```
#include "mbed.h"
#define MEM_SIZE 5000
#define MEM_TYPE uint32_t
#define MAXPOS 50        // en milimetros
#define SS_TIME 100      // en microsegundos
#define POSDRAW 10
#define periodo 20
#define CM_EJECUTAR 0xff
#define CM_GUARDAR 0xfe
#define CM_VERTEX2D 0xfd
#define CM_DRAW 0xfc
#define CM_NODRAW 0xfb
#define CM_STOP 0xfa
#define CM_END 0xf0
```

Definimos variables globales

```
int coord2us(float coord)
{
    if(0 <= coord <= MAXPOS)
        return int(750+coord*1900/50);// u6
    return 750;

}

void draw()
{
    z=coord2us(POSDRAW);
    motor_z.pulsewidth_us(z);
    wait(1);
}
void nodraw()
{

    z=coord2us(MAXPOS);
    motor_z.pulsewidth_us(z);
    wait(1);
}
```

Subrutinas para conversión y subir, bajar eje z

```
void vertex2d(float x, float y){

    int pulseX = coord2us(x);
    int pulseY = coord2us(y);

    motor_x.pulsewidth_us(pulseX);
    motor_y.pulsewidth_us(pulseY);
    wait_ms(SS_TIME);

}
////////////////////////////////////////////////////
void ejecutar(){
    MEM_TYPE bandera=0;
    pc.printf("se esta ejecutando el dibujo...\r\n");
    for(i=0;i<mem_head-1;i++)
    {
        bandera=arreglo2[i];
        while (bandera==CM_VERTEX2D)//FD F0
        {

            pc.printf("entro if porque arreglo2[%i] =%x\r\n",i,arreglo2[i]);
            i++;
```

```
76              i++;
77              pc.printf("[%x]",arreglo2[i]);
78
79              if(arreglo2[i]==CM_END)
80              {
81                  vertex2d(px,py);
82                  wait(1);
83              }
84              i++;
85              bandera=arreglo2[i];
86          }
87          if(arreglo2[i]==CM_DRAW)
88          {
89              i++;
90              if(arreglo2[i]==CM_END)
91              {
92                  draw();
93              }
94          }
95          if(arreglo2[i]==CM_NODRAW)
96          {
97              i++;
98              if(arreglo2[i]==CM_END)
```

Subrutina para dibujar en los ejes x, y

Subrutina ejecutar datos guardados

```
void guardar(){
    pc.printf("se inicia el comado de guardar..\r\n");
    mem_head=0;
            do
            {
               // pc.printf("entro while llenar\n\r");
                arreglo2[mem_head]=pc.getc();
                b=arreglo2[mem_head];
               // pc.printf("posicion %i se lleno con %x\r\n",mem_head,arreglo
                mem_head++;
            }
            while(b!=CM_STOP);

 }
```

Subrutina guardar datos, hasta comando FA

```
int main() {
    // configuracion de  periodo
    motor_x.period_ms(periodo);
    motor_y.period_ms(periodo);
    motor_z.period_ms(periodo);
    motor_x.pulsewidth_us(725);
    motor_y.pulsewidth_us(725);
    motor_z.pulsewidth_us(2650);
    int posx=0;
    int posy=0;
    char caracter;
  while(1)
    {
        caracter=pc.getc();
        switch (caracter) {
            case  CM_EJECUTAR: ejecutar(); break;
            case  CM_GUARDAR: guardar(); break;
            default: pc.printf("error de comando\r\n");break ;
        }
    }
}
```

Programa principal: inicializar periodo de los servo motor y la posición inicial de estos

Recibir carácter desde el computador para iniciar guardar datos, ejecución de datos