



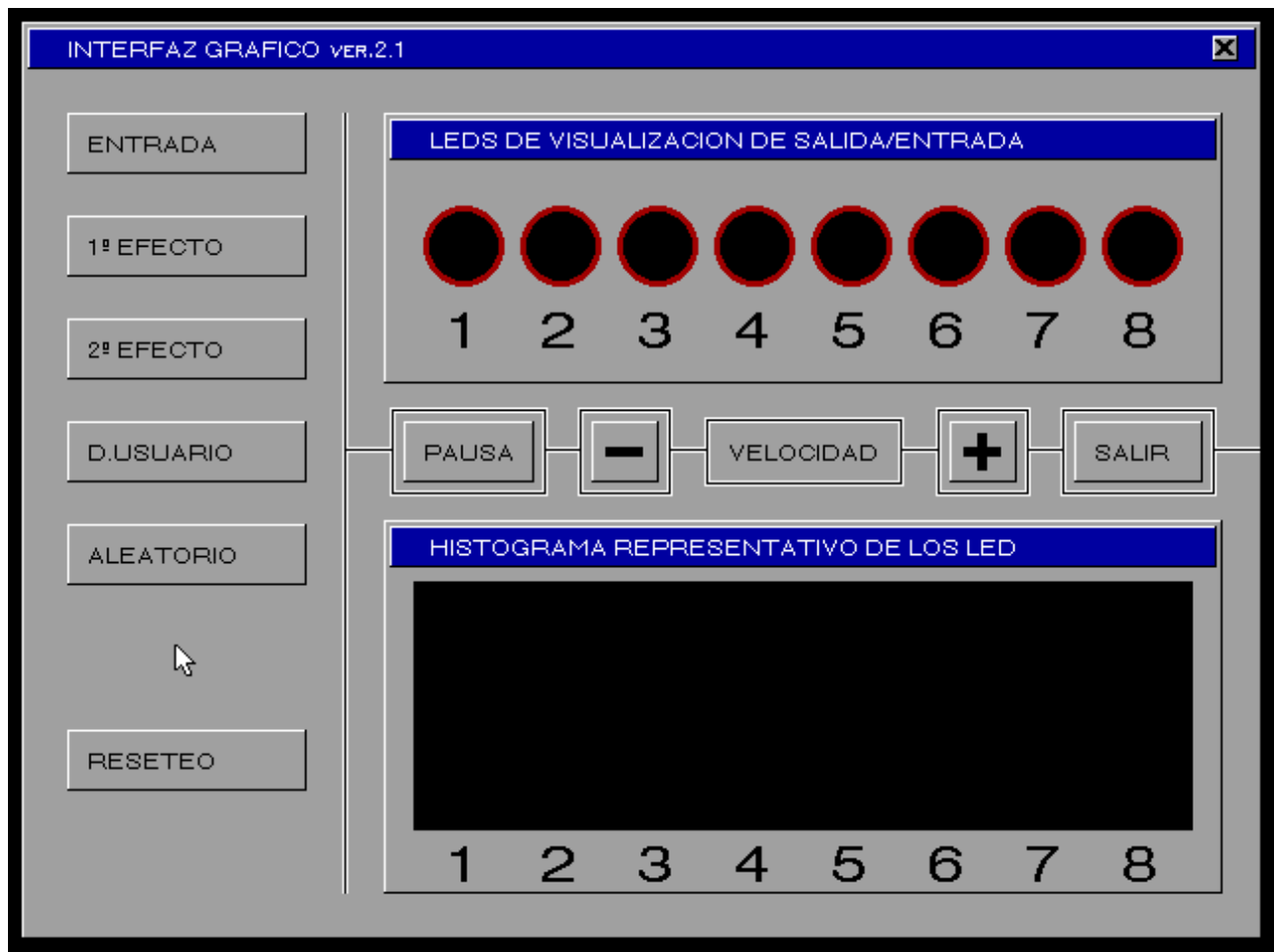
INTRODUCCIÓN

Este programa ha sido creado en lenguaje de programación QUICK C, y su finalidad es la siguiente, con el se podrá controlar un prototipo que este conectado al puerto paralelo.

Introduciendo una serie de combinaciones de leds o seleccionando un efecto ya establecido, serán representado por pantalla y enviado en forma de información hexadecimal al puerto paralelo donde el prototipo lo decodificara y iluminara los leds correspondientes. Además este programa tiene otra función que es la de recoger datos, en este caso es el prototipo el que envía en código hexadecimal para ser recogido desde el puerto paralelo, seguidamente se decodifica y el programa nos mostrara el resultado por pantalla.

CONOCIENDO EL ENTORNO GRAFICO

El entorno grafico de la aplicación es bastante sencillo y de fácil comprensión. En la figura podemos observar la pantalla principal del entorno.



Como podemos observar a la izquierda de la pantalla encontramos los botones de selección de efecto, seleccionando cada uno de estos botones podremos visualizar un efecto diferente.

En la parte derecha superior encontramos los led representativos de cada efecto, cada uno con un numero debajo que representa su peso, es decir, 8 el de mayor peso y 1 el de menor.



Debajo de los leds encontramos unos botones de control, estos botones solo están disponibles durante el efecto, el único que fuera del efecto tiene alguna utilidad será el de salir que junto con la X de la ventana nos sacara del programa.



Estos botones durante el efecto nos permitirán pararlo, aumentarle la velocidad, disminuirla y salir de el.

Debajo de los botones nos encontramos con el histograma grafico, en el se representa gráficamente la cantidad de veces que ha lucido un led, y con el podemos realizar sumas de efectos ya que mientras no pulsemos el botón de reseteo que se encuentra debajo de los botones de selección no se pondrá a cero.



BOTONES DE ACCESO RAPIDO

FUERA DEL EFECTO:

`E'->Entrada, Ejecuta la entrada.

`1'->Efecto primero, ejecuta el primer efecto.

`2'->Efecto segundo, ejecuta el segundo efecto.

`D'->Efecto usuario, abre la ventana de creación de efecto usuario.

`A'->Efecto aleatorio, abre la ventana de creación de efecto aleatorio.

`R'->Reseteo, resetea el histograma grafico, lo pone a cero.

`S'->Salir, sales del programa.

DENTRO DE UN EFECTO:

`P'->Pausa, realiza una pausa hasta que se vuelva a pulsar el botón.

`+'->Aumenta la velocidad del efecto.

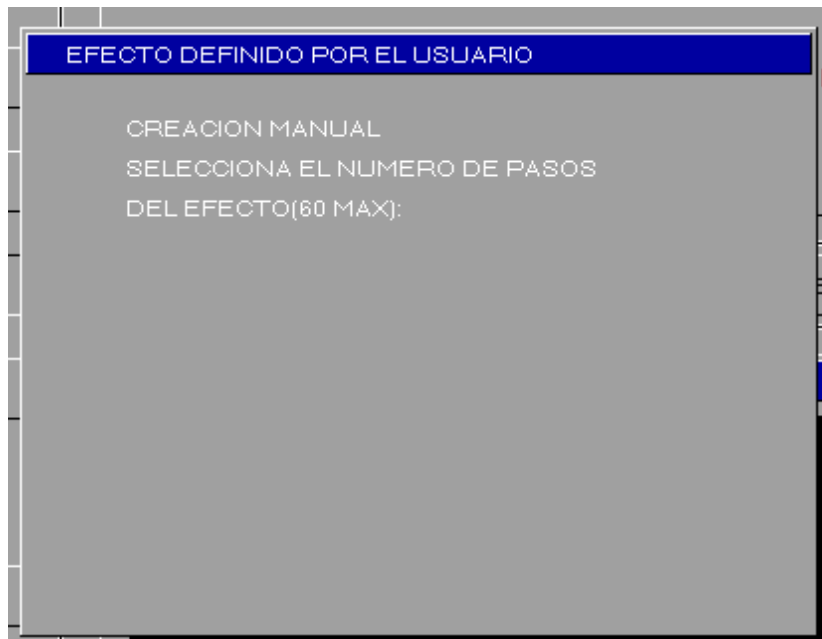
`-`->Disminuye la velocidad del efecto.

`S'->Salir, sales del efecto al menú principal.

COMO CONFIGURAR UN EFECTO DEFINIDO Y ALEATORIO

EFECTO DEFINIDO:

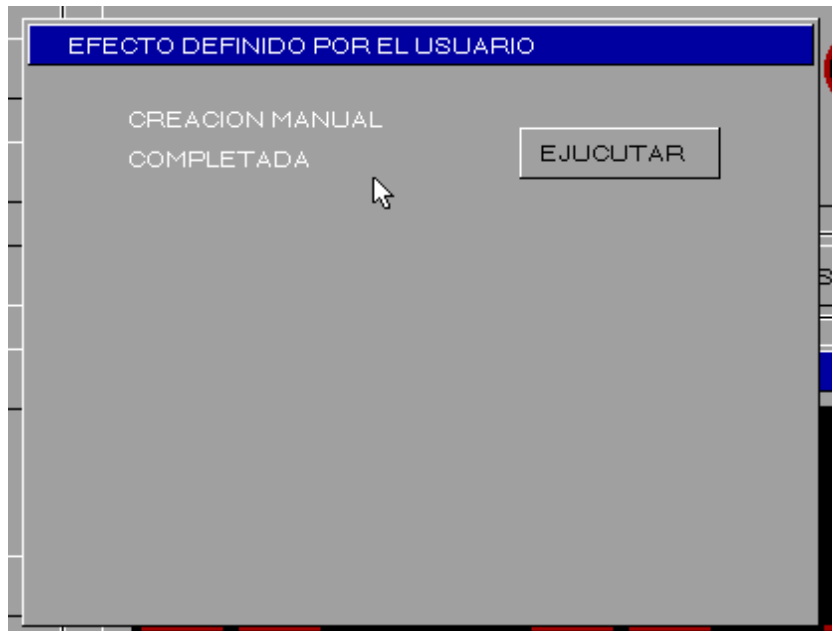
Lo primero que debemos hacer es seleccionar en los botones de selección (los de la izquierda) el botón de `D.USUARIO', esto nos llevara a una subventana donde nos indicara el numero de pasos que vamos a realizar, con un máximo de 60.



Una vez seleccionado el numero de pasos o cambios, nos dispondremos a llenar de información el efecto para cada uno de los led. Esta acción se realizara pulsando el `1' si queremos que se encienda el `0' si queremos que no se ilumine y el `2' si queremos volver a la pantalla anterior.



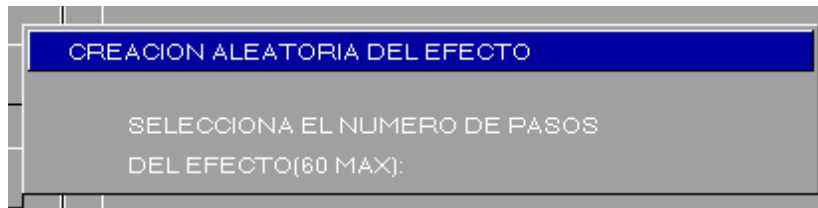
Una vez introducidos correctamente todos los datos en el efecto nos indicara que ejecutemos el efecto.



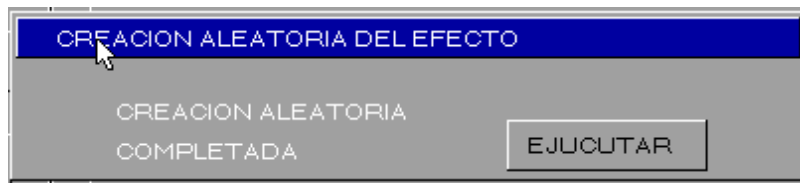
Una vez que pulsemos el botón de ejecutar ejecutaremos el efecto.

EFECTO ALEATORIO:

Lo primero es seleccionar 'ALEATORIO' en los botones de selección con lo que se nos abrirá una subventana. Aquí seleccionaremos el numero de pasos a realizar.



Seguidamente el programa rellanara e efecto automáticamente y cuando termine nos indicara que lo ejecutemos.



Pulsando el botón de ejecutar ejecutaremos el efecto.

CONOCIENDO EL PROGRAMA POR DENTRO

FUNCIÓN EJECUCIÓN

Esta es la base de todo el programa aquí son enviados todos los datos de cada uno de los efectos mediante un array bidimensional llamado 'usuario'. Aquí es seleccionado el led que se va a iluminar en cada momento. Su estructura es bastante simple, mediante IF se van seleccionando los led que están iluminados o a '1' o los que están apagados o a '0' este dato lo proporciona el array 'usuario', y una vez que se conozca los led que están encendidos se realiza una llamada a la función codigout, salida de código, donde mediante una suma lógica

conseguimos el valor hexadecimal que mandamos al puerto paralelo para que lo reciba el prototipo.

Luego una vez acabado el efecto realiza una función de apagado de leds para poner todos los led a cero.

Cada uno de los led tiene una función de encendido, apagado, y de display que significa si ha de enviar un cuadro al histograma.

LAS FUNCIONES DE LOS LED

Como he comentado anteriormente el programa realiza una serie de llamadas a las funciones de cada led para realizar su encendido en el interfaz y en el prototipo.

Cada led posee tres funciones:

–Led encendido: Aquí se dibuja el led encendido y se pone a 1 una variable llamada ``leda'`, ejemplo del primero, que comunica que ese led ya ha sido encendido y que esta esperando que lo apaguen, en esta misma función se hace la llamada al encendido del display.

–Led apagado: Aquí se dibuja el led en estado de apagado, la variable ``leda'` pasa a tener valor ``0'` y así ya se sabe que puede volver a ser encendido.

–Led display: Aquí se envía la información al histograma para que dibuje el cuadro en el contador. Esto vendrá sujeto al valor de la variable ``leda'` ya que si esta a uno no lo va a dibujar para así evitar que cuente varias veces un mismo led que no se apaga.

EL HISTOGRAMA GRAFICO

El histograma se basa en representar cuantas veces se ha iluminado un led por lo que viene relacionado con la función de los led, concretamente con la display. Cada led en el display tiene establecidas las coordenadas del eje X para dibujar el cuadrado en el histograma y las del eje Y están definidas en otra función llamada ``tabla'` ya que estos valores van a ser los mismos para todos. El valor que corresponde en la tabla para cada caso viene dado por la variable ``posi[0]++'` para el caso primero, que va incrementándose cada vez que un led se enciende y se apaga.

ENTRADA

La entrada es una función de recogida de datos desde el prototipo hasta el ordenador. El prototipo posee unos switch que selecciona el dato que va a mandar al puerto paralelo donde el programa los recibe y los decodifica para luego representarlos por pantalla.

Esta operación de decodificación se realiza mediante una mascara que trata los datos enviados en dos bloques y los une para obtener el resultado final pasándoles por unos IF que irán restringiendo las opciones hasta dar con el código correcto.

EFFECTO PRIMERO Y SEGUNDO

Esta función del programa se centra en la representación del efecto por pantalla y enviarlo al prototipo para que lo represente. La realización de esta función se basa en el array ``usuario'` donde se guardara toda la información del efecto y se enviara a la función de ejecución para ser ejecutado.

EFFECTO DEFINIDO POR EL USUARIO

Esta función es parecida a las anteriores con la diferencia que el array `usuario` en vez de tener unos valores ya establecidos nosotros le introducimos los que queremos pudiendo realizar así el efecto que queramos. El número máximo de cambios que puede tener nuestro efecto será de 60, entendemos por cambios a cada una de las secuencias.

EFEECTO ALEATORIO

En esta función la base es la misma que en el anterior caso con la diferencia que con solo introducir el número de cambios que queremos el ordenador con la función `rand` nos introducirá el código en el array `usuario` para su próxima ejecución.

LA PAUSA, SALIDA, VELOCIDAD DEL EFECTO

La pausa, salida y velocidad del efecto se basa en el retardo, que es el tiempo que debemos dejar entre cada cambio porque sino el efecto sería casi inmediato. El fundamento es un `switch` que hay en el retardo que cuando se pulsa una tecla con la función `bkhit()` actúa.

Si pulsamos la pausa entrará en un `do while` donde lo mantendrá hasta que volvamos a pulsar la pausa.

Si pulsamos los botones de velocidad multiplicaremos el retardo por un número mayor o menor según queramos menos o más velocidad respectivamente.

Si pulsamos el botón de salir saldremos del efecto finalizando la función de ejecución.

LAS LIBRERIAS ADICIONALES

`GRAFICOS.H':

Esta librería tiene como función ejecutar el resto de librerías adicionales, como son `boton.h`, `ventana.h`, `lineas.h`, `leds.h`.

`VENTANA.H':

Esta librería es la principal para casi todo el entorno gráfico, en ella se crea la función `crear\_window` que contiene la siguiente sintaxis:

```
Crear_window(coordenadas del rectángulo,'características');
```

Características:

`c' = Esto hará que aparezca el botón de cerrar.

`d' = Esto hará que aparezca sin el botón de cerrar

Seguidamente podremos escribir el texto que queramos como nombre de la ventana.

Ejemplo:

```
crear_window(100,100,400,400,'c');
```

```
outgtext(INTERFAZ GRAFICO);
```

Esto nos creara una ventana con el botón de cerrar y como titulo `INTERFAZ GRAFICO`

`BOTON.H`

Esta es una librería que nos va a crear todo tipo de botones, contiene la función `crear\_boton`. Su sintaxis es la siguiente.

Crear_boton(coordenadas del botón,'características');

Características:

`d`:Si queremos que la letra sea la de por defecto.

Ejemplo:

Crear_boton(100,200,200,280,'d');

Outgtext(BOTON A);

Esto nos creara un botón que tendrá de texto dentro de el `BOTON A`

`LEDS.H`

Esta librería contiene una función que nos crea los led con sus números, los números únicamente o solo los led, dependiendo del tipo de característica seleccionado. Su sintaxis es algo complicada:

Crea_leds(coordenadas del primer led, numero de leds,espacio disponible para los led,'con numero','sin leds');

Numero de leds= Es el numero de leds que queremos

Espacio disponible: Es el espacio de que disponemos desde el primer led hasta el ultimo.

Con numero:'s' ->si queremos que aparezca el numero debajo de cada led.'n'->si no queremos que aparezca.

Sin led:'s'->si no queremos que aparezcan los led y `n'->si queremos que aparezcan.

Ejemplo:

Crear_leds(100,150.150,200,8,300,'s','n');

Esto nos dibujara 8 leds con su respectivo numero.

`LINEAS.H`

Esta librería tiene una función que crea líneas de tres franjas una blanca otra negra y luego otra blanca. Su sintaxis es asi:

Linea(coordenadas inicio, coordenadas finales);

Un ejemplo:

Linea(100,150,100,300); ->Esto nos crea una linea.

`RATON.H'

En esta librería encontramos las funciones del ratón, con sus registros. Con ella podemos mostrar el ratón, quitarlo, ver su posición, y comprobar si se ha pulsado algún botón.

Yo he modificado dos funciones:

Posición del ratón que he suprimido los punteros y los he cambiado por dos variables.

Ver si se ha pulsado el botón izquierdo que recojo en una variable los datos en vez de usar `RETURN'.

BOTON.H

```
#include"stdio.h"
```

```
#include"graph.h"
```

```
void crear_boton(int botxa,int botya,int botxb,int botyb,char eleccion_por_defecto)
```

```
{
```

```
  _setcolor(7);
```

```
  _rectangle(_GFillINTERIOR,botxa,botya,botxb,botyb);
```

```
  _setcolor(15);
```

```
  _rectangle(_GBORDER,botxa,botya,botxb,botyb);
```

```
  _setcolor(0);
```

```
  _moveto(botxa,botyb);
```

```
  _lineto(botxb,botyb);
```

```
  _lineto(botxb,botya);
```

```
  _moveto(botxa+10,((((botyb-botya)/2)+botya)-5));
```

```
  if(eleccion_por_defecto=='d')
```

```
  {_setfont("t'helv'h8w8b");}
```

```
}
```

LINEAS.H

```
#include<stdio.h>
```

```
#include<graph.h>
```

```
void linea(int lineainix,int lineainiy,int lineafinx,int lineafiny)
```

```

{
    _setcolor(0);

    _moveto(lineainix,lineainiy);

    _lineto(lineafinx,lineafiny);

    if(lineainix==lineafinx)
    {
        _setcolor(15);

        _moveto(lineainix-1,lineafiny);

        _lineto(lineainix-1,lineainiy);

        _moveto(lineainix+1,lineainiy);

        _lineto(lineainix+1,lineafiny);

    }

    if(lineainiy==lineafiny)
    {
        _setcolor(15);

        _moveto(lineafinx,lineainiy-1);

        _lineto(lineainix,lineainiy-1);

        _moveto(lineainix,lineainiy+1);

        _lineto(lineafinx,lineainiy+1);

    }

}

LEDS.H

#include<stdio.h>

#include<graph.h>

void crear_leds(int ledxa,int ledya,int ledxb,int ledyb,int numero_leds,int espacio,char con_numero,char
sin_leds)

{

```

```

int contador_for,resultado_separador,ledxacopy;

resultado_separador=((espacio-((ledxb-ledxa)*numero_leds))/(numero_leds-1));

for(contador_for=1;contador_for<=numero_leds;contador_for++)

{

if(sin_leds!='s')

{

_setcolor(4);

_ellipse(_GFillInterior,ledxa,ledya,ledxb,ledyb);

_setcolor(0);

_ellipse(_GFillInterior,ledxa+3,ledya+3,ledxb-3,ledyb-3);

}

if(con_numero=='n')

{

_setfont("t'helv'h25w20b");

_moveto((((ledxb-ledxa)/2)+ledxa)-10,ledyb+10);

if(contador_for==1)_outgtext("1");

if(contador_for==2)_outgtext("2");

if(contador_for==3)_outgtext("3");

if(contador_for==4)_outgtext("4");

if(contador_for==5)_outgtext("5");

if(contador_for==6)_outgtext("6");

if(contador_for==7)_outgtext("7");

if(contador_for==8)_outgtext("8");

}

ledxacopy=ledxa;

ledxa=ledxb+resultado_separador;

```

```
ledxb=ledxb-ledxacopy+resultado_separador+ledxb;
```

```
}
```

```
}
```

OTRO2.C

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<graph.h>
```

```
#include<stdlib.h>
```

```
#include"raton.h"
```

```
int config[30];
```

```
#include"graficos.h"
```

```
void inicio1(void);
```

```
void inicio2(void);
```

```
void inicio3(void);
```

```
void inicio4(void);
```

```
void led1on(void);
```

```
void led2on(void);
```

```
void led3on(void);
```

```
void led4on(void);
```

```
void led5on(void);
```

```
void led6on(void);
```

```
void led7on(void);
```

```
void led8on(void);
```

```
void led1off(void);
```

```
void led2off(void);
```

```
void led3off(void);
```

```
void led4off(void);  
void led5off(void);  
void led6off(void);  
void led7off(void);  
void led8off(void);  
void led1display1(void);  
void led2display1(void);  
void led3display1(void);  
void led4display1(void);  
void led5display1(void);  
void led6display1(void);  
void led7display1(void);  
void led8display1(void);  
void led1display2(void);  
void led2display2(void);  
void led3display2(void);  
void led4display2(void);  
void led5display2(void);  
void led6display2(void);  
void led7display2(void);  
void led8display2(void);  
void ledoffdisplay(void);  
void codigoout(void);  
void ejecucion(void);  
void entrada(void);  
void pulsarboton(void);
```

```
void efecto1(void);  
void efecto2(void);  
void creacionaleatoria(void);  
void definido_usuario(void);  
void creamanual2(void);  
void aleatorio(void);  
void aleatorio2(void);  
void pulsos(void);  
void apagadoleds(void);  
void error_raton(void);  
void combinatorio(void);  
void reseteo(void);  
void organi(void);  
void tabla(void);  
void cerrar(void);  
void cerrarmenu(void);  
void histograma_graf(void);  
void grafica_graf(void);  
void grafica_dia(void);  
void histograma_dia(void);  
void menu_principal(void);  
void comprobacion_bizquierdo(void);  
void principal(void);  
void retardo(float vel);  
void menu_retardo(void);  
void valores_por_defecto(void);
```

```

int
estoyen,ponerdia,ejecutar,levanta,posi[8],usuario[75][8],posicion,reset,leda,ledb,ledc,ledd,lede,ledf,ledg,ledh,a,b,tipop;

char menu,colornúmero,lolo,ejecuta;

unsigned char dataa,datab,datoc,datot,datoe,datof,datog,datoh,datoresult;

float vel,m;

void main()

{

struct videoconfig confi;

_setvideomode(_VRES16COLOR);

_getvideoconfig(&confi);

_registerfonts("c:\\qc25\\samples\\*.FON");

_setgtextvector(1,0);

_setbkcolor(_BLACK);

tipo=raton_instalado();

if (tipo==0)inicializa_raton();

poner_raton_en(320,240);

valores_por_defecto();

principal();

}

void retardo(float vel)

{

float retar;

lolo='A';

m=0;

for(retar=0;retar<5000*vel;retar++)

{

if(kbhit()){lolo=getch();}

```

```

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

}

menu_retardo();

}

void menu_retardo(void)

{

estoyen=2;

switch(lolo)

{

case '+':pulsarboton();m=(-0.2);break;

case '-':pulsarboton();m=0.2;break;

case 'p':

case 'P':

{

pulsarboton();

lolo='A';

for(;;)

{

if(kbhit()){lolo=getch();}

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

if(lolo=='p')break;

}

_setcolor(15);

_moveto(265,209);

```

```

_lineto(200,209);

_lineto(200,239);

_setcolor(0);

_lineto(265,239);

_lineto(265,209);

for(estoyen=0;estoyen<1000;estoyen++);

break;

}

case 's':

case 'S':reset=1;pulsarboton();quitar_raton();principal();break;

case 'g':

case 'G':quitar_raton();grafica_graf();mostrar_raton();break;

case 'h':

case 'H':quitar_raton();histograma_graf();mostrar_raton();break;

}

}

void valores_por_defecto(void)

{

vel=1;

reset=1;

config[0]=0;

config[2]=1;

config[3]=7;

config[4]=15;

config[7]=4;

config[9]=4;

```

```

config[10]=15;

config[23]=1;

}

void principal(void)

{

if(reset==1)inicio1();

inicio2();

inicio3();

if(reset==1){

if(config[0]==0)histograma_graf();

if(config[0]==1)grafica_graf();}

menu_principal();

}

void inicio1(void)

{

crear_window(7,7,633,470,'c');

_outgtext("INTERFAZ GRAFICO");

_setfont("t'modern'h8w6b");

_moveto(163,18);

_outgtext("VER.");

_setfont("t'helv'h8w8b");

_moveto(185,15);

_outgtext("2.1");

}

void inicio2(void)

{

```

```

crear_boton(30,53,150,83,'d');

_outgtext("ENTRADA");

crear_boton(30,105,150,135,'d');

_outgtext("1§ EFECTO");

crear_boton(30,157,150,187,'d');

_outgtext("2§ EFECTO");

crear_boton(30,209,150,239,'d');

_outgtext("D.USUARIO");

crear_boton(30,261,150,291,'d');

_outgtext("ALEATORIO");

/*

crear_boton(30,313,150,343,'d');

_outgtext("POR PULSOS");

crear_boton(30,365,150,395,'d');

_outgtext("COMBINATORIO");

*/

/* coordenadas del ultimo boton=30,417,150,447,'d' */

crear_boton(30,365,150,395,'d');

_outgtext("RESETEO");

linea(170,53,170,447);

}

void pulsarboton(void)

{

int lpuno,lpdos,lptres,lpcuatro,qwert,qwerty;

if(estoyen==0)

{

```

```

qwert=0;qwerty=15;

if(menu=='e'||menu=='E'){lpuno=30;lpdos=53;lptres=150;lpcuatro=83;}

if(menu=='1'){lpuno=30;lpdos=105;lptres=150;lpcuatro=135;}

if(menu=='2'){lpuno=30;lpdos=157;lptres=150;lpcuatro=187;}

if(menu=='d'||menu=='D'){lpuno=30;lpdos=209;lptres=150;lpcuatro=239;}

if(menu=='a'||menu=='A'){lpuno=30;lpdos=261;lptres=150;lpcuatro=291;}

/*

if(menu=='p'||menu=='P'){lpuno=30;lpdos=313;lptres=150;lpcuatro=343;}

if(menu=='c'||menu=='C'){lpuno=30;lpdos=365;lptres=150;lpcuatro=395;}

*/

if(menu=='s'||menu=='S'){lpuno=539;lpdos=209;lptres=603;lpcuatro=239;}

if(menu=='r'||menu=='R'){lpuno=30;lpdos=365;lptres=150;lpcuatro=395;}

}

if(estoyen==1)

{

if(menu=='n'||menu=='N'){lpuno=217;lpdos=270;lptres=252;lpcuatro=290;levanta=1;}

if(menu=='s'||menu=='S'){lpuno=267;lpdos=270;lptres=302;lpcuatro=290;levanta=1;}

if(menu=='c'||menu=='C'){lpuno=317;lpdos=270;lptres=392;lpcuatro=290;levanta=1;}

}

if(estoyen==2)

{

qwert=0;qwerty=15;

if(lolo=='p'||lolo=='P'){lpuno=200;lpdos=209;lptres=265;lpcuatro=239;}

if(lolo=='+' ){lpuno=476;lpdos=209;lptres=509;lpcuatro=239;estoyen=1;levanta=1;}

if(lolo=='-'){lpuno=295;lpdos=209;lptres=328;lpcuatro=239;estoyen=1;levanta=1;}

if(lolo=='s'||lolo=='S'){lpuno=539;lpdos=209;lptres=603;lpcuatro=239;levanta=1;}

```

```

}

_setcolor(qwert);

_moveto(lptres,lpdos);

_lineto(lpuno,lpdos);

_lineto(lpuno,lpcuatro);

_setcolor(qwerty);

_lineto(lptres,lpcuatro);

_lineto(lptres,lpdos);

if(levanta==1)

{

retardo(0.1);

qwert=15;qwerty=0;

_setcolor(qwert);

_moveto(lptres,lpdos);

_lineto(lpuno,lpdos);

_lineto(lpuno,lpcuatro);

_setcolor(qwerty);

_lineto(lptres,lpcuatro);

_lineto(lptres,lpdos);

levanta=0;

}

}

void inicio3(void)

{

crear_window(190,53,613,189,'a');

_outgtext("LEDS DE VISUALIZACION DE SALIDA/ENTRADA");

```

```

/* LEDS */

crear_leds(210,100,250,140,8,383,'n','0');

/* BOTONES DE FUNCION */

crear_boton(200,209,265,239,'d');

_outgtext("PAUSA");

crear_boton(539,209,603,239,'d');

_outgtext("SALIR");

crear_boton(354,209,450,239,'d');

_outgtext("VELOCIDAD");

crear_boton(295,209,328,239,'a');

_rectangle(_GFillInterior,302,222,320,226);

crear_boton(476,209,509,239,'a');

_rectangle(_GFillInterior,483,222,501,226);

_rectangle(_GFillInterior,490,215,494,233);

_setcolor(15);

_rectangle(_GBorder,354,209,450,239);

_rectangle(_GBorder,352,207,452,241);

_rectangle(_GBorder,290,204,333,244);

_rectangle(_GBorder,288,202,335,246);

_rectangle(_GBorder,471,204,514,244);

_rectangle(_GBorder,469,202,516,246);

linea(335,223,352,223);

linea(452,223,469,223);

_rectangle(_GBorder,195,204,270,244);

_rectangle(_GBorder,193,202,272,246);

_rectangle(_GBorder,534,204,608,244);

```

```

_rectangle(_GBORDER,532,202,610,246);

linea(171,223,193,223);

linea(272,223,288,223);

linea(516,223,532,223);

linea(610,223,632,223);

_setcolor(0);

_rectangle(_GBORDER,353,208,451,240);

_rectangle(_GBORDER,289,203,334,245);

_rectangle(_GBORDER,470,203,515,245);

_rectangle(_GBORDER,194,203,271,245);

_rectangle(_GBORDER,533,203,609,245);

}

void histograma_graf(void)

{

crear_window(190,259,613,447,'a');

_outgtext("HISTOGRAMA REPRESENTATIVO DE LOS LED");

/*

crear_boton(525,266,601,278,'d');

_outgtext("GRAFICA");

*/

_setcolor(0);

crear_leds(210,360,250,410,8,383,'n','s');

_rectangle(_GFILLINTERIOR,205,290,598,415);

config[0]=0;

config[23]=1;

if(ejecutar==1)histograma_dia();

```

```

}

void histograma_dia(void)

{
ponerdia=0;

led1display1();

led2display1();

led3display1();

led4display1();

led5display1();

led6display1();

led7display1();

led8display1();

ponerdia=1;

}

void grafica_graf(void)

{

crear_window(190,259,613,447,'a');

_outgtext("GRAFICA REPRESENTATIVA DE LOS LED");

crear_boton(525,266,601,278,'d');

_outgtext("HISTOGR.");

crear_leds(210,360,250,410,8,383,'n','s');

_rectangle(_GFILLINTERIOR,205,290,598,415);

config[0]=1;

config[23]=2;

if(ejecutar==1)grafica_dia();

}

```

```

void grafica_dia(void)

{

ponerdia=0;

led1display2();

led2display2();

led3display2();

led4display2();

led5display2();

led6display2();

led7display2();

led8display2();

ponerdia=1;

}

void comprobacion_bizquierdo(void)

{

if(estoyen==0)

{

if(dx>=0&dx<=640&dy>=0&dy<=480)menu='x';

if(dx>=30&dx<=150&dy>=53&dy<=83)menu='e';

if(dx>=30&dx<=150&dy>=105&dy<=135)menu='1';

if(dx>=30&dx<=150&dy>=157&dy<=187)menu='2';

if(dx>=30&dx<=150&dy>=209&dy<=239)menu='d';

if(dx>=30&dx<=150&dy>=261&dy<=291)menu='a';

/*

if(dx>=30&dx<=150&dy>=313&dy<=343)menu='c';

if(dx>=30&dx<=150&dy>=365&dy<=395)menu='p';

```

```

*/

if(dx>=30&dx<=150&dy>=365&dy<=395)menu='r';

/*

if(config[0]==0){if(dx>=525&dx<=601&dy>=266&dy<=278)menu='g';}

if(config[0]==1){if(dx>=525&dx<=601&dy>=266&dy<=278)menu='h';}

*/

if(dx>=539&dx<=603&dy>=209&dy<=239)menu='s';

if(dx>=609&dx<=621&dy>=14&dy<=26)menu='w';

}

if(estoyen==1)

{

if(dx>=217&dx<=252&dy>=270&dy<=290)menu='s';

if(dx>=267&dx<=302&dy>=270&dy<=290)menu='n';

if(dx>=317&dx<=392&dy>=270&dy<=290)menu='c';

if(dx>=155&dx<=210&dy>=455&dy<=300)menu='p';

quitar_raton();

}

if(estoyen==2)

{

if(dx>=200&dx<=600&dy>=160&dy<=280)error_raton();lolo='B';

if(dx>=200&dx<=265&dy>=209&dy<=239)lolo='p';

if(dx>=295&dx<=328&dy>=209&dy<=239)lolo='-';

if(dx>=476&dx<=509&dy>=209&dy<=239)lolo='+';

if(dx>=539&dx<=603&dy>=209&dy<=239)lolo='s';

/*

if(config[0]==0){if(dx>=525&dx<=601&dy>=266&dy<=278)lolo='g';}

```

```

if(config[0]==1){if(dx>=525&dx<=601&dy>=266&dy<=278)lolo='h';}

*/

}

if(estoyen==3)

{

if(dx>=150&dx<=550&dy>=95&dy<=180)error_raton();ejecuta='0';

if(dx>=400&dx<=500&dy>=150&dy<=175)ejecuta='e';

}

t_izq=0;

}

void error_raton(void)

{

_setcolor(0);

_rectangle(_G_FILLINTERIOR,0,0,5,480);

_rectangle(_G_FILLINTERIOR,633,0,640,480);

_rectangle(_G_FILLINTERIOR,0,471,640,480);

_rectangle(_G_FILLINTERIOR,0,0,640,6);

}

void menu_principal(void)

{

ventana_raton(0,0,630,465);

menu='0';

estoyen=0;

mostrar_raton();

do{

if(kbhit()){menu=getch();}

```

```

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

}while(menu=='0');

quitar_raton();

switch(menu)

{

case 'e':

case 'E':pulsarboton();mostrar_raton();entrada();break;

case '1':pulsarboton();mostrar_raton();efecto1();break;

case '2':pulsarboton();mostrar_raton();efecto2();break;

case 'd':

case 'D':pulsarboton();mostrar_raton();definido_usuario();break;

case 'a':

case 'A':pulsarboton();mostrar_raton();aleatorio();break;

/*

case 'c':

case 'C':combinatorio();break;

case 'p':

case 'P':pulsos();break;

*/

case 'r':

case 'R':pulsarboton();mostrar_raton();reseteo();break;

/*

case 'g':

case 'G':

{

```

```

    retardo(0.2);

    grafica_graf();

    menu_principal();

    break;

}

case 'h':

case 'H':

{

    retardo(0.2);

    histograma_graf();

    menu_principal();

    break;

}

*/

case 'w':retardo(0.2);cerrar();break;

case 'S':

case 's':pulsarboton();retardo(0.2);cerrar();break;

case 'x':

{

    error_raton();

    menu_principal();

    break;

}

default:menu_principal();break;

}

}

```

```
void organi(void)
```

```
{
```

```
vel=1;
```

```
if(reset==1)
```

```
{
```

```
posi[0]=1;
```

```
posi[1]=1;
```

```
posi[2]=1;
```

```
posi[3]=1;
```

```
posi[4]=1;
```

```
posi[5]=1;
```

```
posi[6]=1;
```

```
posi[7]=1;
```

```
}
```

```
leda=0;
```

```
ledb=0;
```

```
ledc=0;
```

```
ledd=0;
```

```
lede=0;
```

```
ledf=0;
```

```
ledg=0;
```

```
ledh=0;
```

```
datoa=0x00;
```

```
datob=0x00;
```

```
datoc=0x00;
```

```
datod=0x00;
```

```

datoe=0x00;

datof=0x00;

datog=0x00;

datoh=0x00;

}

void tabla(void)

{

switch(posicion)

{

case 1:b=410;a=408;break;

case 2:b=406;a=404;break;

case 3:b=402;a=400;break;

case 4:b=398;a=396;break;

case 5:b=394;a=392;break;

case 6:b=390;a=388;break;

case 7:b=386;a=384;break;

case 8:b=382;a=380;break;

case 9:b=378;a=376;break;

case 10:b=374;a=372;break;

case 11:b=370;a=368;break;

case 12:b=366;a=364;break;

case 13:b=362;a=360;break;

case 14:b=358;a=356;break;

case 15:b=354;a=352;break;

case 16:b=350;a=348;break;

case 17:b=346;a=344;break;

```

```

case 18:b=342;a=340;break;

case 19:b=338;a=336;break;

case 20:b=334;a=332;break;

case 21:b=330;a=328;break;

case 22:b=326;a=324;break;

case 23:b=322;a=320;break;

case 24:b=318;a=316;break;

case 25:b=314;a=312;break;

case 26:b=310;a=308;break;

case 27:b=306;a=304;break;

case 28:b=302;a=300;break;

case 29:b=298;a=296;break;

case 30:b=294;a=292;break;

}

}

void led1on(void)

{

  _setcolor(config[7]);

  _ellipse(_GFILLINTERIOR,210,100,250,140);

  datoa=0x80;

  if(leda==0)

  {

    if(config[23]==1)

    {led1display1();}

    if(config[23]==2)

    {led1display2();}

```

```

}

}

void led1off(void)

{

    _setcolor(0);

    _ellipse(_GFILLINTERIOR,213,103,247,137);

    dataa=0x00;

    ledoffdisplay();

    leda=0;

}

void led1display1(void)

{

    for(posicion=1;posicion<=posi[0];posicion++)

    {

        tabla();

        _setcolor(config[9]);

        _rectangle(_GFILLINTERIOR,210,a,250,b);

    }

    /*

    if(!(ponerdia==0))

    {

        */

        posi[0]++;

        leda=1;

        /*

    }

```

```

*/

}

void led1display2(void)

{

if(!(ponerdia==0))

{

posi[0]++;

leda=1;

}

}

void led2on(void)

{

_setcolor(config[7]);

_ellipse(_GFillInterior,259,100,299,140);

datob=0x40;

if(ledb==0)

{

if(config[23]==1)

{led2display1();}

if(config[23]==2)

{led2display2();}

}

}

void led2off(void)

{

_setcolor(0);

```

```

_ellipse(_G_FILLINTERIOR,262,103,296,137);

datob=0x00;

ledoffdisplay();

ledb=0;

}

void led2display1(void)

{

for(posicion=1;posicion<=posi[1];posicion++)

{

tabla();

_setcolor(config[9]);

_rectangle(_G_FILLINTERIOR,259,a,299,b);

}

/*

if(!(ponerdia==0))

{

*/

posi[1]++;

ledb=1;

/*

}

*/

}

void led2display2(void)

{

if(!(ponerdia==0))

```

```

{
posi[1]++;
ledb=1;
}
}

void led3on(void)
{
_setcolor(config[7]);
_ellipse(_G_FILLINTERIOR,308,100,348,140);
datoc=0x20;
if(ledc==0)
{
if(config[23]==1)
{led3display1();}
if(config[23]==2)
{led3display2();}
}
}

void led3off(void)
{
_setcolor(0);
_ellipse(_G_FILLINTERIOR,311,103,345,137);
datoc=0x00;
ledoffdisplay();
ledc=0;
}

```

```

void led3display1(void)

{

for(posicion=1;posicion<=posi[2];posicion++)

{

tabla();

_setcolor(config[9]);

_rectangle(_GFILLINTERIOR,308,a,348,b);

}

/*

if(!(ponerdia==0))

{

*/

posi[2]++;

ledc=1;

/*

}

*/

}

void led3display2(void)

{

if(!(ponerdia==0))

{

posi[2]++;

ledc=1;

}

}

```

```

void led4on(void)

{

_setcolor(config[7]);

_ellipse(_G_FILLINTERIOR,357,100,397,140);

datod=0x10;

if(ledd==0)

{

if(config[23]==1)

{led4display1();}

if(config[23]==2)

{led4display2();}

}

}

void led4off(void)

{

_setcolor(0);

_ellipse(_G_FILLINTERIOR,360,103,394,137);

datod=0x00;

ledoffdisplay();

ledd=0;

}

void led4display1(void)

{

for(posicion=1;posicion<=posi[3];posicion++)

{

tabla();

```

```

_setcolor(config[9]);

_rectangle(_G_FILLINTERIOR,357,a,397,b);

}

/*

if(!(ponerdia==0))

{

*/

posi[3]++;

ledd=1;

/*

}

*/

}

void led4display2(void)

{

if(!(ponerdia==0))

{

posi[3]++;

ledd=1;

}

}

void led5on(void)

{

_setcolor(config[7]);

_ellipse(_G_FILLINTERIOR,406,100,446,140);

datoe=0x08;

```

```

if(led==0)

{

if(config[23]==1)

{led5display1();}

if(config[23]==2)

{led5display2();}

}

}

void led5off(void)

{

_setcolor(0);

_ellipse(_G_FILLINTERIOR,409,103,443,137);

datoe=0x00;

ledoffdisplay();

led=0;

}

void led5display1(void)

{

for(posicion=1;posicion<=posi[4];posicion++)

{

tabla();

_setcolor(config[9]);

_rectangle(_G_FILLINTERIOR,406,a,446,b);

}

/*

if(!(ponerdia==0))

```

```

{
*/

posi[4]++;

lede=1;

/*

}

*/

}

void led5display2(void)

{

if(!(ponerdia==0))

{

posi[4]++;

lede=1;

}

}

void led6on(void)

{

_setcolor(config[7]);

_ellipse(_GFillInterior,455,100,495,140);

datof=0x04;

if(ledf==0)

{

if(config[23]==1)

{led6display1();}

if(config[23]==2)

```

```

{led6display2();}

}

}

void led6off(void)

{

  _setcolor(0);

  _ellipse(_GFILLINTERIOR,458,103,492,137);

  datof=0x00;

  ledoffdisplay();

  ledf=0;

}

void led6display1(void)

{

  for(posicion=1;posicion<=posi[5];posicion++)

  {

    tabla();

    _setcolor(config[9]);

    _rectangle(_GFILLINTERIOR,455,a,495,b);

  }

  /*

  if(!(ponerdia==0))

  {

    */

    posi[5]++;

    ledf=1;

    /*

```

```

}

*/

}

void led6display2(void)

{

if(!(ponerdia==0))

{

posi[5]++;

ledf=1;

}

}

void led7on(void)

{

_setcolor(config[7]);

_ellipse(_GFillInterior,504,100,544,140);

datog=0x02;

if(ledg==0)

{

if(config[23]==1)

{led7display1();}

if(config[23]==2)

{led7display2();}

}

}

void led7off(void)

{

```

```

_setcolor(0);

_ellipse(_G_FILLINTERIOR,507,103,541,137);

datog=0x00;

ledoffdisplay();

ledg=0;

}

void led7display1(void)

{

for(posicion=1;posicion<=posi[6];posicion++)

{

tabla();

_setcolor(config[9]);

_rectangle(_G_FILLINTERIOR,504,a,544,b);

}

/*

if(!(ponerdia==0))

{

*/

posi[6]++;

ledg=1;

/*

}

*/

}

void led7display2(void)

{

```

```

if(!(ponerdia==0))

{

posi[6]++;

ledg=1;

}

}

void led8on(void)

{

_setcolor(config[7]);

_ellipse(_G_FILLINTERIOR,553,100,593,140);

datoh=0x01;

if(ledh==0)

{

if(config[23]==1)

{led8display1();}

if(config[23]==2)

{led8display2();}

}

}

void led8off(void)

{

_setcolor(0);

_ellipse(_G_FILLINTERIOR,556,103,590,137);

datoh=0x00;

ledoffdisplay();

ledh=0;

```

```

}

void led8display1(void)

{

for(posicion=1;posicion<=posi[7];posicion++)

{

tabla();

_setcolor(config[9]);

_rectangle(_GFILLINTERIOR,553,a,593,b);

}

/*

if(!(ponerdia==0))

{

*/

posi[7]++;

ledh=1;

/*

}

*/

}

void led8display2(void)

{

if(!(ponerdia==0))

{

posi[7]++;

ledh=1;

}

}

```

```

}

void ledoffdisplay()

{
    _setcolor(config[0]);
    _rectangle(_G_FILLINTERIOR,0,0,1,480);
}

void codigoout(void)

{
    /* En esta funcion se recoge el dato enviado por cada led y se opera
    realizando una suma hexadecimal y metiendo el resultado en una variable
    la cual sera el dato que introducimos en la salida */
    datosresul=datoa+datob+datoc+datod+datoe+datof+datog+datoh;
    _settextposition(3,42);
    printf("SAL.HEXA=%x ",datosresul);
    vel=vel+m;
    printf(" VEL=%.2f",vel);
    outp(0x37A,0);
    outp(0x378,datosresul);
    outp(0x37A,1);
}

void entrada(void)

{
    /* En esta funcion se realiza la entrada de datos del prototipo a el
    programa que son tratados con una mascara para luego representarlo
    por pantalla */
    int i;

```

```

unsigned char entradaa[2],entradab[2],mascara;

outp(0x37A,2);

entradaa[0]=inp(0x379);

outp(0x37A,0);

entradaa[1]=inp(0x379);

mascara=0xf0;

_settextposition(4,20);

printf("LA 0=%x ",entradaa[0]);

_settextposition(5,20);

printf("LA 1=%x ",entradaa[1]);

_settextposition(6,20);

printf("LA MASCARA=%x ",mascara);

getch();

entradaa[0]=entradaa[0]&mascara;

entradaa[1]=entradaa[1]&mascara;

_settextposition(7,20);

printf("LA 0 CON MASCARA=%x ",entradaa[0]);

_settextposition(8,20);

printf("LA 1 CON MASCARA=%x ",entradaa[1]);

getch();

for(i=0;i<2;i++)

{

if(entradaa[i]==0xb0)entradab[i]=0xf0;

if(entradaa[i]==0xf0)entradab[i]=0xe0;

if(entradaa[i]==0x30)entradab[i]=0xd0;

if(entradaa[i]==0x70)entradab[i]=0xc0;

```

```

if(entrada[i]==0x90)entradab[i]=0xb0;

if(entrada[i]==0xd0)entradab[i]=0xa0;

if(entrada[i]==0x10)entradab[i]=0x90;

if(entrada[i]==0x50)entradab[i]=0x80;

if(entrada[i]==0xa0)entradab[i]=0x70;

if(entrada[i]==0xe0)entradab[i]=0x60;

if(entrada[i]==0x20)entradab[i]=0x50;

if(entrada[i]==0x60)entradab[i]=0x40;

if(entrada[i]==0x80)entradab[i]=0x30;

if(entrada[i]==0xc0)entradab[i]=0x20;

if(entrada[i]==0x00)entradab[i]=0x10;

if(entrada[i]==0x40)entradab[i]=0x00;

}

organi();

_settextposition(9,20);

printf("LA 0 CON MASCARA2=%x ",entradab[0]);

_settextposition(10,20);

printf("LA 1 CON MASCARA2=%x ",entradab[1]);

if(entradab[0]>=0x80){entradab[0]=entradab[0]-0x80;led1on();}

if(entradab[0]>=0x40){entradab[0]=entradab[0]-0x40;led2on();}

if(entradab[0]>=0x20){entradab[0]=entradab[0]-0x20;led3on();}

if(entradab[0]>=0x10){entradab[0]=entradab[0]-0x10;led4on();}

if(entradab[1]>=0x80){entradab[1]=entradab[1]-0x80;led5on();}

if(entradab[1]>=0x40){entradab[1]=entradab[1]-0x40;led6on();}

if(entradab[1]>=0x20){entradab[1]=entradab[1]-0x20;led7on();}

if(entradab[1]>=0x10){entradab[1]=entradab[1]-0x10;led8on();}

```

```

codigoout();

getch();

inicio2();

inicio3();

menu_principal();

}

void efecto1(void)

{

/* En esta funcion se introducen los datos del efecto 1 en le array
usuario para luego ejecutarlo en la funcion ejecutar */

int t;

organi();

paso=26;

for(t=0;t<8;t++)

{usuario[0][t]=0;

if(t==0)usuario[0][t]=1;}

for(t=0;t<8;t++)

{usuario[1][t]=0;

if(t==1)usuario[1][t]=1;}

for(t=0;t<8;t++)

{usuario[2][t]=0;

if(t==2)usuario[2][t]=1;}

for(t=0;t<8;t++)

{usuario[3][t]=0;

if(t==3)usuario[3][t]=1;}

for(t=0;t<8;t++)

```

```
{usuario[4][t]=0;

if(t==4)usuario[4][t]=1;}

for(t=0;t<8;t++)

{usuario[5][t]=0;

if(t==5)usuario[5][t]=1;}

for(t=0;t<8;t++)

{usuario[6][t]=0;

if(t==6)usuario[6][t]=1;}

for(t=0;t<8;t++)

{usuario[7][t]=0;

if(t==7)usuario[7][t]=1;}

for(t=0;t<8;t++)

{usuario[8][t]=0;

if(t==7)usuario[8][t]=1;}

for(t=0;t<8;t++)

{usuario[9][t]=0;

if(t==6)usuario[9][t]=1;}

for(t=0;t<8;t++)

{usuario[10][t]=0;

if(t==5)usuario[10][t]=1;}

for(t=0;t<8;t++)

{usuario[11][t]=0;

if(t==4)usuario[11][t]=1;}

for(t=0;t<8;t++)

{usuario[12][t]=0;

if(t==3)usuario[12][t]=1;}
```

```

for(t=0;t<8;t++)

{usuario[13][t]=0;

if(t==2)usuario[13][t]=1;}

for(t=0;t<8;t++)

{usuario[14][t]=0;

if(t==1)usuario[14][t]=1;}

for(t=0;t<8;t++)

{usuario[15][t]=0;

if(t==0)usuario[15][t]=1;}

for(t=0;t<8;t++)

{usuario[16][t]=0;

if(t==0)usuario[16][t]=1;}

for(t=0;t<8;t++)

{usuario[17][t]=0;

if(t==3||t==4)usuario[17][t]=1;}

for(t=0;t<8;t++)

{usuario[18][t]=0;

if(t==2||t==5)usuario[18][t]=1;}

for(t=0;t<8;t++)

{usuario[19][t]=0;

if(t==1||t==6)usuario[19][t]=1;}

for(t=0;t<8;t++)

{usuario[20][t]=0;

if(t==0||t==7)usuario[20][t]=1;}

for(t=0;t<8;t++)

{usuario[21][t]=0;

```

```

if(t==0||t==7)usuario[21][t]=1;}

for(t=0;t<8;t++)

{usuario[22][t]=0;

if(t==1||t==6)usuario[22][t]=1;}

for(t=0;t<8;t++)

{usuario[23][t]=0;

if(t==2||t==5)usuario[23][t]=1;}

for(t=0;t<8;t++)

{usuario[24][t]=0;

if(t==3||t==4)usuario[24][t]=1;}

ejecucion();

}

void efecto2(void)

{

/* En esta funcion se introducen los datos en el array del efecto 2

para luego ejecutarlo en la funcion ejecutar */

int t;

paso=18;

organi();

for(t=0;t<8;t++)

{usuario[0][t]=0;

if(t==0||t==2||t==4||t==6)usuario[0][t]=1;}

for(t=0;t<8;t++)

{usuario[1][t]=0;

if(t==1||t==3||t==5||t==7)usuario[1][t]=1;}

for(t=0;t<8;t++)

```

```

{usuario[2][t]=0;

if(t==0||t==2||t==4||t==6)usuario[2][t]=1;}

for(t=0;t<8;t++)

{usuario[3][t]=0;

if(t==0||t==7)usuario[3][t]=1;}

for(t=0;t<8;t++)

{usuario[4][t]=0;

if(t==1||t==6)usuario[4][t]=1;}

for(t=0;t<8;t++)

{usuario[5][t]=0;

if(t==2||t==5)usuario[5][t]=1;}

for(t=0;t<8;t++)

{usuario[6][t]=0;

if(t==3||t==4)usuario[6][t]=1;}

for(t=0;t<8;t++)

{usuario[7][t]=0;

if(t==0)usuario[7][t]=1;}

for(t=0;t<8;t++)

{usuario[8][t]=0;

if(t==7)usuario[8][t]=1;}

for(t=0;t<8;t++)

{usuario[9][t]=0;

if(t==1)usuario[9][t]=1;}

for(t=0;t<8;t++)

{usuario[10][t]=0;

if(t==6)usuario[10][t]=1;}

```

```

for(t=0;t<8;t++)

{usuario[11][t]=0;

if(t==2)usuario[11][t]=1;}

for(t=0;t<8;t++)

{usuario[12][t]=0;

if(t==5)usuario[12][t]=1;}

for(t=0;t<8;t++)

{usuario[13][t]=0;

if(t==3)usuario[13][t]=1;}

for(t=0;t<8;t++)

{usuario[14][t]=0;

if(t==4)usuario[14][t]=1;}

for(t=0;t<8;t++)

{usuario[15][t]=0;

if(t==0||t==1||t==2||t==3||t==4||t==5||t==6||t==7)usuario[15][t]=1;}

for(t=0;t<8;t++)

{usuario[16][t]=0;

}

for(t=0;t<8;t++)

{usuario[17][t]=0;

if(t==0||t==1||t==2||t==3||t==4||t==5||t==6||t==7)usuario[17][t]=1;}

ejecucion();

}

void definido_usuario(void)

{

int contador,camb;

```

```

crear_window(150,95,550,400,'a');

_outgtext("EFECTO DEFINIDO POR EL USUARIO");

_setcolor(15);

_moveto(200,140);

_outgtext(" CREACION MANUAL ");

_moveto(200,160);

_outgtext(" SELECCIONA EL NUMERO DE PASOS ");

_moveto(200,180);

_outgtext(" DEL EFECTO(60 MAX):");

_settextposition(11,44);

scanf("%d",&paso);

fflush(stdin);

for(contador=0;contador<paso;contador++)

{

crear_window(150,95,505,400,'a');

_outgtext("EFECTO DEFINIDO POR EL USUARIO");

_setcolor(15);

_moveto(200,140);

_outgtext(" CREACION MANUAL ");

_moveto(200,160);

_outgtext(" LED APAGADO=0 ");

_moveto(200,180);

_outgtext(" LED ENCENDIDO=1 ");

_moveto(200,200);

_outgtext(" VOLVER=2 ");

_settextposition(15,30);

```

```

printf(" PASO %d ",contador+1);

/*Por razon inexplicable se le tiene que restar 48 al array usuario[60][8]*/

do{

    _moveto(200,240);

    _outgtext(" LED 1=");

    _setttextposition(17,23);

    usuario[contador][0]=(getche()-48);

    fflush(stdin);} while(usuario[contador][0]<0||usuario[contador][0]>2);

    if(usuario[contador][0]==2)

    { camb=1;break;}

do{

    _moveto(200,260);

    _outgtext(" LED 2=");

    _setttextposition(18,23);

    usuario[contador][1]=(getche()-48);

    fflush(stdin);} while(usuario[contador][1]<0||usuario[contador][1]>2);

    if(usuario[contador][1]==2)

    { camb=1;break;}

do{

    _moveto(200,280);

    _outgtext(" LED 3=");

    _setttextposition(19,23);

    usuario[contador][2]=(getche()-48);

    fflush(stdin);} while(usuario[contador][2]<0||usuario[contador][2]>2);

    if(usuario[contador][2]==2)

    { camb=1;break;}

```

```

do{

_moveto(200,300);

_outgtext(" LED 4=");

_settextposition(20,23);

usuario[contador][3]=(getche()-48);

fflush(stdin);} while(usuario[contador][3]<0||usuario[contador][3]>2);

if(usuario[contador][3]==2)

{ camb=1;break;}

do{

_moveto(200,320);

_settextposition(21,23);

_outgtext(" LED 5=");

_settextposition(21,23);

usuario[contador][4]=(getche()-48);

fflush(stdin);} while(usuario[contador][4]<0||usuario[contador][4]>2);

if(usuario[contador][4]==2)

{ camb=1;break;}

do{

_moveto(200,340);

_outgtext(" LED 6=");

_settextposition(22,23);

usuario[contador][5]=(getche()-48);

fflush(stdin);} while(usuario[contador][5]<0||usuario[contador][5]>2);

if(usuario[contador][5]==2)

{ camb=1;break;}

do{

```

```

_moveto(200,360);

_outgtext(" LED 7=");

_settextposition(23,23);

usuario[contador][6]=(getche()-48);

fflush(stdin);} while(usuario[contador][6]<0||usuario[contador][6]>2);

if(usuario[contador][6]==2)

{ camb=1;break;}

do{

_moveto(200,380);

_outgtext(" LED 8=");

_settextposition(24,23);

usuario[contador][7]=(getche()-48);

fflush(stdin);} while(usuario[contador][7]<0||usuario[contador][7]>2);

if(usuario[contador][7]==2)

{ camb=1;break;}

}

if(camb==1)

{ camb=2;definido_usuario();}

creamanual2();

}

void creamanual2(void)

{

crear_window(150,95,550,400,'a');

_outgtext("EFECTO DEFINIDO POR EL USUARIO");

_setcolor(15);

_moveto(200,140);

```

```

_outgtext(" CREACION MANUAL ");

_moveto(200,160);

_outgtext(" COMPLETADA ");

crear_boton(400,150,500,175,'d');

_outgtext("EJUCUTAR");

ejecuta='0';

estoyen=3;

ventana_raton(150,95,550,180);

mostrar_raton();

do{

if(kbhit()){ejecuta=getch();}

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

}while(ejecuta=='0');

quitar_raton();

switch(ejecuta)

{

case 'e':

case 'E':inicio1();inicio2();inicio3();histograma_graf();mostrar_raton();ejecucion();break;

default:aleatorio2();break;

}

}

void aleatorio(void)

{

crear_window(150,95,550,180,'a');

_outgtext("CREACION ALEATORIA DEL EFECTO");

```

```

_setcolor(15);

_moveto(200,140);

_outgtext(" SELECCIONA EL NUMERO DE PASOS");

_moveto(200,160);

_outgtext(" DEL EFECTO(60 MAX):");

_settextposition(11,44);

scanf("%d",&paso);

fflush(stdin);

creacionaleatoria();

}

void aleatorio2(void)

{

crear_window(150,95,550,180,'a');

_outgtext("CREACION ALEATORIA DEL EFECTO");

_setcolor(15);

_moveto(200,140);

_outgtext(" CREACION ALEATORIA ");

_moveto(200,160);

_outgtext(" COMPLETADA ");

crear_boton(400,150,500,175,'d');

_outgtext("EJUCUTAR");

ejecuta='0';

estoyen=3;

ventana_raton(150,95,550,180);

mostrar_raton();

do{

```

```

if(kbhit()){ejecuta=getch();}

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

}while(ejecuta=='0');

quitar_raton();

switch(ejecuta)

{

case 'e':

case 'E':inicio1();inicio2();inicio3();histograma_graf();mostrar_raton();ejecucion();break;

default:aleatorio2();break;

}

}

void creacionaleatoria(void)

{

int x,y;

for(x=0;x<pasos;x++)

{

for(y=0;y<8;y++)

{

usuario[x][y]=rand() %2;

}

}

aleatorio2();

}

void pulsos(void)

{

```

```

/* printf("pulsos");*/

}

void combinatorio(void)

{

/* printf("combinatorio");*/

}

void reseteo(void)

{

reset=1;

ejecutar=0;

principal();

}

void ejecucion(void)

{

int i,j,pin;

organi();

ventana_raton(200,160,600,280);

ejecutar=1;

retardo(1);

for(i=0;i<paso;i++)

{

for(j=0;j<8;j++)

{

switch(usuario[i][j])

{

case 0:

```

```
{  
if(j==0)  
{led1off();break;}  
if(j==1)  
{led2off();break;}  
if(j==2)  
{led3off();break;}  
if(j==3)  
{led4off();break;}  
if(j==4)  
{led5off();break;}  
if(j==5)  
{led6off();break;}  
if(j==6)  
{led7off();break;}  
if(j==7)  
{led8off();break;}  
}  
case 1:  
{  
if(j==0)  
{led1on();break;}  
if(j==1)  
{led2on();break;}  
if(j==2)  
{led3on();break;}
```

```

if(j==3)

{led4on();break;}

if(j==4)

{led5on();break;}

if(j==5)

{led6on();break;}

if(j==6)

{led7on();break;}

if(j==7)

{led8on();break;}

}

}

}

codigoout();

retardo(vel);

}

apagadoleds();

reset=0;

principal();

}

void apagadoleds(void)

{

led1off();

led2off();

led3off();

led4off();

```

```

led5off();

led6off();

led7off();

led8off();

}

void cerrar(void)

{

crear_window(155,210,455,300,'a');

_outgtext("INTERFAZ GRAFICO");

crear_boton(217,270,252,290,'d');

_outgtext("Si");

crear_boton(267,270,302,290,'d');

_outgtext("No");

crear_boton(317,270,392,290,'d');

_outgtext("Cancelar");

_moveto(175,250);

_outgtext("¿ESTA SEGURO DE QUE QUIERE SALIR?");

cerrarmenu();

}

void cerrarmenu()

{

ventana_raton(155,210,445,287);

poner_raton_en(284,280);

menu='0';

estoyen=1;

do{

```

```

mostrar_raton();

if(kbhit()){menu=getch();}

tecla_iz();

if(t_izq!=0){pos_raton();comprobacion_bizquierdo();}

}while(menu=='0');

switch(menu)

{

case 's':

case 'S':_clearscreen(_GCLEARSCREEN);break;

case 'c':

case 'C':

case 'n':

case 'N':principal();break;

default:cerrar();break;

}

exit(1);

}

```

GRAFICOS.H

```
#include"boton.h"
```

```
#include"ventana.h"
```

```
#include"lineas.h"
```

```
#include"leds.h"
```

VENTANA.H

```
#include"stdio.h"
```

```
#include"graph.h"
```

```
void crear_window(int winxa,int winya,int winxb,int winyb,char con_boton_caracteristico)
```

```

{
    _setcolor(config[3]);

    _rectangle(_G_FILLINTERIOR,winxa,winya,winxb,winyb);

    _setcolor(config[2]);

    _rectangle(_G_FILLINTERIOR,winxa+3,winya+3,winxb-3,winya+23);

    _setcolor(15);

    _rectangle(_G_BORDER,winxa,winya,winxb,winyb);

    _rectangle(_G_BORDER,winxa+3,winya+3,winxb-3,winya+23);

    _setcolor(0);

    _moveto(winxa+3,winya+23);

    _lineto(winxb-3,winya+23);

    _lineto(winxb-3,winya+3);

    _moveto(winxa,winyb);

    _lineto(winxb,winyb);

    _lineto(winxb,winya);

    if(con_boton_caracteristico=='c')
    {
        /* CUADRO DE CERRAR */

        _setfont("t'helv'h8w8b");

        _setcolor(7);

        _rectangle(_G_FILLINTERIOR,winxb-24,winya+7,winxb-12,winya+19);

        _setcolor(15);

        _rectangle(_G_BORDER,winxb-24,winya+7,winxb-12,winya+19);

        _setcolor(0);

        _moveto(winxb-24,winya+19);

        _lineto(winxb-12,winya+19);
    }
}

```

```

_lineto(winxb-12,winya+7);

_moveto(winxb-23,winya+7);

_outgtext("X");

_moveto(winxb-22,winya+7);

_outgtext("X");

_moveto(winxb-21,winya+7);

_outgtext("X");

}

/* TEXTO DE PANTALLA */

_setfont("t'helv'h8w8b");

_setcolor(config[4]);

_moveto(winx+23,winya+8);

}

```

RATON.H

```

#include "dos.h"

#include "stdio.h"

union REGS regs;

void tipo_cursor(char caracter,char atributos)

{

regs.x.ax=0x000A;

regs.x.bx=0x1;

regs.x.cx=caracter;

regs.x.dx=atributos;

int86(0x33,&regs,&regs);

}

int raton_instalado(void)

```

```

{

regs.x.ax=0x00;

int86(0x33,&regs,&regs);

if(regs.x.ax==0)

return(0);

else

return(1);

}

void inicializa_raton(void)

{

regs.x.ax=0x00;

int86(0x33,&regs,&regs);

}

void mostrar_raton(void)

{

regs.x.ax=0x01;

int86(0x33,&regs,&regs);

}

void quitar_raton(void)

{

regs.x.ax=0x02;

int86(0x33,&regs,&regs);

}

char t_izq;

void tecla_iz(void)

{

```

```

regs.x.ax=0x03;

int86(0x33,&regs,&regs);

t_izq=((regs.x.bx)&(0x01));

}

char t_der;

void tecla_der(void)

{

regs.x.ax=0x03;

int86(0x33,&regs,&regs);

t_der=((regs.x.bx)&(0x02));

}

char t_cen;

void tecla_cen(void)

{

regs.x.ax=0x03;

int86(0x33,&regs,&regs);

t_cen=((regs.x.bx)>(0x02));

}

void no_tecla_iz(void)

{

regs.x.ax=0x06;

regs.x.bx=0x00;

int86(0x33,&regs,&regs);

if((regs.x.bx)>0)

return(1);

else

```

```

return(0);

}

void poner_raton_en(int x,int y)

{

regs.x.ax=0x04;

regs.x.cx=x;

regs.x.dx=y;

int86(0x33,&regs,&regs);

}

int dx,dy;

void pos_raton()

{

regs.x.ax=0x03;

int86(0x33,&regs,&regs);

dx=regs.x.cx;

dy=regs.x.dx;

}

void ventana_raton(int left,int top,int right,int bottom)

{

regs.x.ax=0x08;

regs.x.cx=top;

regs.x.dx=bottom;

int86(0x33,&regs,&regs);

regs.x.ax=0x07;

regs.x.cx=left;

regs.x.dx=right;

```

```
int86(0x33,&regs,&regs);
```

```
}
```

Este programa está desarrollado en Quick C++ 2.2 de Microsoft

El ejecutable puede dar problemas de velocidad pero una

vez compilado desaparecen.

Este programa ha sido desarrollado por Andres Rodriguez Sorribe

para la asignatura de desarrollo de prototipos del instituto

Julio Verne, para el Ciclo Formativo de Grado Superior de Desarrollo de

Productos Electronicos, la nota que supuso este trabajo fue 10.

Este programa está casi completo, lo único que le falta es

depurarlo un poco ya que en un principio tenía más opciones que luego

por falta de memoria las suprimí. El botón de Entrada está casi terminado

y me consta que con este método se puede sacar ya que el último día en

clase me funcionó cambiando un par de cosas que creo haber cambiado ya

en este programa.

Mando una imagen del esquema electrónico del prototipo que

controla con las salidas del puerto paralelo.

Todo lo que está entre /* */ puede ser borrado.

