

TABLA DE CONTENIDO

ANEXO A. MANUAL DE USUARIO.

	Pag
1. Entorno del simulador.	1
1.1 Menú Principal.	1
1.1.1 Configuración general.	2
1.1.1.1. Configuración de la Topología.	3
1.1.1.2. Configuración de los enlaces.	8
1.1.1.3. Configuración de las longitudes de onda.	11
1.1.1.4. Configuración de los puntos de origen y destino.	15
1.1.2. Interfaz de simulación.	17
1.1.2.1. Interfaz grafica de simulación.	19
1.1.2.2. Interfaz de tabla de recorridos o búsquedas.	19
1.1.2.3. Interfaz de resultados.	20
1.1.2.4. Botones de la interfaz de simulación.	20
1.1.3. Botón de ayuda.	22
1.1.4. Botón para salir.	22

ANEXO B. CODIGO FUENTE.

1. Función Inicio.	23
2. Función AS.	29
3. Función Configuración.	32
4. Función Interfaces.	35
5. Función Resultados.	38

LISTA DE FIGURAS

	Pag
Figura 1. Menú principal.	1
Figura 2. Opción para entrar al menú de configuración.	2
Figura 3. Menú de configuración.	2
Figura 4. Escoger la opción para la configuración de la topología,	4
Figura 5. Interfaz para configurar gráficamente la topología de la red.	5
Figura 6. Topologías de red. Ejemplo 1.	5
Figura 7. Topologías de red. Ejemplo 2.	5
Figura 8. Topologías de red. Ejemplo 3.	6
Figura 9. Topologías de red. Ejemplo 4.	6
Figura 10. Topologías de red. Ejemplo 5.	7
Figura 11. Topologías de red. Ejemplo 6.	7
Figura 12. Elegir opción para la configuración de los enlaces.	8
Figura 13. Interfaz de configuración de rutas o enlaces entre routers.	9
Figura 14. Topología de red correspondiente al ejemplo.	10
Figura 15. Interfaz de configuración de rutas o enlaces entre routers.	10
Figura 16. Topología de red correspondiente al ejemplo.	11
Figura 17. Elegir opción para configurar las longitudes de onda.	11
Figura 18. Interfaz de configuración de longitudes de onda.	12
Figura 19. Representación grafica de asignación de longitudes de onda igual a 15, 8 y 2 o (1111 , 1000 , 0010) .	13
Figura 20. Interfaz de configuración de longitudes de onda.	14
Figura. 21. Longitudes de onda habilitadas.	14
Figura 22. Elegir puntos de origen y destino para simular entre ellos.	15
Figura 23. Escoger puntos de origen y destino.	15
Figura 24. Elección de los puntos de origen y destino sobre los cuales se quiere aplicar la simulación.	16
Figura 25. Para volver al menú principal.	16
Figura 26. Elegir interfaz de simulación.	17
Figura 27. Interfaz de simulación.	18

Figura 28.	Interfaz grafica de la red.	19
Figura 29.	Interfaz de recorridos o búsquedas.	19
Figura 30.	Interfaz de resultados.	20
Figura 31.	Botones de la interfaz de simulación.	20
Figura 32.	Escoger puntos de origen y destino desde la interfaz de simulación.	22

LISTA DE TABLAS

	Pag
Tabla 5.1. Tabla de representación numérica de las longitudes de onda.	13

ANEXO A. MANUAL DE USUARIO.

1. Entorno del Simulador.

1.1 Menú Principal:



Figura 1. Menú principal.

Esta es la ventana de bienvenida que permite escoger haciendo clic sobre las opciones correspondientes todo el proceso de simulación como se explicara detalladamente a continuación.

1.1.1 Configuración general:



Figura 2. Opción para entrar al menú de configuración.

Una vez se halla hecho clic sobre la opción de configuración esta se iluminara como indica la figura y se desplegara la siguiente ventana.



Figura 3. Menú de configuración.

Dentro de este menú se encuentran todas las opciones de configuración necesarias para llevar a cabo la simulación, las cuales son: configurar la topología de la red, las rutas o enlaces, las longitudes de onda y los puntos de origen / destino como se explica a continuación.

1.1.1.1. Configuración de la Topología.



Figura 4. escoger la opción para la configuración de la topología,

Para entrar la modo de configuración de la topología se debe hacer clic sobre el botón topología, se desplegara la grafica siguiente.

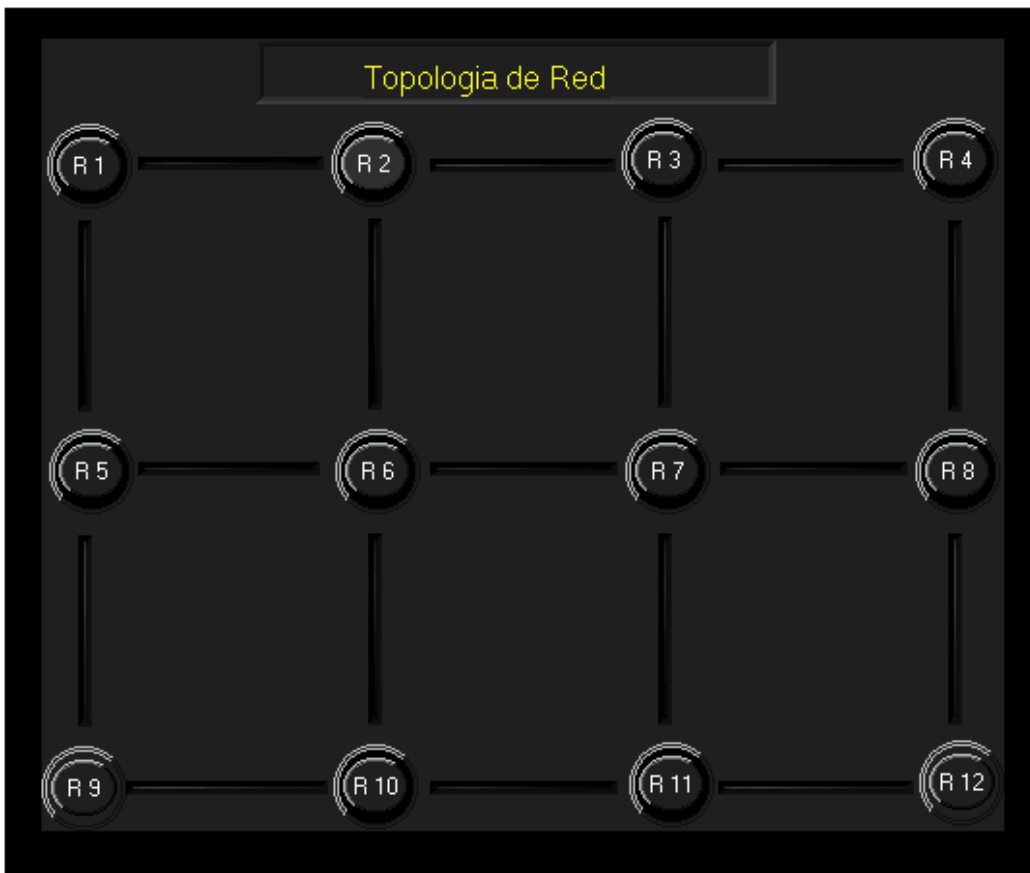


Figura 5. Interfaz para configurar gráficamente la topología de la red.

Estando en este punto se podrá habilitar los routers que se deseen, se puede interconectar mínimo dos y un máximo de doce en forma de matriz de 4 x 3, para esto simplemente se hace clic sobre el router que se desee habilitar, este se iluminara indicando su estado de router activo, a continuación se muestran algunos ejemplos de las topología que se pueden lograr.

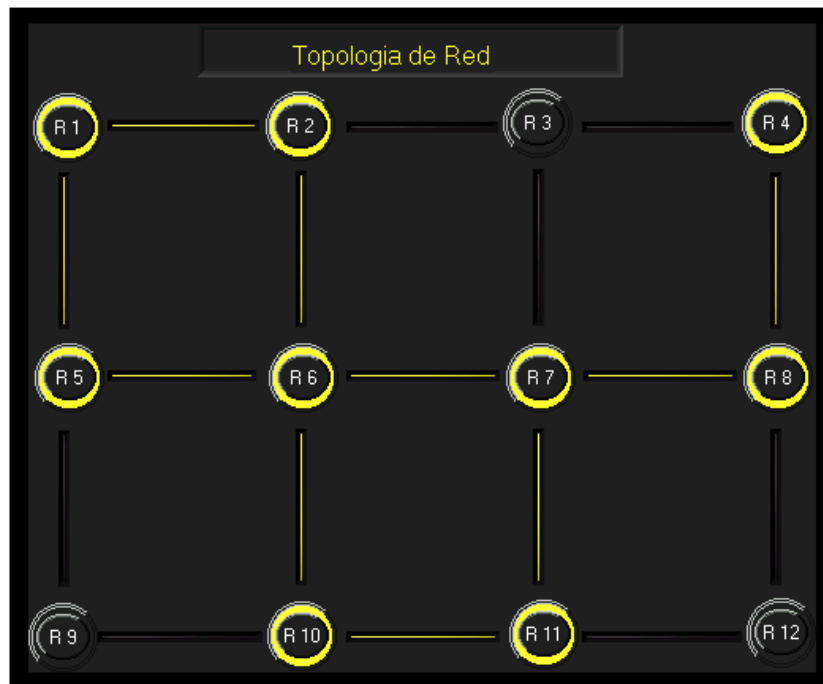


Figura 6. Topologías de red , ejemplo 1.

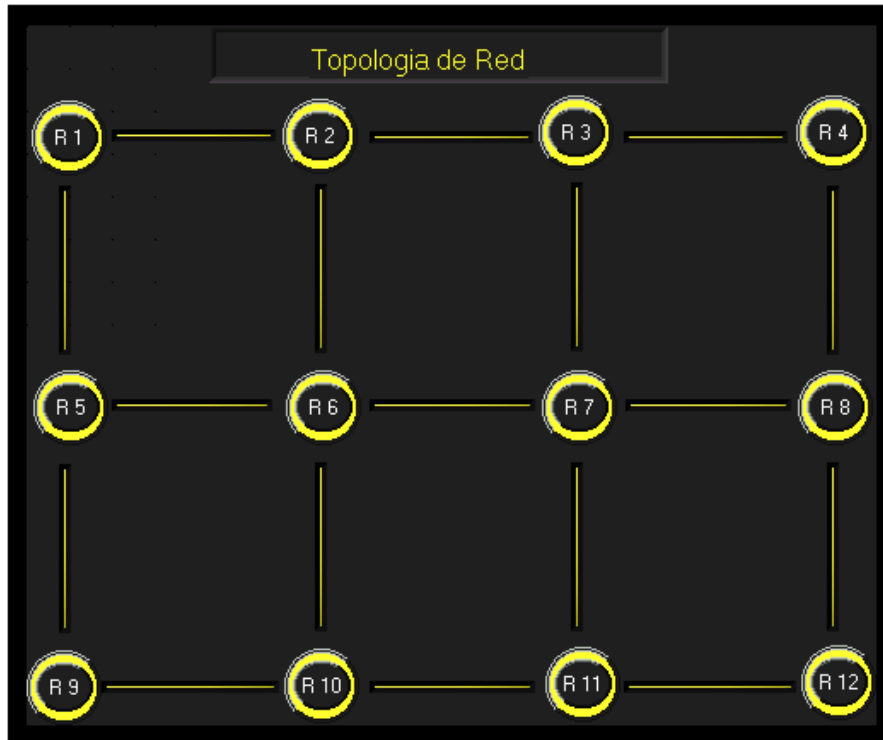


Figura 7. Topologías de red, ejemplo 2.

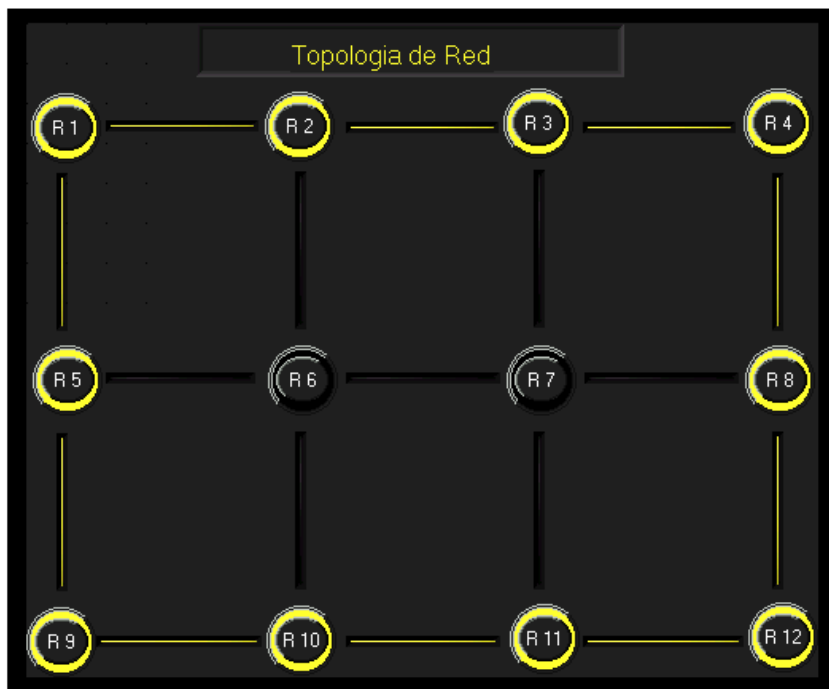


Figura 8. Topologías de red, ejemplo 3.

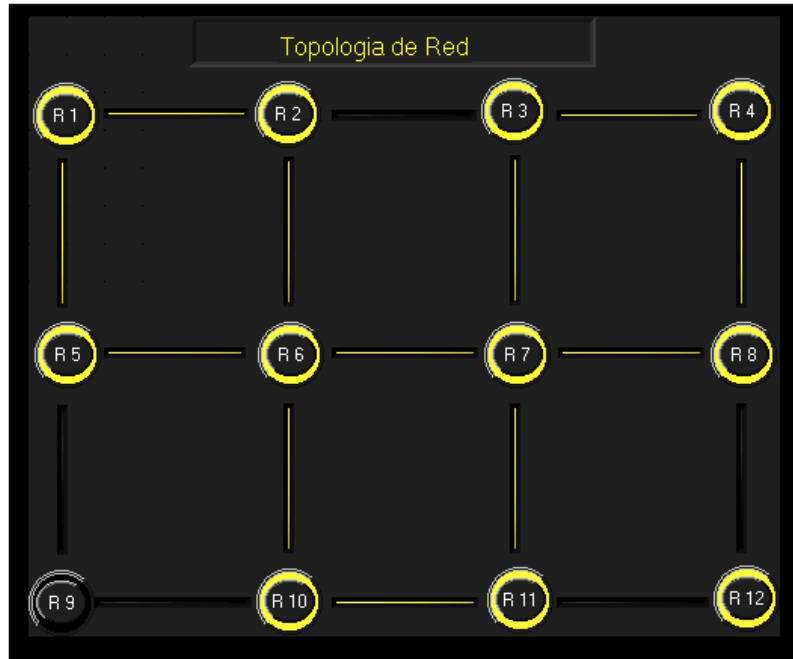


Figura 9. Topologías de red, ejemplo 4.

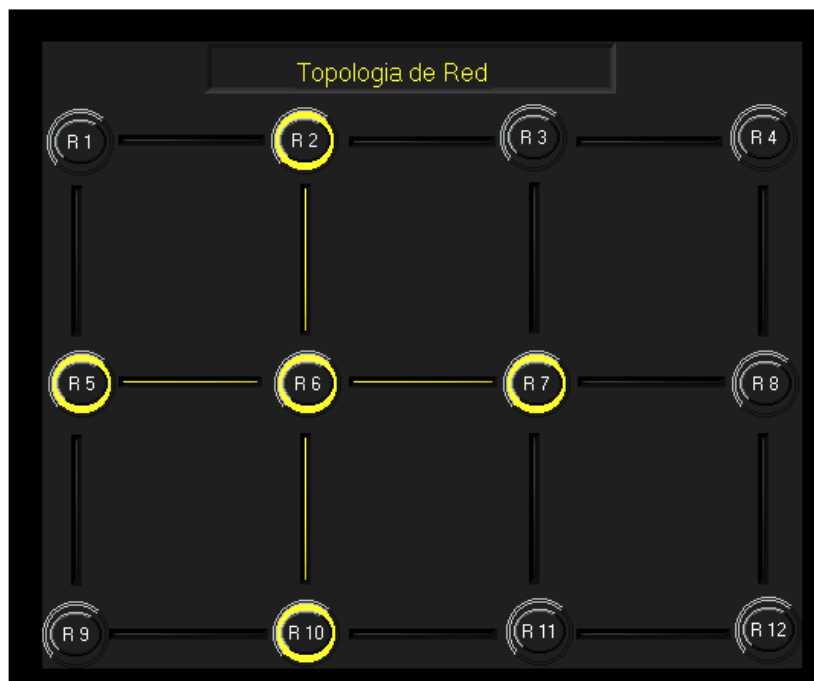


Figura 10. Topología de red, ejemplo 5.

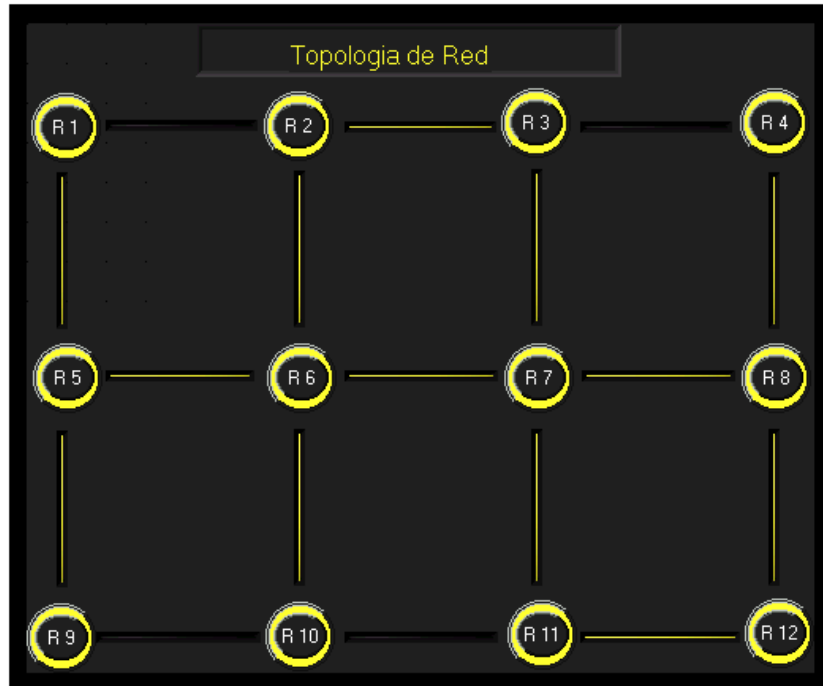


Figura 11. Topologías de red, ejemplo 6.

1.1.1.2. Configuración de los enlaces:

Para entrar al modo de configuración de los enlaces se debe primero hacer clic sobre el botón de enlaces en el menú principal.



Figura 12. Elegir opción para la configuración de los enlaces.

Aparecerá la siguiente tabla.

Tabla de Configuración de Rutas												
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
R1	0	1	0	0	1	0	0	0	0	0	0	0
R2	1	0	1	0	0	1	0	0	0	0	0	0
R3	0	1	0	1	0	0	1	0	0	0	0	0
R4	0	0	1	0	0	0	0	1	0	0	0	0
R5	1	0	0	0	0	1	0	0	1	0	0	0
R6	0	1	0	0	1	0	1	0	0	1	0	0
R7	0	0	1	0	0	1	0	1	0	0	1	0
R8	0	0	0	1	0	0	1	0	0	0	0	1
R9	0	0	0	0	1	0	0	0	0	1	0	0
R10	0	0	0	0	0	1	0	0	1	0	1	0
R11	0	0	0	0	0	0	1	0	0	1	0	1
R12	0	0	0	0	0	0	0	1	0	0	1	0

V O L V E R

Figura 13. Interfaz de configuración de rutas o enlaces entre routers.

Los espacios blancos representan los cruces posibles entre routers, los espacios grises representan la imposibilidad de cruce entre routers, solo se puede escribir en los espacios blancos donde un uno significa conexión y un cero significa que no hay conexión en ese enlace, para este ejemplo en particular donde todos los cruces posibles(en blanco) están en uno la topología correspondiente sería la siguiente.

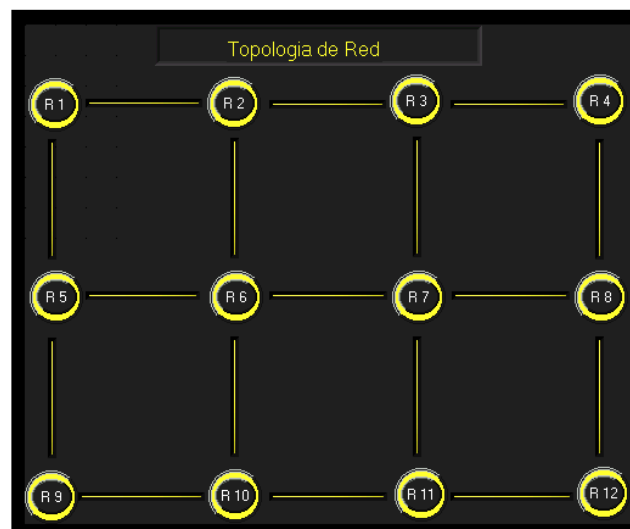


Figura 14. Topologías de red correspondiente al ejemplo.

Y para la siguiente configuración :

Tabla de Configuración de Rutas												
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
R1	0	0	0	0	1	0	0	0	0	0	0	0
R2	0	0	0	0	0	0	0	0	0	0	0	0
R3	0	0	0	1	0	0	0	0	0	0	0	0
R4	0	0	1	0	0	0	0	1	0	0	0	0
R5	1	0	0	0	0	1	0	0	0	0	0	0
R6	0	0	0	0	1	0	1	0	0	0	0	0
R7	0	0	0	0	0	1	0	1	0	0	1	0
R8	0	0	0	1	0	0	1	0	0	0	0	1
R9	0	0	0	0	0	0	0	0	0	0	0	0
R10	0	0	0	0	0	0	0	0	0	0	1	0
R11	0	0	0	0	0	0	1	0	0	1	0	0
R12	0	0	0	0	0	0	0	1	0	0	0	0

VOLVER

Figura 15. Interfaz de configuración de rutas o enlaces entre routers.

La topología correspondiente sera:

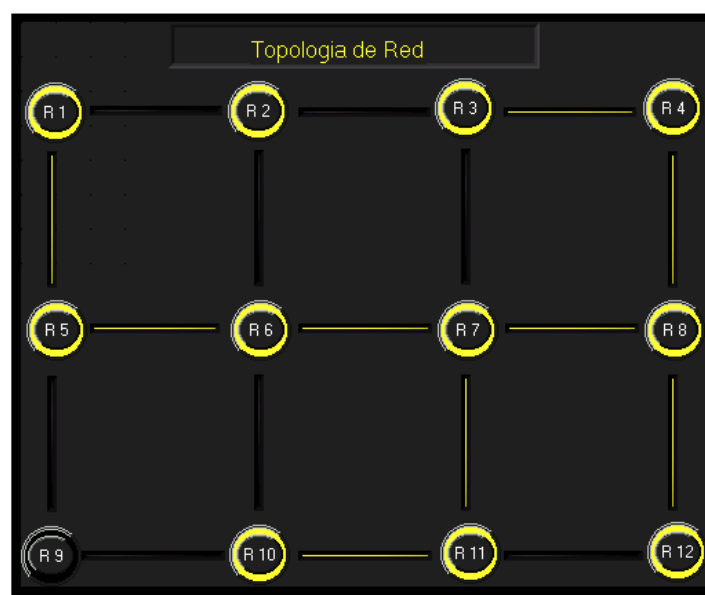


Figura 16. Topología de red correspondiente al ejemplo anterior.

1.1.1.3. Configuración de las longitudes de onda:

Para entrar al modo de configuración de las longitudes de onda de cada enlace, se debe primero hacer clic sobre la opción longitudes de onda en el menú principal.



Figura 17. Elegir opción para configurar las longitudes de onda.

Se desplegara la siguiente interfaz:

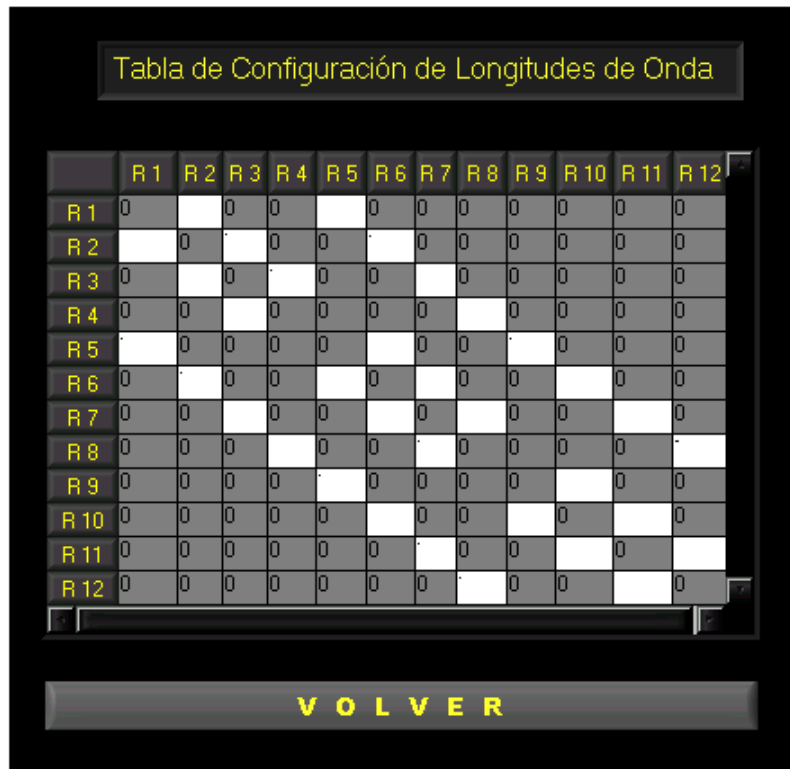


Figura 18. Interfaz de configuración de longitudes de onda.

Igual que en la interfaz de configuración de enlaces en esta tabla los espacios blancos donde se puede escribir representa los cruces de los rorters, el usuario podrá escoger entre cero y cuatro longitudes de onda por enlace , para mayor facilidad la siguiente tabla muestra el numero que corresponde a las diferentes alternativas posibles de para habilitar las longitudes de onda.

Numero Representativo	Numero binario	Longitudes de onda habilitadas
0	0000	Ninguna
1	0001	L1
2	0010	L2
3	0011	L2-L1
4	0100	L3

5	0101	L3-L1
6	0110	L3-L2
7	0111	L3-L2-L1
8	1000	L4
9	1001	L4-L1
10	1010	L4-L2
11	1011	L4-L2-L1
12	1100	L4-L3
13	1101	L4-L3-L1
14	1110	L4-L3-L2
15	1111	L4-L3-L2-L1

Tabla 5.1. Tabla de representación numérica de las longitudes de onda.

Por ejemplo la representación grafica para 15 , 8 y 2 , donde 15=1111 todas las longitudes de onda activas, 8=1000 solo activa la primera y 2= 0010 solo activa la tercera seria:



Figura 19. Representación grafica de asignación de longitudes de onda igual a 15 , 8 y 2 o (1111 , 1000 , 0010) .

y la interfaz de configuración será de la siguiente manera:

Tabla de Configuración de Longitudes de Onda

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
R1	0	15	0	0	0	0	0	0	0	0	0	0
R2	15	0	8	0	0	0	0	0	0	0	0	0
R3	0	8	0	2	0	0	0	0	0	0	0	0
R4	0	0	2	0	0	0	0	0	0	0	0	0
R5	0	0	0	0	0	0	0	0	0	0	0	0
R6	0	0	0	0	0	0	0	0	0	0	0	0
R7	0	0	0	0	0	0	0	0	0	0	0	0
R8	0	0	0	0	0	0	0	0	0	0	0	0
R9	0	0	0	0	0	0	0	0	0	0	0	0
R10	0	0	0	0	0	0	0	0	0	0	0	0
R11	0	0	0	0	0	0	0	0	0	0	0	0
R12	0	0	0	0	0	0	0	0	0	0	0	0

V O L V E R

Figura 20. Interfaz de configuración de longitudes de onda.

Nos mostraría el siguiente resultado en la interfaz de simulación:

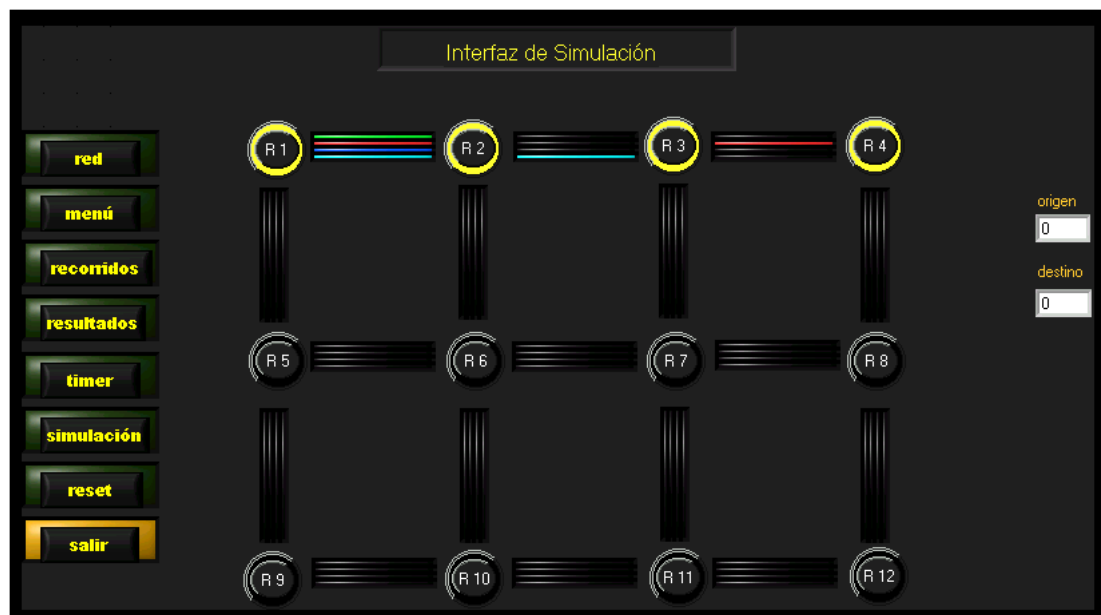


Figura. 21 Longitudes de onda habilitadas.

1.1.1.4. Configuración de los puntos de origen y destino:

Para entrar al modo de configuración de los puntos de origen y destino se debe primero escoger la opción correspondiente:



Figura 22. Elegir puntos de origen y destino para simular entre ellos.

Después simplemente se debe escribir el numero del router correspondiente según la topología en cada espacio como se indica a continuación:



Figura 23. Escoger puntos de origen y destino.



Figura 24. Elección de los puntos de origen y destino sobre los cuales se quiere aplicar la simulación.

Esto también se podría hacer directamente desde los espacios similares que se encuentran sobre la interfaz de simulación como se explicara mas adelante.

El ultimo botón dentro del menú de configuración como su nombre lo indica sirve para volver al menú principal.



Figura 25 . Para volver al menú principal.

1.1.2. Interfaz de simulación.

Estando sobre el menú principal, y habiendo configurado ya la red en la cual se desea simular el algoritmo, se puede ir a la interfaz de simulación mediante el botón simulación.



Figura 26. Elegir interfaz de simulación.

Se desplegara la siguiente interfaz que se describirá a continuación.

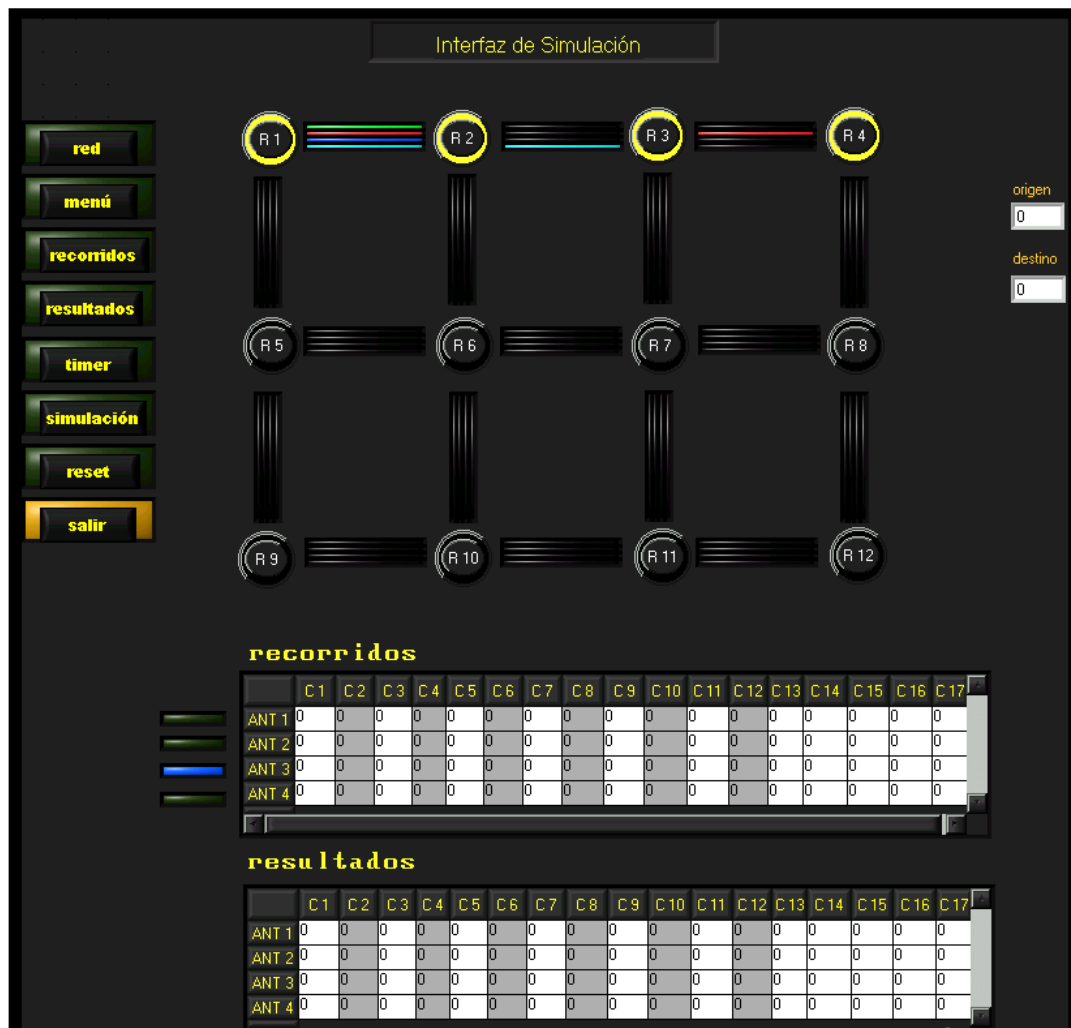


Figura 27. Interfaz de simulación.

La primera parte de la interfaz de simulación que se describirá es la interfaz grafica de simulación.

1.1.2.1. Interfaz grafica de simulación:

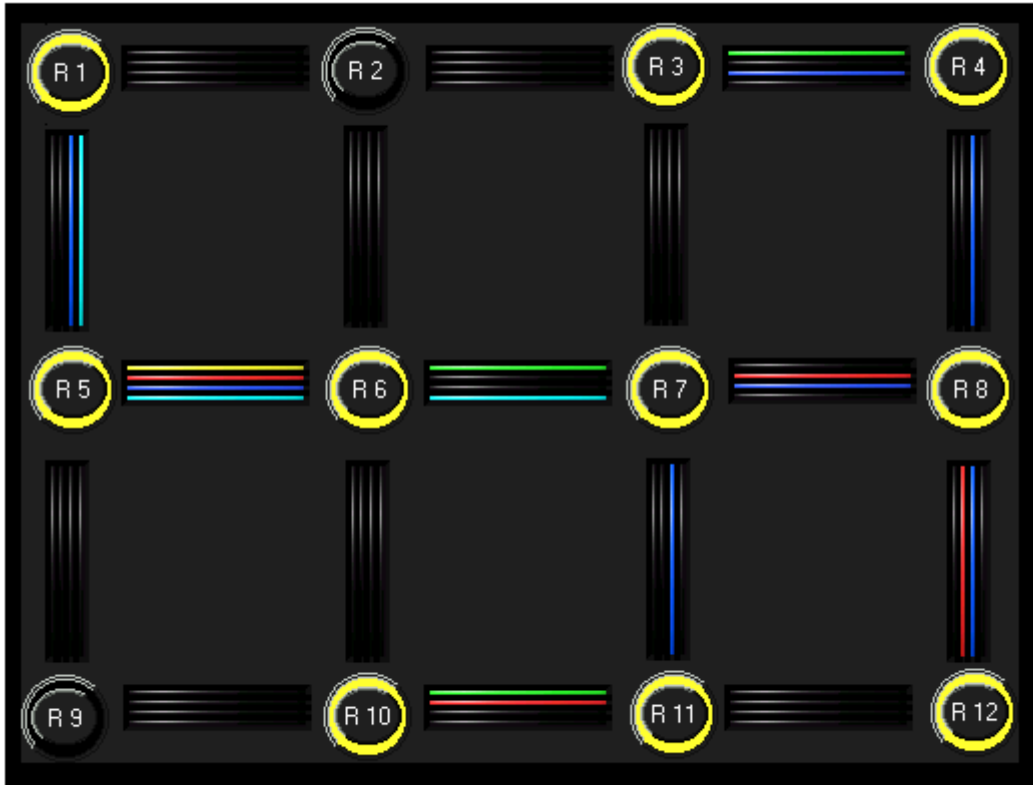


Figura 28 . Interfaz grafica de la red.

En esta interfaz se puede ver una representación grafica de la topología de la red , donde los routers habilitados están iluminados al igual que se iluminan las longitudes de onda correspondiente a los recorridos de búsqueda del algoritmo durante su proceso de ejecución.

1.1.2.2. Interfaz de tabla de recorridos o búsquedas:

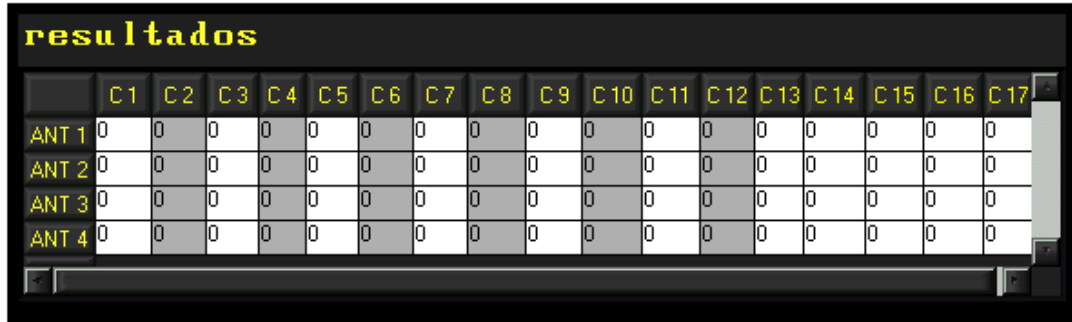
recorridos

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
ANT 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 29. Interfaz de recorridos o búsquedas.

Esta interfaz muestra en forma numérica los recorridos de búsqueda del algoritmo, se iluminara al lado izquierdo el led del color correspondiente a la longitud de onda que se este actualizando ese momento.

1.1.2.3. Interfaz de resultados:



	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
ANT 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANT 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 30. Interfaz de resultados.

En esta interfaz se podrá observar los resultados obtenidos al finalizar el proceso de cada simulación marcando el recorrido completo que se halla considerado como una mejor solución.

1.1.2.4. Botones de la interfaz de simulación:



Figura 31. Botones de la interfaz de simulación.

La función de cada botón de esta interfaz se explica a continuación:

Red: Nos permite observar como se ha configurado la topología de la red en su totalidad.

Menú: Vuelve al menú principal

Recorridos: Permite observar la interfaz de recorridos en forma individual y en una ventana mas grande.

Resultados: Permite observar la interfaz de resultados en forma individual y en una ventana mas grande.

Timer: Permite modificar el tiempo de ejecución del proceso de simulación para poder observar los cambios mas lentamente.

Simulación : Permite empezar el proceso de simulación entre los dos puntos escogidos.

Reset: Resetea toda la interfaz de simulación permitiendo iniciar nuevos procesos, sin borrar la configuración inicial.

Salir: sale del sistema

Para finalizar como se había dicho en la parte de configuración de los puntos de origen y destino el sistema permite configurar estos desde la propia interfaz grafica de simulación lo que facilita la repetición de procesos entre diferentes puntos de referencia, se hace escribiendo sobre los espacios el valor correspondiente a los routers deseados.



Figura 32. Escoger puntos de origen y destino desde la interfaz de simulación.

1.1.3. Botón de ayuda:

Presionando esta opción se tiene acceso a este documento de manual de usuario directamente .

1.1.4. Botón para salir:

Presionando esta opción se sale del simulador.

ANEXO B. CODIGO FUENTE.

En el presente anexo se presentan el código fuente de las principales funciones que conforman este proyecto, las funciones restantes se encuentran en el CD ROM que se entrega con la monografía.

1. Función Inicio.

```
#include <ansi_c.h>
#include <utility.h>
#include <cvirte.h>
#include <userint.h>
#include "DEFI.h"

static int panel1, panel2, panel3, panel4, panel5, panel6;

int a;L;L1;L2;L3;i;VAR1;
int CHECKBOX_1,CHECKBOX_2,CHECKBOX_3,CHECKBOX_4;
double timery;

double
ppp[25]={0xFF0000L,0x00FF00L,0x0000FFL,0x00FFFFL,0xFF00FFL,0xFFFF0
0L,0x800000L,0x000080L,
0x008000L,0x008080L,0x800080L,0x808000L,0xC0C0C0L,0x808080L,0x0000
00L,0xFFFFFFL,
0xC0C0C0L,0xA0A0A0L };

int ruta[20]={1,2,3,4,8,7,6,5,9,10,11,12 };

int way;rut;longi;

void ways (void);
void longs (void);
void waysplus(void);
void tablas(void);

int main (int argc, char *argv[])
```

```

{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;
    if ((panel1 = LoadPanel (0, "DEFI.uir", PANEL)) < 0)// Panel de interfaz
de simulación.
        return -1;
    if ((panel2 = LoadPanel (0, "DEFI.uir", PANEL_2)) < 0)// Panel de interfaz
de configuración de rutassimulación.
        return -1;
    if ((panel3 = LoadPanel (0, "DEFI.uir", PANEL_3)) < 0)// Panel de interfaz
de configuración. de longitudes de onda.
        return -1;
    if ((panel4 = LoadPanel (0, "DEFI.uir", PANEL_4)) < 0)// Panel de tablas
de resultados y de recorridos.
        return -1;
    if ((panel5 = LoadPanel (0, "DEFI.uir", PANEL_5)) < 0)// Panel de interfaz
de menu pricipal.
        return -1;
    if ((panel6 = LoadPanel (0, "DEFI.uir", PANEL_6)) < 0)// Panel de interfaz
de simulación grafica.
        return -1;
    DisplayPanel (panel1);
    RunUserInterface ();
    DiscardPanel (panel1);
    return 0;
}

```

```

int CVICALLBACK CB_R1 (int panel, int control, int event, // Evento del router
1

```

```

    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (panel1, PANEL_LED_7,&a);
            if (a==0)
            {
                SetCtrlVal (panel1, PANEL_LED_R1,1);
                Delay(.2);
                DisplayPanel (panel2);
            }
            if (a==1)
            {
                SetCtrlVal (panel1, PANEL_LED_R1,0);
            }

            break;
        }
    return 0;
}

```

```
}
```

```
int CVICALLBACK CB_2 (int panel, int control, int event, // Vuelve del panel de configuracion de los routers.
```

```
void *callbackData, int eventData1, int eventData2)
```

```
{
```

```
    switch (event)
```

```
    {
```

```
        case EVENT_COMMIT:
```

```
            DisplayPanel(panel1);
```

```
            HidePanel(panel2);
```

```
            break;
```

```
    }
```

```
    return 0;
```

```
}
```

```
int CVICALLBACK CB_3 (int panel, int control, int event, // Resetea los caminos de leds
```

```
void *callbackData, int eventData1, int eventData2)
```

```
{
```

```
    int i=0;
```

```
    int j=0;
```

```
    switch (event)
```

```
    {
```

```
        case EVENT_COMMIT:
```

```
            SetCtrlVal (panel1, PANEL_NUMERIC_1,0);
```

```
            SetCtrlVal (panel1, PANEL_NUMERIC_2,0);
```

```
            while (j<=4)
```

```
            {
```

```
                for (i=1;j<=17;i++)
```

```
                {
```

```
                    SetTableCellVal (panel1, PANEL_TABLE, MakePoint  
(i,j), 0);
```

```
                }
```

```
                j=j+1;
```

```
            }
```

```
            if(L==1)
```

```
                L=0;
```

```
            else
```

```
                L=1;
```

```
                ways();
```

```
            break;
```

```
    }
```

```
    return 0;
```

```
}
```

```

int CVICALLBACK CB_4 (int panel, int control, int event, // Asignación de
colores para las longitudes de onda
void *callbackData, int eventData1, int eventData2)
{
    int j=0;
    switch (event)
    {
        case EVENT_COMMIT:
            for(j=1;j<=3;j++)
            {
                for(i=1;i<=18;i++)
                {
                    L=ppp[i];
                    L1=ppp[i+1];
                    L2=ppp[i+2];
                    L3=ppp[i+3];

                    SetCtrlVal (panel1, PANEL_COLORNUM_1,L);
                    SetCtrlVal (panel1, PANEL_COLORNUM_2,L1);
                    SetCtrlVal (panel1, PANEL_COLORNUM_4,L3);
                    Delay(.1);
                }
            }
            break;
    }
    return 0;
}

```

```

int CVICALLBACK CB_5 (int panel, int control, int event, // Desliega el panel de
conectividad de routers.
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlVal (panel1, PANEL_LED_tabla1,1);
            DisplayPanel(panel5);

            break;
    }
    return 0;
}

```

```

int CVICALLBACK CB_6 (int panel, int control, int event, // Vuelve del panel de
tablas
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

```

```

        SetCtrlVal (panel1, PANEL_LED_tabla1,0);
        DisplayPanel(panel1);
        HidePanel(panel4);
        break;
    }
    return 0;
}

int CVICALLBACK CB_7 (int panel, int control, int event, // Vuelve del panel de
conectividad
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlVal (panel1, PANEL_LED_tabla2,0);
            DisplayPanel(panel1);
            HidePanel(panel3);
            break;
    }
    return 0;
}

int CVICALLBACK CB_8 (int panel, int control, int event, // Depliega el panel de
longitudes de onda
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlVal (panel1, PANEL_LED_tabla2,1);
            DisplayPanel(panel6);

            break;
    }
    return 0;
}

int CVICALLBACK CB_9 (int panel, int control, int event, // Resetea los routers
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            if(L==1)
                L=0;
            else
                L=1;
    }
}

```

```

        timery=0.2;
        SetCtrlVal (panel1, PANEL_LED_resetone,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R1,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R2,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R3,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R4,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R5,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R6,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R7,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R8,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R9,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R10,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R11,L);
        Delay(timery);
        SetCtrlVal (panel1, PANEL_LED_R12,L);

        break;
    }
    return 0;
}

int CVICALLBACK QuitCallback (int panel, int control, int event, // Sale de la
aplicacion
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```

2. Función AS.

```

int CVICALLBACK CB_11 (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)

```

```

{
    switch (event)
    {
        case EVENT_COMMIT:
            if(L==1)
                L=0;
            else
                L=1;
            timery=1;

            SetCtrlVal (panel1, PANEL_LED_18,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint (5,1),
1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_46,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint (8,1),
1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_61,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint
(16,1), 1);

            Delay(timery);

            SetCtrlVal (panel1, PANEL_LED_1,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint (1,1),
1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_16,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint (4,1),
1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_42,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint
(11,1), 1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_57,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint
(15,1), 1);

            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_61,L);
            SetTableCellVal (panel, PANEL_TABLE, MakePoint
(16,1), 2);

            Delay(timery);

            SetCtrlVal (panel1, PANEL_LED_5,L);
            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_9,L);
            Delay(timery);
            SetCtrlVal (panel1, PANEL_LED_26,L);
            Delay(timery);

```



```
SetCtrlVal (panel1, PANEL_LED_54,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_65,L);  
Delay(timery);
```

```
SetCtrlVal (panel1, PANEL_LED_22,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_50,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_26,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_54,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_65,L);  
Delay(timery);
```

```
SetCtrlVal (panel1, PANEL_LED_6,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_10,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_25,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_53,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_66,L);  
Delay(timery);
```

```
SetCtrlVal (panel1, PANEL_LED_3,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_14,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_43,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_59,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_63,L);  
Delay(timery);
```

```
SetCtrlVal (panel1, PANEL_LED_8,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_12,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_28,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_40,L);  
Delay(timery);  
SetCtrlVal (panel1, PANEL_LED_52,L);
```

```

    }
    return 0;
}

```

```

    Delay(timery);
    break;

```

```

int CVICALLBACK CB_10 (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{

```

```

    int orig=0;
    int dest=0;
    int alar=0;

```

```

    switch (event)
    {

```

```

        case EVENT_COMMIT:

```

```

            GetCtrlVal (panel1, PANEL_NUMERIC_1,&orig);
            GetCtrlVal (panel1, PANEL_NUMERIC_2,&dest);

```

```

            for (rut=0;rut<=11;rut++)

```

```

        {

```

```

            if ( ruta[rut]>=orig&&ruta[rut]<dest)
            {

```

```

                GetTableCellVal (panel6, PANEL_6_TABLE, MakePoint
(ruta[rut],ruta[rut+1]), &longi);

```

```

                if (longi==0 )
                {

```

```

                }

```

```

                if (longi!=0 )
                {

```

```

                    longs();
                    L=1;
                    waysplus();
                    tablas();

```

```

        }
    }
}
break;
}
return 0;
}

```

3. Función configuración.

```

int CVICALLBACK CB_13 (int panel, int control, int event,// Vuelve del panel de
conectividad
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            GetCtrlVal (panel2, PANEL_2_STRING_1,&CHECKBOX_1);
            GetCtrlVal (panel2, PANEL_2_STRING_2,&CHECKBOX_2);
            GetCtrlVal (panel2, PANEL_2_CHECKBOX_1,&CHECKBOX_1);
                GetCtrlVal (panel2,
PANEL_2_CHECKBOX_2,&CHECKBOX_2);
                GetCtrlVal (panel2,
PANEL_2_CHECKBOX_3,&CHECKBOX_3);
                GetCtrlVal (panel2,
PANEL_2_CHECKBOX_4,&CHECKBOX_4);

                GetTableCellVal (panel4, PANEL_4_TABLE_2, MakePoint
(CHECKBOX_1,CHECKBOX_2), CHECKBOX_5);

                if (CHECKBOX_5)
                    break;
            }
        return 0;
    }
}

```

```
int CVICALLBACK CB_14 (int panel, int control, int event, // Vuelve del panel 5
conectuvudad entre routers
```

```
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlVal (panel1, PANEL_LED_tabla1,0);
            DisplayPanel(panel1);
            HidePanel(panel5);
            break;
    }
    return 0;
}
```

```
int CVICALLBACK CB_15 (int panel, int control, int event, // Vuelve panel 6
longitudes de onda
```

```
void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlVal (panel1, PANEL_LED_tabla2,0);
            DisplayPanel(panel1);
            HidePanel(panel6);
            break;
    }
    return 0;
}
```

```
void tablas(void)
```

```
{
    int L2=0;
    int L3=1;

    if(L2==1)
        L2=0;
    if(L2==0)
        L2=1;

    if(L3==1)
        L3=0;
    if(L3==0)
        L3=1;

    SetCtrlVal (panel1, PANEL_avigreen,L2);
    Delay(.5);
    SetCtrlVal (panel1, PANEL_avired,L3);
    Delay(.5);
}
```

```
SetCtrlVal (panel1, PANEL_aviblue,L2);
Delay(.5);
SetCtrlVal (panel1, PANEL_avician,L3);
Delay(.5);
```

```
if(L1==1)
    SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,1), +1);
    if(L1==2)
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);
    if(L1==3)
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,3), +1);
    if(L1==4)
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
    if(L1==5)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,1), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,3), +1);}
    if(L1==6)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,3), +1);}
    if(L1==7)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,3), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,1), +1);}
    if(L1==8)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);}
    if(L1==9)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,1), +1);}
    if(L1==10)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);}
    if(L1==11)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
```

```

        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,1), +1);}
        if(L1==12)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);}
        if(L1==13)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,3), +1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,1), +1);}
        if(L1==14)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,3), +1);}
        if(L1==15)
        {SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,4), +1);
        SetTableCellVal (panel1, PANEL_TABLE, MakePoint
(way,2), +1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,1),
+1);SetTableCellVal (panel1, PANEL_TABLE, MakePoint (way,3), +1);}

    }

```

4. Función Interfases.

```

void longi(void) // FUNCION QUE PERMITE LEER Y DETERMINAR LA
CONFIGURACION DE LAS LONGITUDES DE ONDA.
{
    if (longi==1 )
        L1=1;
    if (longi==2 )
        L1=2;
    if (longi==3 )
        L1=3;
    if (longi==4 )
        L1=4;
    if (longi==5 )
        L1=5;
    if (longi==6 )
        L1=6;
    if (longi==7 )

```

```
L1=7;
if (longi==8 )
L1=8;
if (longi==9 )
L1=9;
if (longi==10 )
L1=10;
if (longi==11 )
L1=11;
if (longi==12 )
L1=12;
if (longi==13 )
L1=13;
if (longi==14 )
L1=14;
if (longi==15 )
L1=15;
if (longi==16 )
L1=16;
```

```
}
```

void waysplus(void) //FUNCION QUE DETERMINA QUE CAMINO HAY
ENTRE RUTAS PARA LA PRIMERA BUSQUEDA.

```
{

if(ruta[rut]==1 && ruta[rut+1]==2)
way=1;
if(ruta[rut]==2 && ruta[rut+1]==3)
way=2;
if(ruta[rut]==3 && ruta[rut+1]==4)
way=3;

if(ruta[rut]==6 && ruta[rut+1]==5)
way=8;
if(ruta[rut]==7 && ruta[rut+1]==6)
way=9;
if(ruta[rut]==8 && ruta[rut+1]==7)
way=10;

if(ruta[rut]==9 && ruta[rut+1]==10)
way=15;
if(ruta[rut]==10 && ruta[rut+1]==11)
way=16;
if(ruta[rut]==11 && ruta[rut+1]==12)
way=17;
```

```

if(ruta[rut]==4 && ruta[rut+1]==8)
    way=11;
if(ruta[rut]==5 && ruta[rut+1]==9)
    way=7;

    if ( way == 1 )
        {

            if(L1==1)
                SetCtrlVal (panel1, PANEL_LED_1,L);
            if(L1==2)
                SetCtrlVal (panel1, PANEL_LED_2,L);
            if(L1==3)
                SetCtrlVal (panel1, PANEL_LED_3,L);
            if(L1==4)
                SetCtrlVal (panel1, PANEL_LED_4,L);
            if(L1==5)
                {SetCtrlVal (panel1, PANEL_LED_1,L);
                SetCtrlVal (panel1, PANEL_LED_3,L);}
            if(L1==6)
                {SetCtrlVal (panel1, PANEL_LED_2,L);
                SetCtrlVal (panel1, PANEL_LED_3,L);}
            if(L1==7)
                {SetCtrlVal (panel1, PANEL_LED_3,L);
                SetCtrlVal (panel1, PANEL_LED_2,L);SetCtrlVal (panel1,
PANEL_LED_1,L);}
            if(L1==8)
                {SetCtrlVal (panel1, PANEL_LED_4,L);}
            if(L1==9)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_1,L);}
            if(L1==10)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_2,L);}
            if(L1==11)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_2,L);SetCtrlVal (panel1,
PANEL_LED_1,L);}
            if(L1==12)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_2,L);}
            if(L1==13)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_3,L);SetCtrlVal (panel1,
PANEL_LED_1,L);}
            if(L1==14)
                {SetCtrlVal (panel1, PANEL_LED_4,L);
                SetCtrlVal (panel1, PANEL_LED_2,L);SetCtrlVal (panel1,
PANEL_LED_3,L);}
            if(L1==15)

```



```

        {SetCtrlVal (panel1, PANEL_LED_4,L);
        SetCtrlVal (panel1, PANEL_LED_2,L);SetCtrlVal (panel1,
PANEL_LED_1,L);SetCtrlVal (panel1, PANEL_LED_3,L);}

    }

```

5. Función Resultados.

```

    if ( way == 2 )
    {

        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_5,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_6,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_7,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_8,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_5,L);
        SetCtrlVal (panel1, PANEL_LED_7,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_6,L);
        SetCtrlVal (panel1, PANEL_LED_7,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_7,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);SetCtrlVal (panel1,
PANEL_LED_5,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_8,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_5,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);}
        if(L1==11)

```

```

        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);SetCtrlVal (panel1,
PANEL_LED_5,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_7,L);SetCtrlVal (panel1,
PANEL_LED_5,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);SetCtrlVal (panel1,
PANEL_LED_7,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_8,L);
        SetCtrlVal (panel1, PANEL_LED_6,L);SetCtrlVal (panel1,
PANEL_LED_5,L);SetCtrlVal (panel1, PANEL_LED_7,L);}

```

```

    }

```

```

if ( way == 3 )
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_9,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_10,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_11,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_12,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_9,L);
        SetCtrlVal (panel1, PANEL_LED_11,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_10,L);
        SetCtrlVal (panel1, PANEL_LED_11,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_11,L);
        SetCtrlVal (panel1, PANEL_LED_10,L);SetCtrlVal
(panel1, PANEL_LED_9,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_12,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_9,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_12,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_10,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_10,L);SetCtrlVal
(panel1, PANEL_LED_9,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_10,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_11,L);SetCtrlVal
(panel1, PANEL_LED_9,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_10,L);SetCtrlVal
(panel1, PANEL_LED_11,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_12,L);
        SetCtrlVal (panel1, PANEL_LED_10,L);SetCtrlVal
(panel1, PANEL_LED_9,L);SetCtrlVal (panel1, PANEL_LED_11,L);}
    }

```

```

if ( way == 4 )
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_13,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_14,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_15,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_16,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_13,L);
        SetCtrlVal (panel1, PANEL_LED_15,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_14,L);
        SetCtrlVal (panel1, PANEL_LED_15,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_15,L);
        SetCtrlVal (panel1, PANEL_LED_14,L);SetCtrlVal
(panel1, PANEL_LED_13,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_16,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_13,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_16,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_14,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_14,L);SetCtrlVal
(panel1, PANEL_LED_13,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_14,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_15,L);SetCtrlVal
(panel1, PANEL_LED_13,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_14,L);SetCtrlVal
(panel1, PANEL_LED_15,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_16,L);
        SetCtrlVal (panel1, PANEL_LED_14,L);SetCtrlVal
(panel1, PANEL_LED_13,L);SetCtrlVal (panel1, PANEL_LED_15,L);}
    }
    if ( way == 5 )
        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_17,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_18,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_19,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_20,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_17,L);
        SetCtrlVal (panel1, PANEL_LED_19,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_18,L);
        SetCtrlVal (panel1, PANEL_LED_19,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_19,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);SetCtrlVal
(panel1, PANEL_LED_17,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_20,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_17,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);}
        if(L1==11)

```

```

        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);SetCtrlVal
(panel1, PANEL_LED_17,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_19,L);SetCtrlVal
(panel1, PANEL_LED_17,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);SetCtrlVal
(panel1, PANEL_LED_19,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_20,L);
        SetCtrlVal (panel1, PANEL_LED_18,L);SetCtrlVal
(panel1, PANEL_LED_17,L);SetCtrlVal (panel1, PANEL_LED_19,L);}
    }
    if ( way == 6 )
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_21,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_22,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_23,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_24,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_21,L);
        SetCtrlVal (panel1, PANEL_LED_23,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_22,L);
        SetCtrlVal (panel1, PANEL_LED_23,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_23,L);
        SetCtrlVal (panel1, PANEL_LED_22,L);SetCtrlVal
(panel1, PANEL_LED_21,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_24,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
        SetCtrlVal (panel1, PANEL_LED_21,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
        SetCtrlVal (panel1, PANEL_LED_22,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_24,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_22,L);SetCtrlVal
(panel1, PANEL_LED_21,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
SetCtrlVal (panel1, PANEL_LED_22,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
SetCtrlVal (panel1, PANEL_LED_23,L);SetCtrlVal
(panel1, PANEL_LED_21,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
SetCtrlVal (panel1, PANEL_LED_22,L);SetCtrlVal
(panel1, PANEL_LED_23,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_24,L);
SetCtrlVal (panel1, PANEL_LED_22,L);SetCtrlVal
(panel1, PANEL_LED_21,L);SetCtrlVal (panel1, PANEL_LED_23,L);}
    }
    if ( way == 7)
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_25,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_26,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_27,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_28,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_25,L);
SetCtrlVal (panel1, PANEL_LED_27,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_26,L);
SetCtrlVal (panel1, PANEL_LED_27,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_27,L);
SetCtrlVal (panel1, PANEL_LED_26,L);SetCtrlVal
(panel1, PANEL_LED_25,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_28,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_25,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_26,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_26,L);SetCtrlVal
(panel1, PANEL_LED_25,L);}

```

```

        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_26,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_27,L);SetCtrlVal
(panel1, PANEL_LED_25,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_26,L);SetCtrlVal
(panel1, PANEL_LED_27,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_28,L);
SetCtrlVal (panel1, PANEL_LED_26,L);SetCtrlVal
(panel1, PANEL_LED_25,L);SetCtrlVal (panel1, PANEL_LED_27,L);}
    }
    if ( way == 8)
    {
        if(L1==1)
SetCtrlVal (panel1, PANEL_LED_29,L);
        if(L1==2)
SetCtrlVal (panel1, PANEL_LED_30,L);
        if(L1==3)
SetCtrlVal (panel1, PANEL_LED_31,L);
        if(L1==4)
SetCtrlVal (panel1, PANEL_LED_32,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_29,L);
SetCtrlVal (panel1, PANEL_LED_31,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_30,L);
SetCtrlVal (panel1, PANEL_LED_31,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_31,L);
SetCtrlVal (panel1, PANEL_LED_30,L);SetCtrlVal
(panel1, PANEL_LED_29,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_32,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
SetCtrlVal (panel1, PANEL_LED_29,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
SetCtrlVal (panel1, PANEL_LED_30,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
SetCtrlVal (panel1, PANEL_LED_30,L);SetCtrlVal
(panel1, PANEL_LED_29,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_32,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_30,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
        SetCtrlVal (panel1, PANEL_LED_31,L);SetCtrlVal
(panel1, PANEL_LED_29,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
        SetCtrlVal (panel1, PANEL_LED_30,L);SetCtrlVal
(panel1, PANEL_LED_31,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_32,L);
        SetCtrlVal (panel1, PANEL_LED_30,L);SetCtrlVal
(panel1, PANEL_LED_29,L);SetCtrlVal (panel1, PANEL_LED_31,L);}
    }
    if ( way == 9)
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_33,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_34,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_35,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_36,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_33,L);
        SetCtrlVal (panel1, PANEL_LED_35,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_34,L);
        SetCtrlVal (panel1, PANEL_LED_35,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_35,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);SetCtrlVal
(panel1, PANEL_LED_33,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_36,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_33,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);SetCtrlVal
(panel1, PANEL_LED_33,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);}
        if(L1==13)

```



```

        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_35,L);SetCtrlVal
(panel1, PANEL_LED_33,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);SetCtrlVal
(panel1, PANEL_LED_35,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_36,L);
        SetCtrlVal (panel1, PANEL_LED_34,L);SetCtrlVal
(panel1, PANEL_LED_33,L);SetCtrlVal (panel1, PANEL_LED_35,L);}
    }
    if ( way == 10)
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_37,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_38,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_39,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_40,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_37,L);
        SetCtrlVal (panel1, PANEL_LED_39,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_38,L);
        SetCtrlVal (panel1, PANEL_LED_39,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_39,L);
        SetCtrlVal (panel1, PANEL_LED_38,L);SetCtrlVal
(panel1, PANEL_LED_37,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_40,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
        SetCtrlVal (panel1, PANEL_LED_37,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
        SetCtrlVal (panel1, PANEL_LED_38,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
        SetCtrlVal (panel1, PANEL_LED_38,L);SetCtrlVal
(panel1, PANEL_LED_37,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
        SetCtrlVal (panel1, PANEL_LED_38,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_40,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_39,L);SetCtrlVal
(panel1, PANEL_LED_37,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
SetCtrlVal (panel1, PANEL_LED_38,L);SetCtrlVal
(panel1, PANEL_LED_39,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_40,L);
SetCtrlVal (panel1, PANEL_LED_38,L);SetCtrlVal
(panel1, PANEL_LED_37,L);SetCtrlVal (panel1, PANEL_LED_39,L);}
    }
    if ( way == 11)
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_41,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_42,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_43,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_44,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_41,L);
SetCtrlVal (panel1, PANEL_LED_43,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_42,L);
SetCtrlVal (panel1, PANEL_LED_43,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_43,L);
SetCtrlVal (panel1, PANEL_LED_42,L);SetCtrlVal
(panel1, PANEL_LED_41,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_44,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
SetCtrlVal (panel1, PANEL_LED_41,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
SetCtrlVal (panel1, PANEL_LED_42,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
SetCtrlVal (panel1, PANEL_LED_42,L);SetCtrlVal
(panel1, PANEL_LED_41,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
SetCtrlVal (panel1, PANEL_LED_42,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
SetCtrlVal (panel1, PANEL_LED_43,L);SetCtrlVal
(panel1, PANEL_LED_41,L);}

```

```

        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
        SetCtrlVal (panel1, PANEL_LED_42,L);SetCtrlVal
(panel1, PANEL_LED_43,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_44,L);
        SetCtrlVal (panel1, PANEL_LED_42,L);SetCtrlVal
(panel1, PANEL_LED_41,L);SetCtrlVal (panel1, PANEL_LED_43,L);}
    }
    if ( way == 12)
    {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_45,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_46,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_47,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_48,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_45,L);
        SetCtrlVal (panel1, PANEL_LED_47,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_46,L);
        SetCtrlVal (panel1, PANEL_LED_47,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_47,L);
        SetCtrlVal (panel1, PANEL_LED_46,L);SetCtrlVal
(panel1, PANEL_LED_45,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_48,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_45,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_46,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_46,L);SetCtrlVal
(panel1, PANEL_LED_45,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_46,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_47,L);SetCtrlVal
(panel1, PANEL_LED_45,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_48,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_46,L);SetCtrlVal
(panel1, PANEL_LED_47,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_48,L);
        SetCtrlVal (panel1, PANEL_LED_46,L);SetCtrlVal
(panel1, PANEL_LED_45,L);SetCtrlVal (panel1, PANEL_LED_47,L);}
        }
        if ( way == 13)
        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_49,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_50,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_51,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_52,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_49,L);
        SetCtrlVal (panel1, PANEL_LED_51,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_50,L);
        SetCtrlVal (panel1, PANEL_LED_51,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_51,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);SetCtrlVal
(panel1, PANEL_LED_49,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_52,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_49,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);SetCtrlVal
(panel1, PANEL_LED_49,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_51,L);SetCtrlVal
(panel1, PANEL_LED_49,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);SetCtrlVal
(panel1, PANEL_LED_51,L);}

```

```

        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_52,L);
        SetCtrlVal (panel1, PANEL_LED_50,L);SetCtrlVal
(panel1, PANEL_LED_49,L);SetCtrlVal (panel1, PANEL_LED_51,L);}
        }
        if ( way == 14)
        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_53,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_54,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_55,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_56,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_53,L);
        SetCtrlVal (panel1, PANEL_LED_55,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_54,L);
        SetCtrlVal (panel1, PANEL_LED_55,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_55,L);
        SetCtrlVal (panel1, PANEL_LED_54,L);SetCtrlVal
(panel1, PANEL_LED_53,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_56,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_53,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_54,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_54,L);SetCtrlVal
(panel1, PANEL_LED_53,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_54,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_55,L);SetCtrlVal
(panel1, PANEL_LED_53,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_56,L);
        SetCtrlVal (panel1, PANEL_LED_54,L);SetCtrlVal
(panel1, PANEL_LED_55,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_56,L);

```

```

        SetCtrlVal (panel1, PANEL_LED_54,L);SetCtrlVal
(panel1, PANEL_LED_53,L);SetCtrlVal (panel1, PANEL_LED_55,L);}
    }
    if ( way == 15)
        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_57,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_58,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_59,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_60,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_57,L);
        SetCtrlVal (panel1, PANEL_LED_59,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_58,L);
        SetCtrlVal (panel1, PANEL_LED_59,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_59,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);SetCtrlVal
(panel1, PANEL_LED_57,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_60,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_57,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);SetCtrlVal
(panel1, PANEL_LED_57,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_59,L);SetCtrlVal
(panel1, PANEL_LED_57,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);SetCtrlVal
(panel1, PANEL_LED_59,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_60,L);
        SetCtrlVal (panel1, PANEL_LED_58,L);SetCtrlVal
(panel1, PANEL_LED_57,L);SetCtrlVal (panel1, PANEL_LED_59,L);}

```

```

    }
    if ( way == 16)
        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_61,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_62,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_63,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_64,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_61,L);
        SetCtrlVal (panel1, PANEL_LED_63,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_62,L);
        SetCtrlVal (panel1, PANEL_LED_63,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_63,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);SetCtrlVal
(panel1, PANEL_LED_61,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_64,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_61,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);SetCtrlVal
(panel1, PANEL_LED_61,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_63,L);SetCtrlVal
(panel1, PANEL_LED_61,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);SetCtrlVal
(panel1, PANEL_LED_63,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_64,L);
        SetCtrlVal (panel1, PANEL_LED_62,L);SetCtrlVal
(panel1, PANEL_LED_61,L);SetCtrlVal (panel1, PANEL_LED_63,L);}
        }
    if ( way == 17)

```

```

        {
        if(L1==1)
        SetCtrlVal (panel1, PANEL_LED_65,L);
        if(L1==2)
        SetCtrlVal (panel1, PANEL_LED_66,L);
        if(L1==3)
        SetCtrlVal (panel1, PANEL_LED_67,L);
        if(L1==4)
        SetCtrlVal (panel1, PANEL_LED_68,L);
        if(L1==5)
        {SetCtrlVal (panel1, PANEL_LED_65,L);
        SetCtrlVal (panel1, PANEL_LED_67,L);}
        if(L1==6)
        {SetCtrlVal (panel1, PANEL_LED_66,L);
        SetCtrlVal (panel1, PANEL_LED_67,L);}
        if(L1==7)
        {SetCtrlVal (panel1, PANEL_LED_67,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);SetCtrlVal
(panel1, PANEL_LED_65,L);}
        if(L1==8)
        {SetCtrlVal (panel1, PANEL_LED_68,L);}
        if(L1==9)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_65,L);}
        if(L1==10)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);}
        if(L1==11)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);SetCtrlVal
(panel1, PANEL_LED_65,L);}
        if(L1==12)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);}
        if(L1==13)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_67,L);SetCtrlVal
(panel1, PANEL_LED_65,L);}
        if(L1==14)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);SetCtrlVal
(panel1, PANEL_LED_67,L);}
        if(L1==15)
        {SetCtrlVal (panel1, PANEL_LED_68,L);
        SetCtrlVal (panel1, PANEL_LED_66,L);SetCtrlVal
(panel1, PANEL_LED_65,L);SetCtrlVal (panel1, PANEL_LED_67,L);}
        }
    }

```



```
void ways (void) //FUNCION QUE DETERMINA QUE LONGITUDD DE  
ONDA LIBRE ENTRE RUTAS PARA LA PRIMERA BUSQUEDA.
```

```
{  
    timery=0.02;  
    SetCtrlVal (panel1, PANEL_LED_reset,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_1,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_2,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_3,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_4,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_5,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_6,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_7,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_16,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_9,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_10,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_11,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_12,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_13,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_14,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_15,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_16,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_17,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_18,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_19,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_20,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_21,L);  
    Delay(timery);  
    SetCtrlVal (panel1, PANEL_LED_22,L);  
    Delay(timery);  
}
```

```
SetCtrlVal (panel1, PANEL_LED_23,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_24,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_25,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_26,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_27,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_28,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_29,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_30,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_31,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_32,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_33,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_34,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_35,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_36,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_37,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_38,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_39,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_40,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_41,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_42,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_43,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_44,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_45,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_46,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_47,L);
Delay(timery);
```

```
SetCtrlVal (panel1, PANEL_LED_48,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_49,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_50,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_51,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_52,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_53,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_54,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_55,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_56,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_57,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_58,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_59,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_60,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_61,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_62,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_63,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_64,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_65,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_66,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_67,L);
Delay(timery);
SetCtrlVal (panel1, PANEL_LED_68,L);

}
```