_Lissajous2=31;
const int VY_Lissajous1=29;
const int VY_SinAmorti=27;
const int VY_Triangle=25;
const int VY_Sinus=23;
const int Plume=43;
const int Depart=47;
const int VX_f_t=53;
const int VX_Triangle=51;
const int VX_Sinus=49;
const int Coeff[10]={41,39,37,35,33,32,34,36,38,40};
const int Phi=8;
//Voie X DAC0
//Voie Y DAC1

//Valeur des coefficients
const float coeffVal[10]={0,0.01,0.02,0.03,0.05,0.1,0.2,0.4,0.7,1};

//indicateur d'interruption  false:pas d'interruption en cours
//sert aussi à "bloquer" les interruption (une fonction vide est executée)
bool etatInterrupt=false;


//Fonction d'interruption
//cette fonction attend que l'utilisateur appuie su départ, et positionne le curseur
//sur la position qui va etre prise ensuite pour repartir


void interruptCoupParCoup(){
  volatile float dephasage=0;
  if (etatInterrupt==false){
    etatInterrupt=true;

    digitalWrite(Plume,LOW);//Plume levée
    delayMicroseconds(1000000);

    while (digitalRead(Depart)==HIGH){

      dephasage=droite(analogRead(Phi),0,4095,0,M_PI);

    if (digitalRead(VX_f_t)==LOW){
      analogWrite(DAC0,0);//Positionnement a gauche de la feuille
      if (digitalRead(VY_Sinus)==LOW){analogWrite(DAC1,int(cosinus(M_PI/2+dephasage)));}
      if (digitalRead(VY_Triangle)==LOW) {analogWrite(DAC1,int(triangle(0+dephasage)));}
      if (digitalRead(VY_SinAmorti)==LOW) {analogWrite(DAC1,int(cosam(0,0)));}
      if (digitalRead(VY_Lissajous1)==LOW) {analogWrite(DAC1,int(cosinus(2*(M_PI/2+dephasage
      ))));}
      if (digitalRead(VY_Lissajous2)==LOW) {analogWrite(DAC1,int(cosinus(3*(M_PI/2+dephasage
```

```
      ))));}
      }
    if (digitalRead(VX_Sinus)==LOW) {
      //---
      if (digitalRead(VY_Sinus)==LOW){
        analogWrite(DAC1,int(cosinus(0+dephasage)));
        analogWrite(DAC0,int(cosinus(0)));}
      if (digitalRead(VY_Triangle)==LOW) {
        analogWrite(DAC1,int(triangle(0+M_PI/2+dephasage)));
        analogWrite(DAC0,int(cosinus(0)));}
      if (digitalRead(VY_SinAmorti)==LOW) {
         analogWrite(DAC1,int(cosam(0,0)));
         analogWrite(DAC0,int(cosinus(0+dephasage)));}
      if (digitalRead(VY_Lissajous1)==LOW) {
        analogWrite(DAC1,int(cosinus(3*(0+dephasage))));
        analogWrite(DAC0,int(cosinus(2*0)));;}
      if (digitalRead(VY_Lissajous2)==LOW) {
        analogWrite(DAC1,int((cosinus(0+dephasage+M_PI/2)+0.4*cosinus(16*(0+dephasage+M_PI/2
        )))/1.4));
        analogWrite(DAC0,int((cosinus(0)+0.4*cosinus(16*0))/1.4));}
     //---
    }

    if (digitalRead(VX_Triangle)==LOW) {
      //---
      if (digitalRead(VY_Sinus)==LOW){
        analogWrite(DAC1,int(cosinus(0+dephasage)));
        analogWrite(DAC0,int(triangle(0+M_PI/2)));}
      if (digitalRead(VY_Triangle)==LOW) {
       analogWrite(DAC1,int(triangle(0+dephasage)));
       analogWrite(DAC0,int(triangle(0+M_PI/2)));}
      if (digitalRead(VY_SinAmorti)==LOW) {
        analogWrite(DAC1,int(cosam(0,0)));
        analogWrite(DAC0,int(triangle(0+dephasage+M_PI/2)));;}
      if (digitalRead(VY_Lissajous1)==LOW) {
        analogWrite(DAC1,int(cosinus(3*(0+dephasage))));;
        analogWrite(DAC0,int(triangle(2*(0+M_PI/2))));}
      if (digitalRead(VY_Lissajous2)==LOW) {
        analogWrite(DAC1,int((cosinus(0+dephasage+M_PI/2)+0.4*cosinus(16*(0+dephasage+M_PI/2
        )))/1.4));
        analogWrite(DAC0,int((triangle(0)+0.4*triangle(16*0))/1.4));;}
      //---
    }

  }//fin while
      digitalWrite(Plume,HIGH);//Plume baissée

  }//fin if

}//fin interrupt

//Fonctions mathématiques

float triangle(float x){
  return (1/M_PI)*acos(cos(x))*4095;
}
```

```
float cosinus(float x) {
  return ((cos(x)+sin(x+M_PI/2))/2+1)*2047;
}

float droite(float x,float  in_min,float in_max,float out_min, float out_max)
{
  return (x*(out_max-out_min)+out_min*in_max-out_max*in_min)/(in_max-in_min);
}

float cosam(float x,float coeff) {
  return (((cos(x)+sin(x+M_PI/2))*exp(-x*coeff))/2+1)*2047;

  }

  //Fonction permettant de "lire" le coefficient d'amortissement
float getcoeff(){
  int i=0;
  while(digitalRead(Coeff[i])==HIGH && i<=9){i++;}
  return coeffVal[i];
}


void setup() {


 //initialisation des pins et activation de la resistance de pull up interne
  pinMode(VY_Lissajous2,INPUT_PULLUP);
  pinMode(VY_Lissajous1,INPUT_PULLUP);
  pinMode(VY_SinAmorti,INPUT_PULLUP);
  pinMode(VY_Triangle,INPUT_PULLUP);
  pinMode(VY_Sinus,INPUT_PULLUP);
  pinMode(Plume,OUTPUT);
  pinMode(Depart,INPUT_PULLUP);
  pinMode(VX_f_t,INPUT_PULLUP);
  pinMode(VX_Triangle,INPUT_PULLUP);
  pinMode(VX_Sinus,INPUT_PULLUP);

  digitalWrite(VY_Lissajous2,HIGH);
  digitalWrite(VY_Lissajous1,HIGH);
  digitalWrite(VY_SinAmorti,HIGH);
  digitalWrite(VY_Triangle,HIGH);
  digitalWrite(VY_Sinus,HIGH);
  digitalWrite(Depart,HIGH);
  digitalWrite(VX_f_t,HIGH);
  digitalWrite(VX_Triangle,HIGH);
  digitalWrite(VX_Sinus,HIGH);

  //idem pour le selecteur "coeff"
  for(int i=0;i<=9;i++){
  pinMode(Coeff[i],INPUT_PULLUP);
  digitalWrite(Coeff[i],HIGH);
  }

  //declaration des interruption sur chaque changement de position des selecteurs
  etatInterrupt=true;
  attachInterrupt(VY_Lissajous2,interruptCoupParCoup,FALLING);
  attachInterrupt(VY_Lissajous1,interruptCoupParCoup,FALLING);
```

```
  attachInterrupt(VY_SinAmorti,interruptCoupParCoup,FALLING);
  attachInterrupt(VY_Triangle,interruptCoupParCoup,FALLING);
  attachInterrupt(VY_Sinus,interruptCoupParCoup,FALLING);
  attachInterrupt(VX_f_t,interruptCoupParCoup,FALLING);
  attachInterrupt(VX_Triangle,interruptCoupParCoup,FALLING);
  attachInterrupt(VX_Sinus,interruptCoupParCoup,FALLING);
  etatInterrupt=false;
  //resolution lecture/ecriture analogique mise à 12 bits
  analogReadResolution(12);
  analogWriteResolution(12);


  interruptCoupParCoup();

}

void loop() {


// Déclaration des variables
 float angle=0;
 float voieX=0;
 float voieY=0;
 float dephasage;
 float coefficient;
 float maximum;
 int i=0;
  //3 position sur la voie x, 5 sur la voies Y = 15 boucles
  angle=0;
  //------------VY_Sinus------------
  while((digitalRead(VY_Sinus)==LOW)&&(digitalRead(VX_Sinus)==LOW)){
    if(angle>=2*M_PI){
     angle=0;
   }
   dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
   voieX=cosinus(angle);
   voieY=cosinus(angle+dephasage);

   analogWrite(DAC0,int(voieX));
   analogWrite(DAC1,int(voieY));

  angle+=0.001;
  delay(2);
  }
    //reinitialisation
  angle=0;
  etatInterrupt=false;

  while((digitalRead(VY_Sinus)==LOW)&&(digitalRead(VX_Triangle)==LOW)){
   if(angle>=2*M_PI){
     angle=0;
   }
   dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
   voieX=triangle(angle+M_PI/2);
   voieY=cosinus(angle+dephasage);
```

```
 analogWrite(DAC0,int(voieX));
 analogWrite(DAC1,int(voieY));

 angle+=0.001;
 delay(2);
}
//reinitialisation
angle=0;
etatInterrupt=false;

while((digitalRead(VY_Sinus)==LOW)&&(digitalRead(VX_f_t)==LOW)){
  etatInterrupt=true;
  dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
  analogWrite(DAC1,int(cosinus(M_PI/2+dephasage)));
  delay(500);
  digitalWrite(Plume,HIGH);//Plume baissée
  for(angle=0;angle<=2*M_PI;angle+=0.001){
     voieX=triangle(angle/2);
     voieY=cosinus(angle+M_PI/2+dephasage);

     analogWrite(DAC0,int(voieX));
     analogWrite(DAC1,int(voieY));

     delay(1);
    }
  etatInterrupt=false;
  interruptCoupParCoup();
}
  //reinitialisation
angle=0;
etatInterrupt=false;

//-----------VY_Triangle-----------
while((digitalRead(VY_Triangle)==LOW)&&(digitalRead(VX_Sinus)==LOW)){
 if(angle>=2*M_PI){
   angle=0;
 }
 dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
 voieX=cosinus(angle);
 voieY=triangle(angle+M_PI/2+dephasage);

 analogWrite(DAC0,int(voieX));
 analogWrite(DAC1,int(voieY));

 angle+=0.001;
 delay(2);

}
   //reinitialisation
angle=0;
etatInterrupt=false;

while((digitalRead(VY_Triangle)==LOW)&&(digitalRead(VX_Triangle)==LOW)){
 if(angle>=2*M_PI){
   angle=0;
 }
 dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
```

```
  voieX=triangle(angle+M_PI/2);
  voieY=triangle(angle+dephasage);

  analogWrite(DAC0,int(voieX));
  analogWrite(DAC1,int(voieY));

angle+=0.001;
delay(2);
}
  //reinitialisation
angle=0;
etatInterrupt=false;

while((digitalRead(VY_Triangle)==LOW)&&(digitalRead(VX_f_t)==LOW)){
  etatInterrupt=true;
  dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
  analogWrite(DAC1,int(triangle(0+dephasage)));
  delay(500);
  digitalWrite(Plume,HIGH);//Plume baissée
  for(angle=0;angle<=2*M_PI;angle+=0.001){
     voieX=triangle(angle/2);
     voieY=triangle(angle+dephasage);

     analogWrite(DAC0,int(voieX));
     analogWrite(DAC1,int(voieY));

     delay(1);
    }
  etatInterrupt=false;
  interruptCoupParCoup();

  }
  //reinitialisation
angle=0;
etatInterrupt=false;

//-------------VY_SinAmorti-------------

while((digitalRead(VY_SinAmorti)==LOW)&&(digitalRead(VX_Sinus)==LOW)){

 if(angle>=12*M_PI){
   angle=0;
 }
 dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
 coefficient=getcoeff();

 voieX=cosinus(angle+dephasage);
 voieY=cosam(angle,coefficient);

 analogWrite(DAC0,int(voieX));
 analogWrite(DAC1,int(voieY));

 angle+=0.001;
 delay(2);
  }
    //reinitialisation
 angle=0;
```

```
  etatInterrupt=false;

   while((digitalRead(VY_SinAmorti)==LOW)&&(digitalRead(VX_Triangle)==LOW)){
   if(angle>=12*M_PI){
     angle=0;
   }
   dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
   coefficient=getcoeff();

   voieX=triangle(angle+dephasage+M_PI/2);
   voieY=cosam(angle,coefficient);

   analogWrite(DAC0,int(voieX));
   analogWrite(DAC1,int(voieY));

   angle+=0.001;
   delay(2);
    }
    //reinitialisation
  angle=0;
 etatInterrupt=false;

  while((digitalRead(VY_SinAmorti)==LOW)&&(digitalRead(VX_f_t)==LOW)){
    etatInterrupt=true;
    coefficient=getcoeff();
    analogWrite(DAC1,int(cosam(0,coefficient)));
    delay(500);
    digitalWrite(Plume,HIGH);//Plume baissée
    for(angle=0;angle<=12*M_PI;angle+=0.001){
       voieX=triangle(angle/12);
       voieY=cosam(angle,coefficient);

       analogWrite(DAC0,int(voieX));
       analogWrite(DAC1,int(voieY));


      }
    etatInterrupt=false;
    interruptCoupParCoup();
    }
    //reinitialisation
  angle=0;
  etatInterrupt=false;

  //------------Lissajous_1-------------
  while((digitalRead(VY_Lissajous1)==LOW)&&(digitalRead(VX_Sinus)==LOW)){
   if(angle>=2*M_PI){
     angle=0;
   }
   dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
   voieX=cosinus(2*angle);
   voieY=cosinus(3*(angle+dephasage));

   analogWrite(DAC0,int(voieX));
   analogWrite(DAC1,int(voieY));

   delay(2);
```

```
angle+=0.001;
}
  //reinitialisation
angle=0;
etatInterrupt=false;

while((digitalRead(VY_Lissajous1)==LOW)&&(digitalRead(VX_Triangle)==LOW)){
 if(angle>=2*M_PI){
   angle=0;
 }
 dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
 voieX=triangle(2*(angle+M_PI/2));
 voieY=cosinus(3*(angle+dephasage));


 analogWrite(DAC0,int(voieX));
 analogWrite(DAC1,int(voieY));

 angle+=0.001;

 delay(2);
  }
    //reinitialisation
angle=0;
etatInterrupt=false;

while((digitalRead(VY_Lissajous1)==LOW)&&(digitalRead(VX_f_t)==LOW)){
  etatInterrupt=true;
  dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
  analogWrite(DAC1,int(cosinus(2*(M_PI/2+dephasage))));
  delay(500);
  digitalWrite(Plume,HIGH);//Plume baissée
  for(angle=0;angle<=2*M_PI;angle+=0.001){
     voieX=triangle(angle/2);
     voieY=cosinus(2*(angle+M_PI/2+dephasage));

     analogWrite(DAC0,int(voieX));
     analogWrite(DAC1,int(voieY));

     delay(1);
    }
  etatInterrupt=false;
  interruptCoupParCoup();
  }
  //reinitialisation
angle=0;
etatInterrupt=false;

//------------Lissajous_2---------------
while((digitalRead(VY_Lissajous2)==LOW)&&(digitalRead(VX_Sinus)==LOW)){
 if(angle>=2*M_PI){
   angle=0;
 }
 dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
 voieX=(cosinus(angle)+0.4*cosinus(16*angle))/1.4;
 voieY=(cosinus(angle+dephasage+M_PI/2)+0.4*cosinus(16*(angle+dephasage+M_PI/2)))/1.4;
```

```
   analogWrite(DAC0,int(voieX));
   analogWrite(DAC1,int(voieY));

  angle+=0.001;

    delay(3);
    }
    //reinitialisation
  angle=0;
  etatInterrupt=false;

  while((digitalRead(VY_Lissajous2)==LOW)&&(digitalRead(VX_Triangle)==LOW)){
   if(angle>=2*M_PI){
     angle=0;
   }
   dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
   voieX=(triangle(angle)+0.4*triangle(16*angle))/1.4;
   voieY=(cosinus(angle+dephasage+M_PI/2)+0.4*cosinus(16*(angle+dephasage+M_PI/2)))/1.4;


   analogWrite(DAC0,int(voieX));
   analogWrite(DAC1,int(voieY));

   angle+=0.001;
   delay(3);
    }
    //reinitialisation
  angle=0;
  etatInterrupt=false;

  while((digitalRead(VY_Lissajous2)==LOW)&&(digitalRead(VX_f_t)==LOW)){
    etatInterrupt=true;
    dephasage=droite(analogRead(Phi),0,4095,0,M_PI);
    analogWrite(DAC1,int(cosinus(3*(angle+M_PI/2+dephasage))));
    delay(500);
    digitalWrite(Plume,HIGH);//Plume baissée
    for(angle=0;angle<=2*M_PI;angle+=0.001){
       voieX=triangle(angle/2);
       voieY=cosinus(3*(angle+M_PI/2+dephasage));

       analogWrite(DAC0,int(voieX));
       analogWrite(DAC1,int(voieY));

       delay(1);
      }
    etatInterrupt=false;
    interruptCoupParCoup();
  }

    //reinitialisation
  angle=0;
  etatInterrupt=false;

}//fin loop()
```