

SEMÁFOROS

- **Introducción**
- **Construcción**
- **Esquema eléctrico**
- **Descripción de los semáforos**



INTRODUCCIÓN

En la presente documentación se van a desarrollar una serie de ejemplos para instruirse en el manejo de la controladora CNICE utilizando el lenguaje de programación Logo y empleando el compilador MSWLogo que distribuye el Instituto de Tecnologías Educativas a través de su Observatorio y el lenguaje C utilizando el compilador DevC++. Con ellos se tratará de que el "alumno" vaya aprendiendo de forma gradual a programar esta controladora con estos lenguajes, por ello, y como se podrá comprobar, hay ejemplos que se desarrollan en varias fases. No obstante, hay que destacar que se presuponen unos conocimientos previos en estos lenguajes de programación y en las primitivas que se emplean para la controladora.

En primer lugar se procederá a explicar en detalle cómo se ha construido el dispositivo que va a permitir probar los programas desarrollados.

Se explicará también en detalle en qué consiste el semáforo.

Por último se pasará a explicar el funcionamiento de cada uno de los ejemplos y las conexiones que se deben realizar en la controladora junto con el código correspondiente. Para obtener más información acerca de cómo se ha realizado la programación se debe consultar los amplios comentarios que se han introducido en el código de cada ejemplo.

CONSTRUCCIÓN

Este dispositivo simula el funcionamiento de un semáforo de tráfico, el cual va cambiando, siguiendo la siguiente secuencia de colores, que es:

- Rojo, Verde, Amarillo.

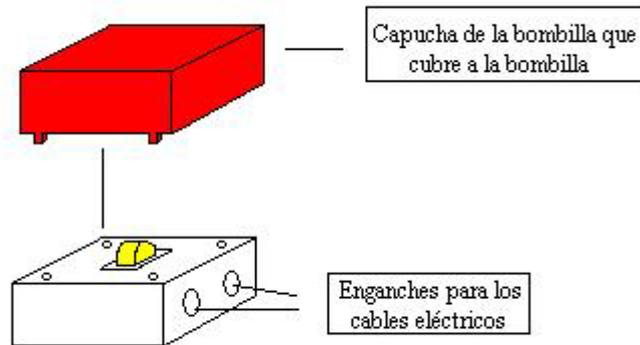
El dispositivo está compuesto por 3 bombillas, a cada una de las cuales se la va a colocar una capucha de color rojo, amarillo y verde.

Toda la estructura del semáforo se va a apoyar sobre un soporte (1), el cual va a estar compuesto por 4 bloques de 3 x 1'5 cm y 1 bloque de 1'5 x 1'5 cm. Los componentes que lo van a integrar son los siguientes:

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

Palo del semáforo (2): Es el palo sobre el cual se montarán las bombillas, éste está compuesto por 6 bloques de 3 x 1'5 cm y 1 bloque de 1'5 x 1'5 cm, éste último será el último del palo, y servirá de enganche con el soporte del semáforo.

3 Bombillas (3): Éstas van a estar compuestas por 2 partes, una parte es la propia bombilla en si, y la otra parte es la capucha de la bombilla, se utilizarán 3 capuchas diferentes, una para cada color del semáforo (rojo, amarillo, verde).



6 Cables eléctricos (4): Se utilizarán 2 cables para electrificar cada bombilla, uno de los cables irá a una salida digital, y el otro cable se conectará a la masa de la controladora.

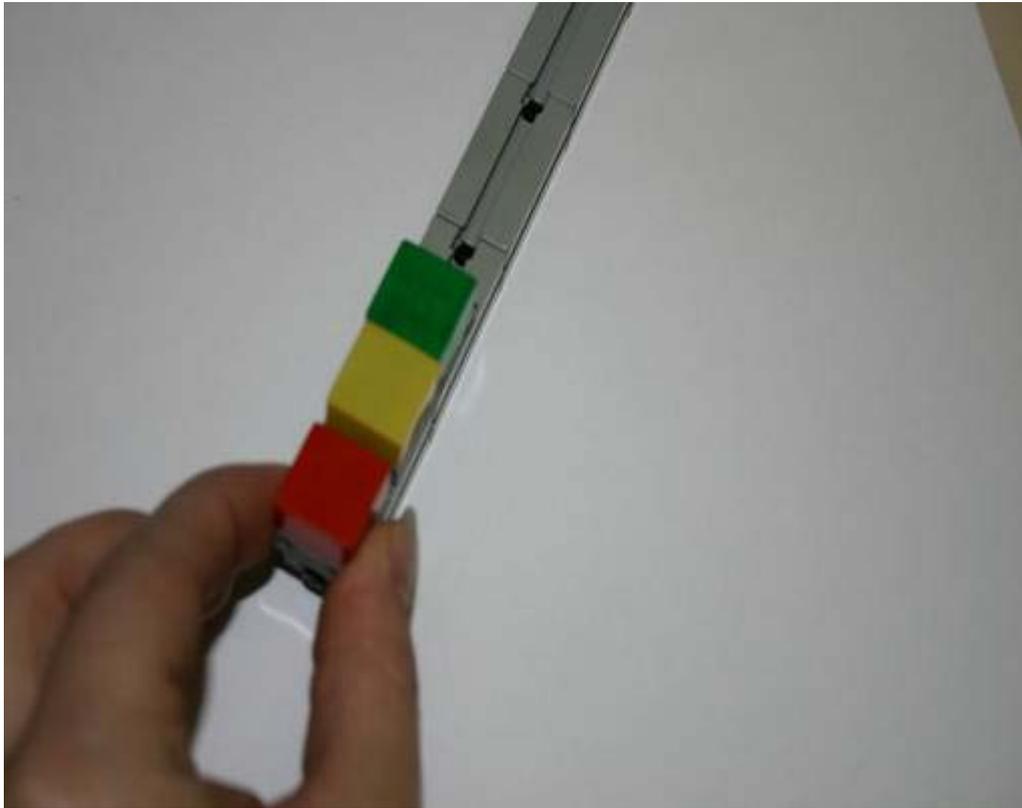
6 Enganches de cables eléctricos (5): Estos sirven para enganchar los cables eléctricos a las bombillas, que son la parte fundamental del semáforo.



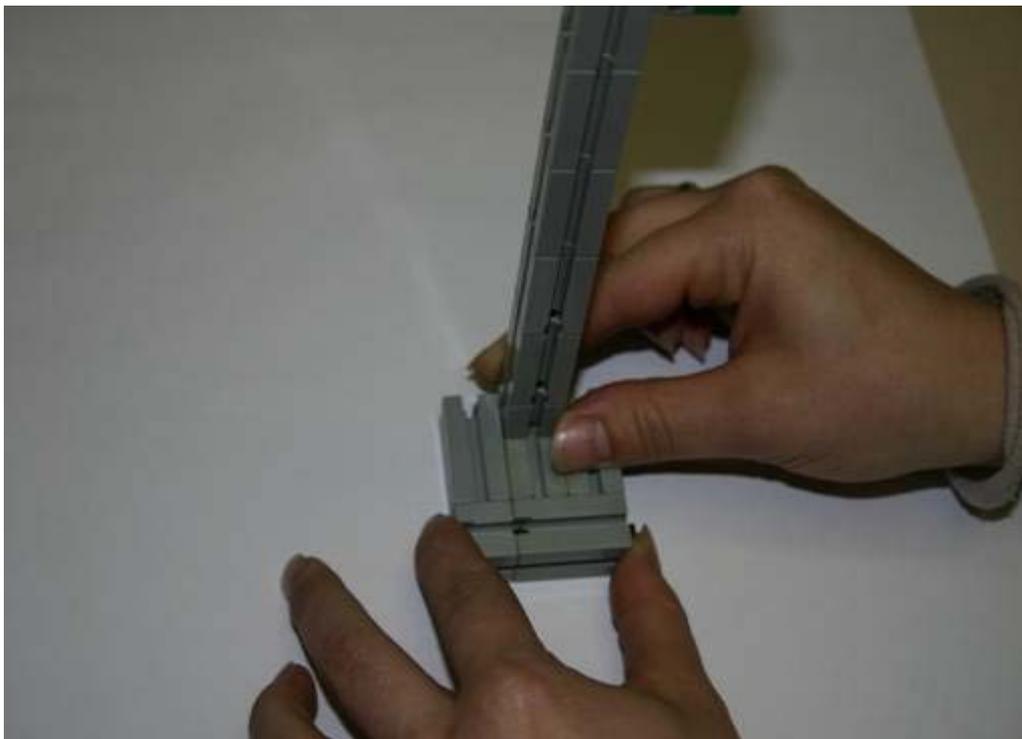
Una vez que ya se ha explicado cuales son los componentes necesarios para la construcción del semáforo, se explicará a continuación paso a paso como se ha realizado el montaje del semáforo, para lo cual se han utilizado piezas de fischer.

1. Se montará el palo del semáforo (2), que estará compuesto por 6 bloques de 3 x 1'5 cm y un último bloque de 1'5 x 1'5 cm.

2. Se montarán las 3 bombillas (3), y se insertarán en la parte superior de palo. Las bombillas se colocarán por este orden: rojo, amarillo y verde, el orden se establece desde la parte superior del palo.

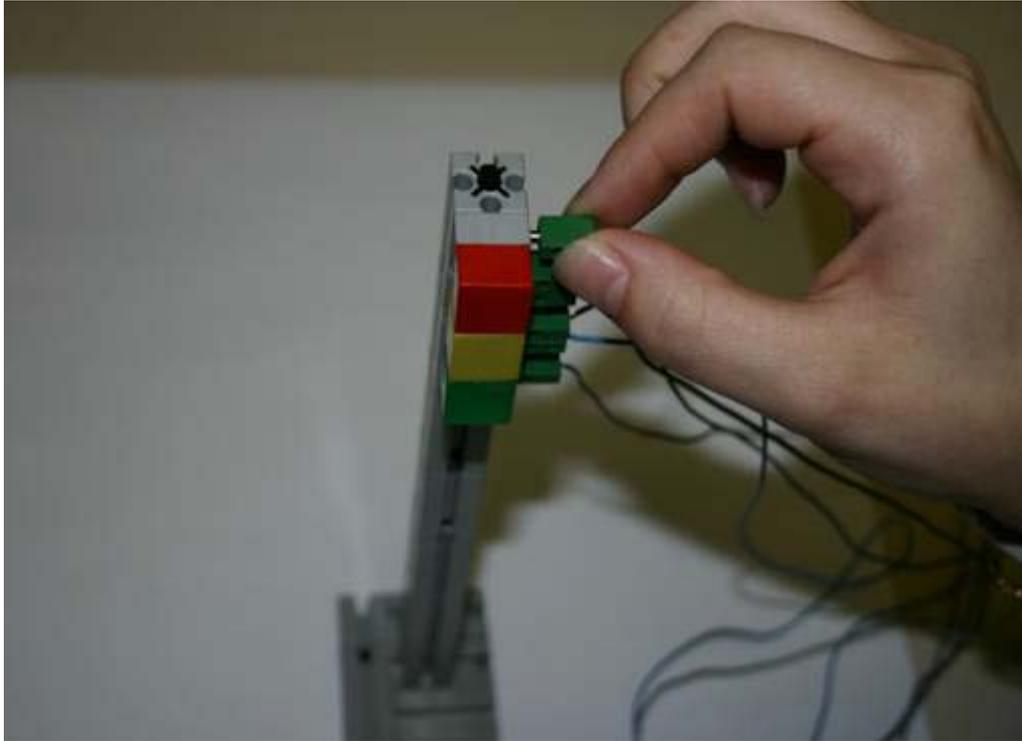


3. Se monta el semáforo sobre un soporte (1), que nos ayude a mantenerlo de pie. El soporte está compuesto por 34 bloques de 3x 1'5 cm y un último bloque de 1'5 x 1'5 cm.



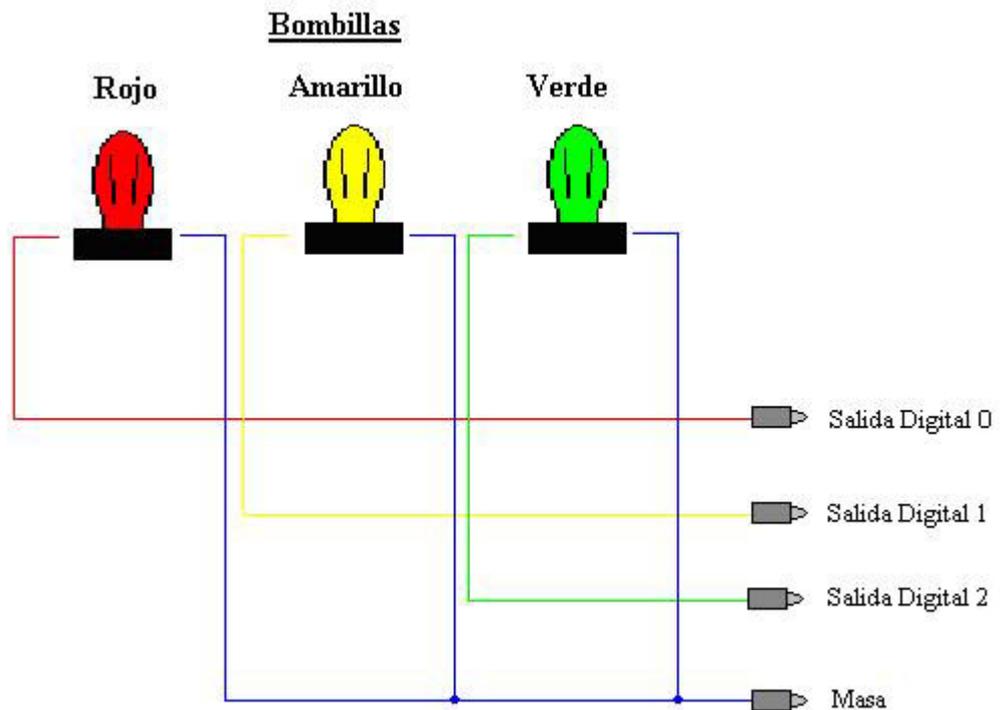
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

4. Por último se electrifica el semáforo, para lo cual se utilizan los cables eléctricos (4) y los conectores (5) de éstos.



ESQUEMA ELÉCTRICO

Para poder alimentar las bombillas es necesario 1 par de hilos eléctricos para cada una de las bombillas, un cable se conectará a las salidas digitales, y el otro cable se conectará a masa. La forma en la que se conectan las bombillas del semáforo a la controladora se muestra en el siguiente esquema:



DESCRIPCIÓN DE LOS SEMÁFOROS

Los conceptos de programación principales que se van a manejar con estos ejemplos son el uso de contadores y la captura de eventos a través del teclado. Con respecto a la controladora se aprenderá a manipular sus salidas digitales.

Se trata de simular el funcionamiento de un semáforo pasando por 3 etapas de menor a mayor dificultad, desarrollando para cada una de ellas un programa.

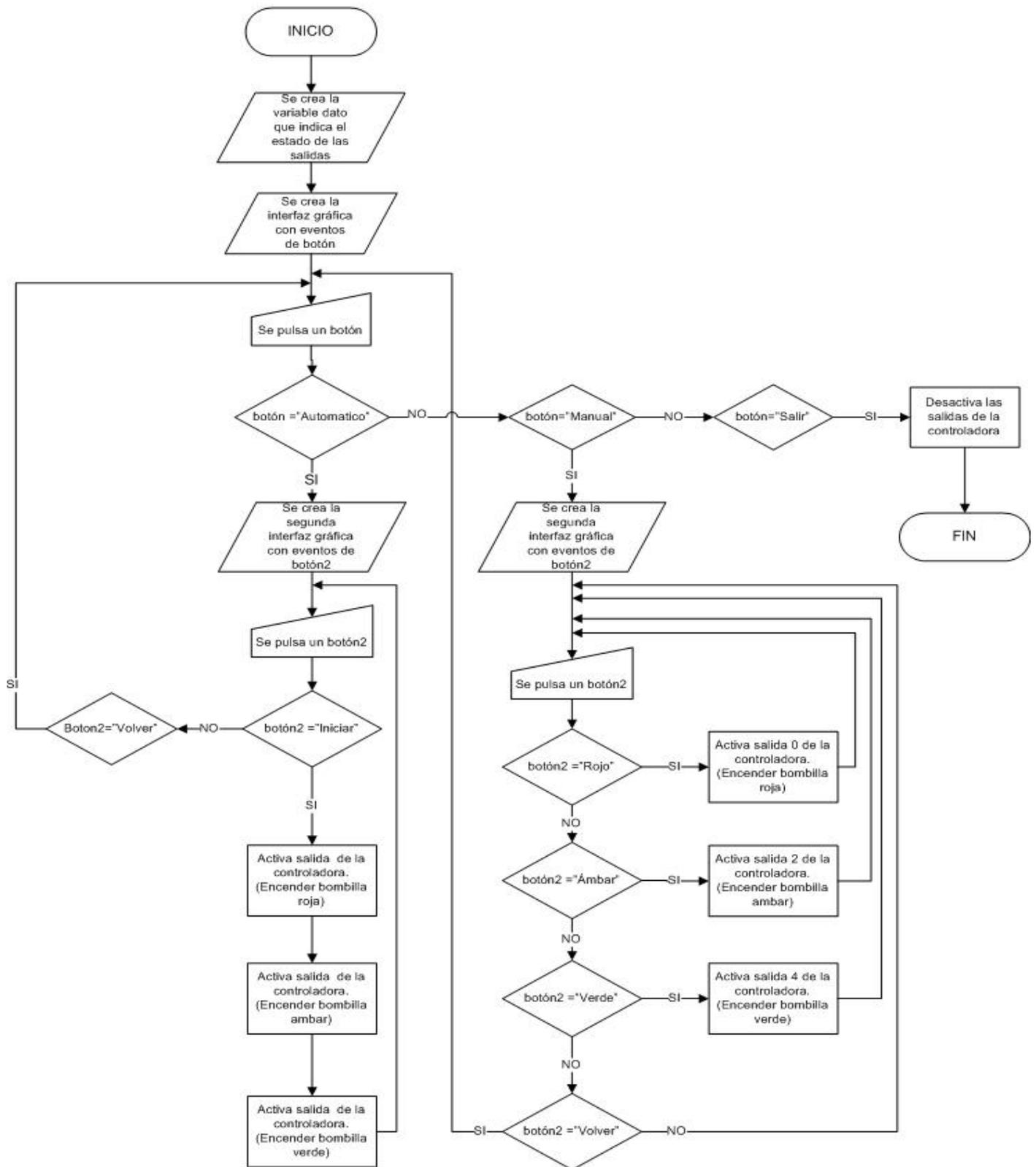
Para comprobar el funcionamiento de cualquiera de los programas de semáforos en la controladora se deberán emplear tres salidas digitales, una para cada color del semáforo. En concreto las salidas serán:

- S0 para el color ROJO.
- S1 para el color ÁMBAR.
- S2 para el color VERDE.

PROGRAMACIÓN

Diagrama de flujo

SEMAFOROS



Programación en MSWLogo

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

1. Se crea un procedimiento para crear la ventana gráfica principal del programa. Para ello se utiliza la función **creaventana**. Dentro de la ventana 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 el botón. Uno de los botones creados será el que se utiliza para salir de la aplicación, para lo cual se utiliza la orden **adios**. Además se mostrará la imagen del semáforo apagado con la función **cargadib**.

```
creaventana " "Principal [Semaforo ] 100 42 150 100 []
creaboton "Principal "Manual "Manual 18 20 50 20 [Modo_Manual]
creaboton "Principal "Automatico "Automático 80 20 50 20 [Modo_Automatico]
creaboton "Principal "Salir "Salir 50 50 50 20 [proc_salir adios]
cargadib ("semáforo.bmp)
```

En este procedimiento se crea y se inicializa a 0 una variable que llamaremos **bucle** que nos servirá para poder leer las entradas digitales de manera continuada.

2. Se crea un procedimiento para crear la ventana gráfica para cada uno de los tipos de funcionamiento del semáforo, Manual o Automático. En el modo manual, aparecerán los botones de los colores del semáforo (ROJO, AMBAR Y VERDE) para que se enciendan manualmente. Entre los corchetes se establecerán las acciones que realizará el semáforo una vez pulsado un botón.

```
borraventana "Principal
creaventana " "Mod_man [Semaforo - Modo Manual] 120 42 200 150 []
creaboton "Mod_man "Rojo "Rojo 75 14 50 20 [Activar 1 cargadib("rojo.bmp)]
creaboton "Mod_man "Ambar "Ambar 75 34 50 20 [Activar 2 cargadib("amarillo.bmp) ]
creaboton "Mod_man "Verde "Verde 75 54 50 20 [Activar 4 cargadib("verde.bmp) ]
creaboton "Mod_man "Apagar "Apagar 45 94 50 20 [Activar 0 cargadib("semaforo.bmp) ]
creaboton "Mod_man "Volver "VOLVER 105 94 50 20 [Activar 0 graficos ]
cargadib("semaforo.bmp) ]
```

Para el modo Automático, se crea una ventana que permite iniciar la secuencia de encendido del semáforo o detenerla.

```
borraventana "Principal
creaventana " "Mod_auto [Semaforo - Modo Automatico] 100 42 150 100 []
creaboton "Mod_auto "Activar "Activar 18 20 50 20 [Luces]
creaboton "Mod_auto "Desactivar "Desactivar 80 20 50 20 [haz "bucle 1 cargadib("semaforo.bmp)]
creaboton "Mod_auto "Volver "VOLVER 50 50 50 20 [Activar 0 graficos]
```

3. El procedimiento **Luces** hace las llamadas correspondientes al procedimiento **Encender** pasándole como parámetros el color de la luz y el tiempo que se quiere tener encendida cada luz. Se va a establecer como tiempo de encendido de cada luz 3 segundos excepto el color ámbar que parpadeara 5 veces. Con estos parámetros determina qué salida digital se debe encender en cada momento:

```
PARA Encender :luz :tiempo
  SISINO (:luz = "ROJO) [SALIDA 1 cargadib("rojo.bmp) ESPERA :tiempo ][
    SISINO (:luz = "AMBAR) [Parpadear :tiempo] [
      SI (:luz = "VERDE) [SALIDA 4 cargadib("verde.bmp) ESPERA :tiempo]
    ] ]
FIN
```

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

4. El procedimiento Parpadear hace que la salida digital S1, que es donde debe conectarse la luz ámbar, se encienda y apague alternativamente. El tiempo que debe permanecer encendida/apagada la luz ámbar depende del valor del parámetro tiempo:

PARA Parpadear : tiempo

DESDE [i 1 5 1] [SALIDA 2 cargadib("amarillo.bmp) ESPERA (: tiempo/6) SALIDA 0 cargadib("semaforo.bmp) ESPERA (: tiempo/6)]

FIN

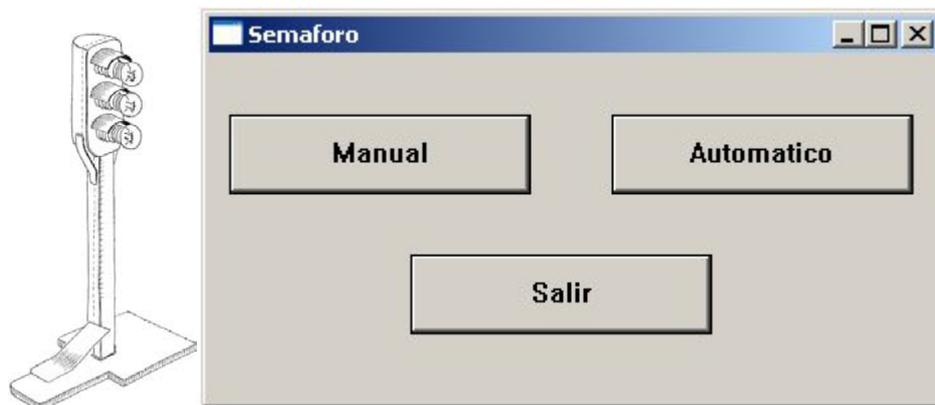
5. Se llama al procedimiento que crea la ventana gráfica fuera de cualquier procedimiento para que se cargue la aplicación gráfica nada más cargar el fichero de logo.

Descargar el archivo programado en **MSWLogo**, descomprímalo y guárdelo en un directorio aparte. Contiene el fichero de código en MSWLogo (semaforo.lgo y las imágenes del semáforo).

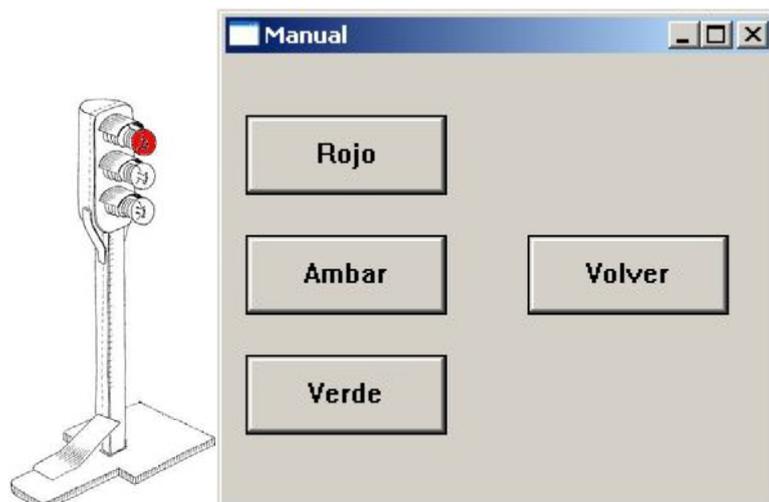
Ejecute el compilador **MSWLogo versión 6.5a** en castellano.

Vaya al menú del programa, Archivo/Abrir y seleccione el fichero **semaforo.lgo** que se descargó previamente.

Se visualizará la siguiente pantalla:



Si se eligió la opción Manual se mostrará la siguiente pantalla:



Si la opción elegida fue la Automática, la pantalla que se mostrará será la siguiente:



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 en 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 dos botones en nuestra ventana, uno para manejar el semáforo manualmente y otro para activar el semáforo de manera automática.

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

7.- Para el botón Manual se crea otra ventana donde aparecen cuatro botones, uno por cada color Rojo, Ámbar y Verde, y otro para volver a la ventana principal. 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 Rojo
        rojo = SDL_LoadBMP("rojo.bmp");
        screen = SDL_SetVideoMode( 35, 220, 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(rojo, NULL,screen,&rect);
        SDL_Flip(screen);
        Escribir_Salidas_Digitales(1);
        break;

    case 2: // botón Ambar
        for(int i=0; i<5 ;i++){
            blanco = SDL_LoadBMP("blanco.bmp");
            screen = SDL_SetVideoMode( 35, 220, 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(blanco, NULL,screen,&rect);
            SDL_Flip(screen);
            Escribir_Salidas_Digitales(0);
            Sleep(500);
            ambar= SDL_LoadBMP("ambar.bmp");
            screen = SDL_SetVideoMode( 35, 220, 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(ambar, NULL,screen,&rect);
            SDL_Flip(screen);
            Escribir_Salidas_Digitales(2);
            Sleep(500);
        }
        break;

    case 3: // botón Verde
        verde= SDL_LoadBMP("verde.bmp");
        screen = SDL_SetVideoMode( 35, 220, 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(verde, NULL, screen, &rect);
    SDL_Flip(screen);
    Escribir_Salidas_Digitales(4);
    break;
case 4: // botón Volver
    Escribir_Salidas_Digitales(0);
    SendMessage(hwnd, WM_CLOSE, 0, 0);
    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;
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 queremos
- 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.

8.- Para el botón Automático se crea otra ventana donde aparecen tres botones, uno para activar el mecanismo del semáforo, otro para desactivarlo, y otro para volver a la ventana principal. En cada uno de ellos se incluye la llamada a la función Escribir_Salidas_Digitales para cambiar el estado de las salidas. Para esta opción se va a crear un hilo que active el semáforo y que compruebe si se ha pulsado el botón de desactivar o no:

```

switch(LOWORD(wParam)) {
case 1: // botón Activar
    dato=1;
    Hilo_Control=CreateThread(NULL, 0, Activar, 0,0, NULL);
    if (Hilo_Control==NULL) {
        MessageBox(0,"Error al crear el hilo", "Welcome Message", 1);
        exit(0);
    }
}

```

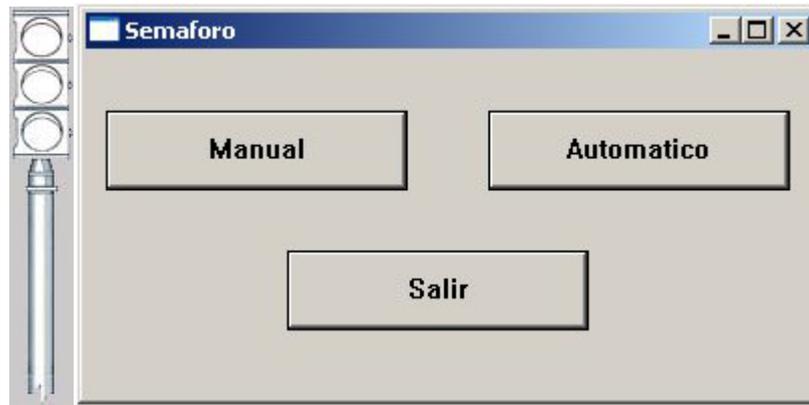
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
    }  
    break;  
case 2: // botón Desactivar  
    dato=0;  
    Escribir_Salidas_Digitales(0);  
    break;  
case 3: // botón Volver  
    Escribir_Salidas_Digitales(0);  
    SendMessage(hwnd, WM_CLOSE, 0, 0);  
    break;  
default:  
    break;  
}
```

9.- Se crea una función que ejecuta el hilo que hemos creado en la opción Automática. La función es la siguiente:

```
DWORD WINAPI Activar(LPVOID){  
    SDL_Surface *blanco, *rojo, *ambar, *verde;  
    SDL_Surface *screen;  
    SDL_Rect rect;  
    for(;;){  
        if(dato==1){  
            rojo = SDL_LoadBMP("rojo.bmp");  
            screen = SDL_SetVideoMode( 35, 220, 0, SDL_NOFRAME );  
  
            if( screen == NULL ) {  
                printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );  
                SDL_Quit();  
            }  
            rect = (SDL_Rect) {0, 0, 200, 200};  
            SDL_BlitSurface(rojo, NULL,screen,&rect);  
            SDL_Flip(screen);  
            Escribir_Salidas_Digitales(1);  
            Sleep(1800);  
            for(int i=0; i<5;i++){  
                blanco = SDL_LoadBMP("blanco.bmp");  
                screen = SDL_SetVideoMode( 35, 220, 0, SDL_NOFRAME );  
                if( screen == NULL ) {  
                    printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );  
                    SDL_Quit();  
                }  
                rect = (SDL_Rect) {0, 0, 200, 200};  
                SDL_BlitSurface(blanco, NULL,screen,&rect);  
                SDL_Flip(screen);  
                Escribir_Salidas_Digitales(0);  
                Sleep(500);  
  
                ambar = SDL_LoadBMP("ambar.bmp");  
                screen = SDL_SetVideoMode( 35, 220, 0, SDL_NOFRAME );  
                if( screen == NULL ) {  
                    printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );  
                    SDL_Quit();  
                }  
            }  
        }  
    }  
}
```


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



Pulsando los botones que se le presentan podrá elegir entre un funcionamiento manual, uno automático o salir de la aplicación.

Si se ha elegido un funcionamiento manual, aparecerá la pantalla siguiente, donde cada botón encenderá una bombilla del semáforo:



Si se ha elegido un funcionamiento automático, aparecerá la siguiente pantalla, donde cada botón activará o desactivará el mecanismo del semáforo:



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.