

Mover un motor

Para llevar a cabo este ejemplo lo dividiremos en las siguientes partes:

- Conexiones
- Diagramas de flujo
- Programación en MSWLogo
- Programación en C

Conexiones

1.- Necesitamos un motor pequeño de tensión baja. Uno de 6 V, 12 V o similar servirá. Tendremos que tener conectados un cable a cada polo del motor.

Las conexiones que se establecerán entre el motor y la tarjeta controladora CNICE se muestran en la siguiente tabla:

CONECTOR	SALIDA DIGITAL
0	Cable del polo positivo
1	Cable del polo negativo

2.- Si la conexión se hace al revés, es decir, se conecta el cable conectado al polo negativo del motor a la salida digital 0 y el otro cable a la salida 1 se invertirá la polaridad y hará que el motor gire en sentido contrario al de las conexiones anteriores.

3.- Se puede usar cualquiera de las otras salidas digitales para realizar la conexión entre el motor y la controladora.

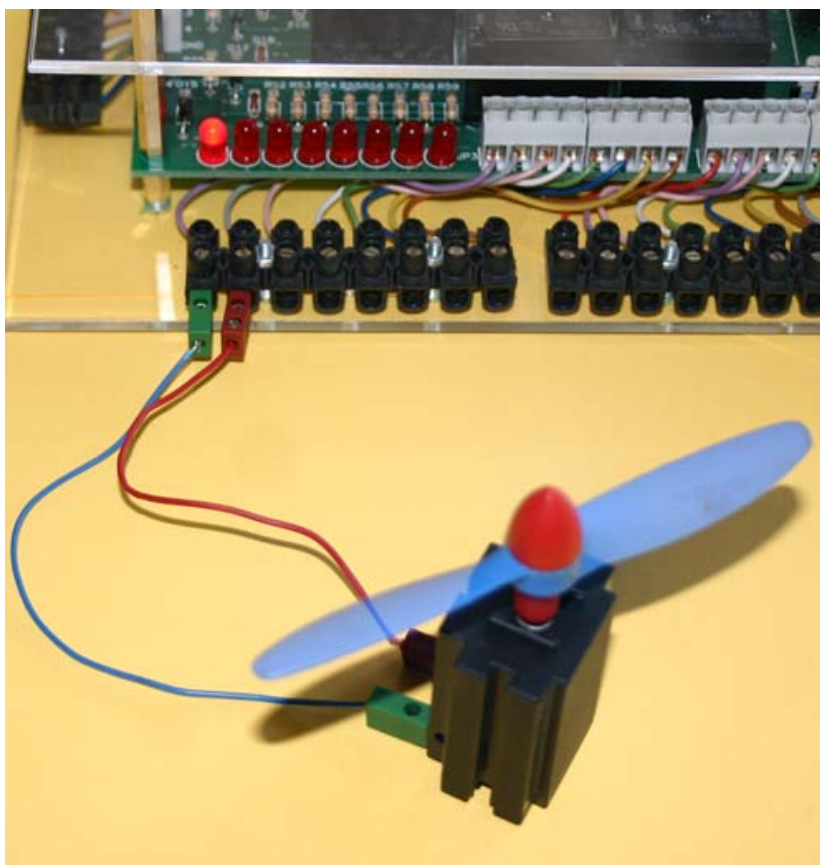
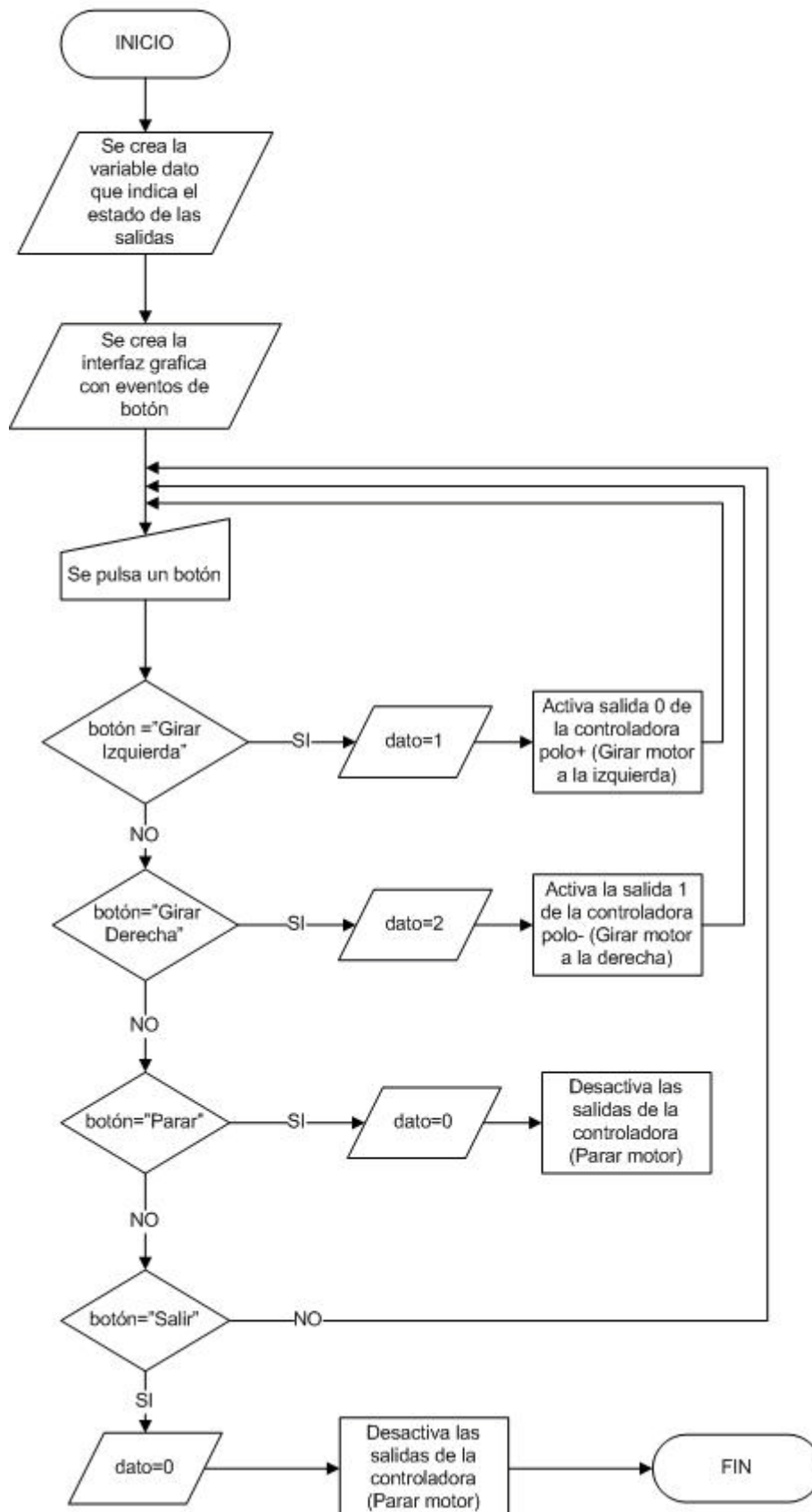


Diagrama de Flujo

MOTOR



Programación en MSWLogo

La programación de este caso se estructura en los siguientes pasos:

1.- Se crea un procedimiento para crear la ventana gráfica del programa. Para crear la ventana utilizamos la función **creaventana**. Para crear un marco dentro de la ventana se utiliza la función **creagroupbox**. Por último dentro del marco se crean los botones con la función **creaboton**. Dentro de cada botón se establecerán entre los corchetes las funciones que se han de ejecutar una vez presionado. Además se mostrarán las imágenes del motor con **cargadib**.

Ejemplo:

```
creaventana "trabajo "ControlMotor [MOTOR] 40 30 150 90 []

creagroupbox "ControlMotor "E_Ventana 5 0 140 60

creaboton "ControlMotor "Apagar "Parar 60 40 35 10 [ApagarMotor]

creaboton "ControlMotor "Izquierda "Izquierda 20 22 35 10 [MotorIzq]

creaboton "ControlMotor "Derecha "Derecha 100 22 35 10 [MotorDer]

creaboton "ControlMotor "Salir "SALIR 60 65 35 10 [SalirMotor]

cargadib "Elice3.bmp
```

2.- Se crean varios procedimientos para que el motor gire a izquierda y a derecha y para desactivar el motor. En estos procedimientos se va a utilizar la función **M** seguida del número que indique el par de salidas digitales a activar, y tras esto se pondrán unas comillas con la letra D o I que indicará que el motor gire a la derecha (activa la salida 1 y desactiva la salida 0) o que el motor gire a izquierda (activa la salida 0 y desactiva la salida 1), con la letra P se le indica al motor que se detenga (desactiva las 2 salidas digitales).

Ejemplo:

```
para ApagarMotor

cargadib "Elice3.bmp

M1 "P

fin

para MotorIzq

cargadib "Elice3.bmp

cargadib "Elice2.bmp

M1 "I

fin

para MotorDer

cargadib "Elice3.bmp

cargadib "Elice4.bmp
```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

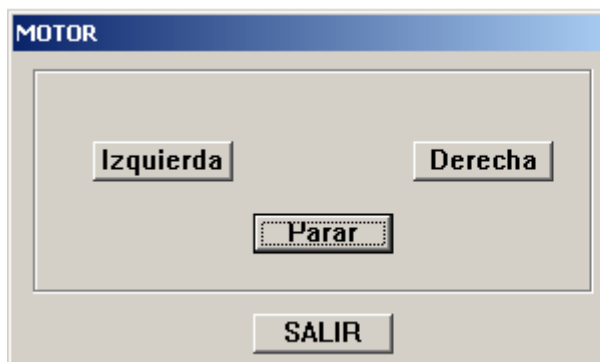
M1 "D

Fin

3.- Se llama al procedimiento que crea la ventana gráfica fuera de cualquier procedimiento, para que se cargue la aplicación gráfica nada más cargar el fichero logo.

Descargue el archivo programado en **MSWLogo**, descomprímalo y guárdelo en un directorio aparte. Contiene el fichero de código en MSWLogo motor.lgo y las imágenes. Ejecute el compilador MSWLogo versión 6.5a en castellano. Vaya al menú del programa, Archivo/Abrir y seleccione el fichero motor.lgo que se descargó previamente.

Se visualizará una pantalla donde pulsando en los botones correspondientes podrá hacer que el motor gire a derecha o izquierda o que se pare.



Programación en C

La programación de este caso se estructura en los siguientes pasos:

- 1.- Se crea un nuevo proyecto.
- 2.- Se añaden al proyecto los archivos io.h, io.cpp, Primitivas_CNICE.CPP y Primitivas_CNICE.HPP y SDL.h (librería que permite añadir imágenes a la aplicación programada con C).
- 3.- Se crea el archivo main.c donde se incluirán las funciones necesarias para crear las ventanas
- 4.- Dentro del archivo main.c creado anteriormente, se añade la declaración a las funciones de la biblioteca io.dll de la siguiente manera:

```
#include "io.h"
```

También se añade la declaración a las funciones de la biblioteca SDL.dll de la siguiente manera:

```
#include <SDL.h>
```

- 5.- En nuestro archivo main.c se define la siguiente función que permite activar o desactivar las entradas digitales de la controladora.

```
void Escribir_Salidas_Digitales(int Dato)
{
    LoadIODLL();
    PortOut(0x37A,0x7);
    PortOut(0x378,Dato);
}
```

6.- Se crearán tres botones en nuestra ventana, uno para hacer girar el motor a la izquierda, otro para hacerlo girar a la derecha y otro para hacer parar el motor, y en cada uno de ellos se incluye la llamada a la función anterior para cambiar el estado de las salidas de la siguiente manera:

```
switch(LOWORD(wParam))
{
    case 1: // botón Girar Izquierda
        sinluz = SDL_LoadBMP("Elice4.bmp");
        screen = SDL_SetVideoMode( 200, 150, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
            return -1;
        }

        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(sinluz, NULL,screen,&rect);
        SDL_Flip(screen);
        Escribir_Salidas_Digitales(1); //Activa la salida 1
        break;
    case 2: // botón Parar
        sinluz = SDL_LoadBMP("Elice2.bmp");
        screen = SDL_SetVideoMode( 200, 150, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
            return -1;
        }

        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(sinluz, NULL,screen,&rect);
        SDL_Flip(screen);
        Escribir_Salidas_Digitales(0); //Desactiva las salidas
        break;
    case 3: // botón Girar Derecha
        sinluz = SDL_LoadBMP("Elice3.bmp");
        screen = SDL_SetVideoMode( 200, 150, 0, SDL_NOFRAME );
        if( screen == NULL ) {
            printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
            SDL_Quit();
            return -1;
        }

        rect = (SDL_Rect) {0, 0, 200, 200};
        SDL_BlitSurface(sinluz, NULL,screen,&rect);
        SDL_Flip(screen);
        Escribir_Salidas_Digitales(2); //Activa la salida 2
        break;
    case 4: // botón Salir
        Escribir_Salidas_Digitales(0); //Desactiva las salidas
        SendMessage(hwnd, WM_CLOSE, 0, 0); //Finaliza la aplicación
        break;
    default:
        break;
}
```

Explicación de las funciones de la librería SDL.

Cada vez que se quiera mostrar una imagen se indicara en nuestro proyecto lo siguiente:

```
foto = SDL_LoadBMP("imagen.bmp");
screen = SDL_SetVideoMode(200, 308, 0, SDL_NOFRAME );
if( screen == NULL ) {
    printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
    SDL_Quit();
    return -1;
}
rect.x=0;
rect.y=0;
```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
rect.w=primera1->w;  
rect.h=primera1->h;  
destino.x=0;  
destino.y=0;  
SDL_BlitSurface(primera1, &rect, screen, &destino);  
SDL_Flip(screen);
```

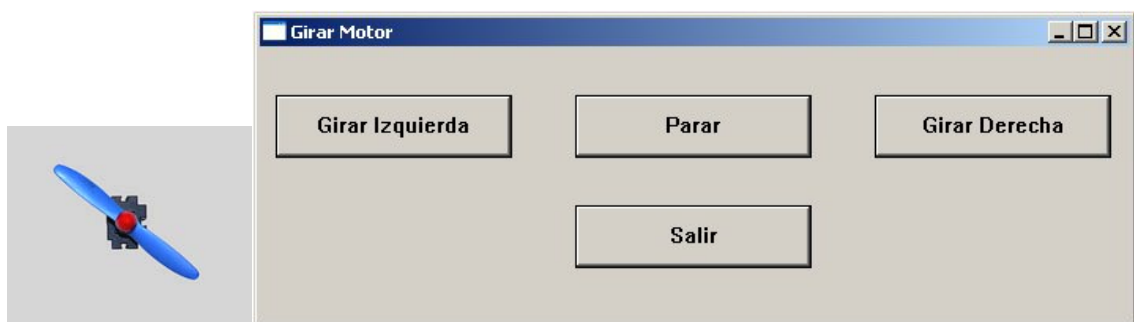
donde **foto** y **screen** son del tipo `SDL_Surface` y **rect** y **destino** es del tipo `SDL_Rect`.

- `SDL_LoadBMP`: carga la imagen .bmp que queramos
- `SDL_SetVideoMode` (int width, int height, int bpp, Uint32 flags): configure un modo de video con una anchura (width), una altura (height) y unos bits-por-pixeles. El parámetro flags indica el tipo de ventana que se quiere. En nuestro caso una ventana sin titulo no borde.
- `SDL_BlitSurface`(imagen, &rect, screen, &destino): pega desde la imagen, la porción seleccionada por rect sobre la superficie screen en el destino indicado por destino.
- `SDL_Flip`(screen): muestra la imagen que se ha seleccionado.

7.- Una vez creados los botones con la función que les corresponden, se compila comprobando que no hay ningún error.

8.- Una vez que se ha comprobado que no hay ningún error en nuestro código, se ejecuta y se comprueba el funcionamiento del motor. Al ejecutar el proyecto se creará el fichero **motor.exe**

Descargue los diferentes archivos que forman todo el proyecto programado en C, descompríalos y guárdelos en un directorio aparte. Ejecute el fichero **motor.exe**. Se visualizará la siguiente pantalla:



Pulsando en los botones correspondientes podrá hacer que el motor gire a derecha o izquierda o que se pare.

Nota:

En la aplicación programada con C, la imagen puede no aparecer al lado de la ventana. En este caso basta con mover nuestra ventana y se verá correctamente la imagen.