/\* Déclaration des ressources externes \*/

#include "mbed.h"

/\* Déclaration des variables globales \*/

#define T 0.001

// Déclaration des sorties

DigitalOut horloge(D3);

DigitalOut enable1(D8);

DigitalOut CW1(D7);

DigitalOut enable2(D2);

DigitalOut CW2(D4);

DigitalIn interrupteur1(D9);

DigitalIn interrupteur2(D10);

PwmOut Servo(D11);

Serial pc(USBTX, USBRX);

void initialisation(void);

void position(int x1, int x2);

void move(int moteur, int sens);

void draw(void);

/\* Fonction principale \*/

int main(){

 pc.baud(9600);

 int x,y;

 int\* px=&x;

 int\* py=&y;

 horloge=0;

 enable1=0;

 enable2=0;

 Servo.period\_ms(20);

 Servo.pulsewidth\_us(1500);

 initialisation();

 while(1){

 if(pc.readable()){

 pc.printf("entrez x\n");

 pc.scanf("%d",&x);

 pc.printf("entrez y\n");

 pc.scanf("%d",&y);

 pc.printf("x=%d et y=%d\n",x,y);

 position(x,y);

 draw();

 initialisation();

 }

}

}

void initialisation(){

 enable1=1;

 enable2=1;

 while(interrupteur2!=0 or interrupteur1!=0){

 if(interrupteur2!=0){

 move(2,0);}

 if(interrupteur2==0){

 enable2=0;}

 if(interrupteur1!=0){

 move(1,0);}

 if(interrupteur1==0){

 enable1=0;}

}

}

void move(int moteur, int sens){

 if(moteur==1){

 if(sens==1){

 CW1=1;

 }

 else{

 CW1=0;

 }

 horloge=1;

 wait(T);

 horloge=0;

 wait(T);

 }

 if(moteur==2){

 if(sens==1){

 CW2=1;

 }

 else{

 CW2=0;

 }

 horloge=1;

 wait(T);

 horloge=0;

 wait(T);}

 }

void position(int x1,int x2){

 /\*x1 et x2 sont des coordonnées en mm donc on transforme l'information pour avoir une commande en nombre de pas\*/

 enable1=1;

 enable2=1;

 int i;

 float pas=0.21;

 float pas\_x1=x1/pas;

 float pas\_x2=x2/pas;

 if(pas\_x1<pas\_x2){

 for(i=pas\_x1;i>0;i--){

 move(2,1);

 move(1,1);

 }

 for(i=pas\_x2-pas\_x1;i>0;i--){

 move(2,1);

 }

 }

 else{

 for(i=pas\_x2;i>0;i--){

 move(2,1);

 move(1,1);

 }

 for(i=pas\_x1-pas\_x2;i>0;i--){

 move(1,1);

 }

 }

 enable1=0;

 enable2=0;

}

void draw(void){

 Servo.pulsewidth\_us(2000);

 position(30,30);

 Servo.pulsewidth\_us(1500);

 }