



SIMIOBOARD SC-PASCAL

GUIA RAPIDA

Doc. Ver.: SIMIOB-QUICKSTART V.1.0 ES

Date: 28/07/2013

URL: www.simioboard.com



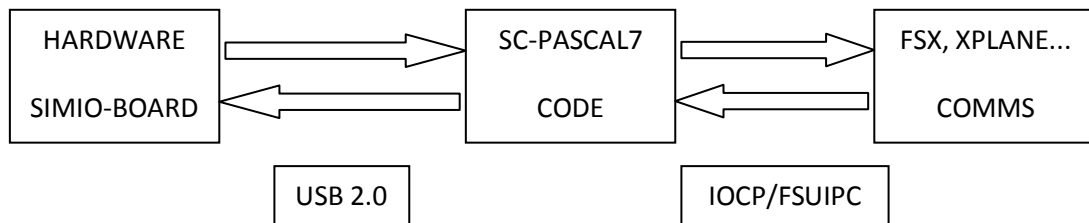
INDICE

1. Introducción.....	3
2. Pre-requisitos.....	3
3. Ejemplo inicial	4
4. Ejemplo Conexión	11
5. Ejemplo Conexiones y Hardware.	20



1. Introducción

Para comenzar a usar SIMIO BOARDs o SC-PASCAL7 no es necesario tener muchos conocimientos de programación, pero si tener claro la estructura de SC-PASCAL7. Hay 3 partes fundamentales en SC-PASCAL7:



Como podemos ver, SC-PASCAL7 está entre el hardware y el simulador, por lo que es fácil unir ambos, que es nuestro propósito final.

No hay que gestionar la parte de comunicaciones entre SC-PASCAL7 y el simulador, ya que esta la hace internamente el compilador.

2. Pre-requisitos.

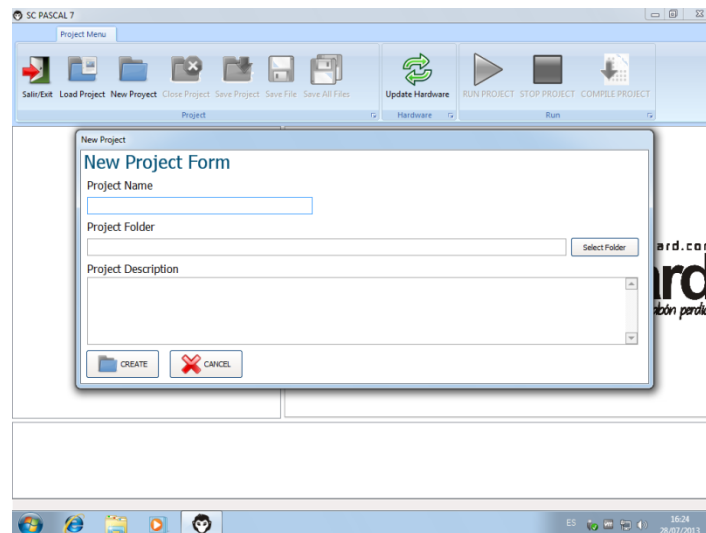
Para seguir este QuickStart se necesita lo siguiente:

- a. FSX
- b. IOCP Server para FSX
- c. FSUIPC para FSX
- d. SC-PASCAL7
- e. Tarjeta SIMIO BOARD USB MAIN, solo capítulo 4.
- f. Tarjeta SIMIO BOARD EXP 32 DISPLAYS, solo capítulo 4.
- g. Un encoder conectado en la tarjeta MAIN, solo capítulo 4.
- h. 6 Displays de 7 segmentos conectados a la tarjeta de displays, solo capítulo 4.
- i. Manual del Programador SC-PASCAL7.

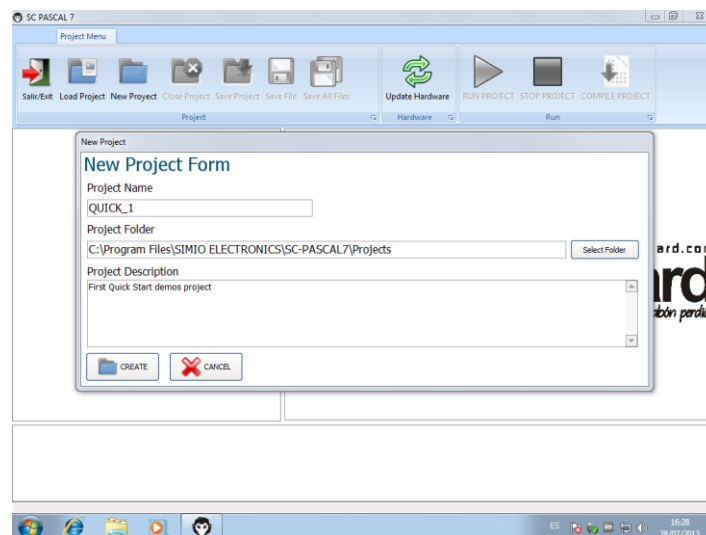


3. Ejemplo inicial

Como ejemplo inicial, vamos a ejecutar un código en SC-PASCAL7 sin usar la parte de hardware ni la del simulador. Para ello crearemos un proyecto con el cual vamos a iniciar un bucle que cuando llegue a cierto valor muestre un mensaje.

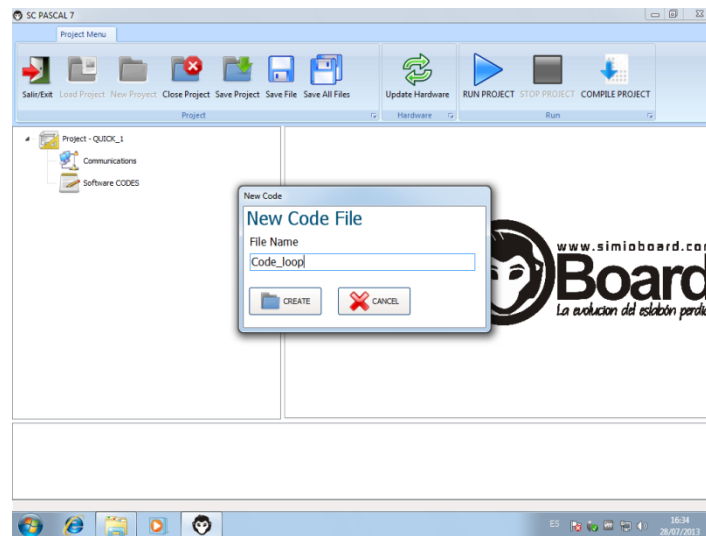


1. Creamos el proyecto, le ponemos como nombre "QUICK_1" y seleccionamos la carpeta "projects" donde está instalado SC-PASCAL7.

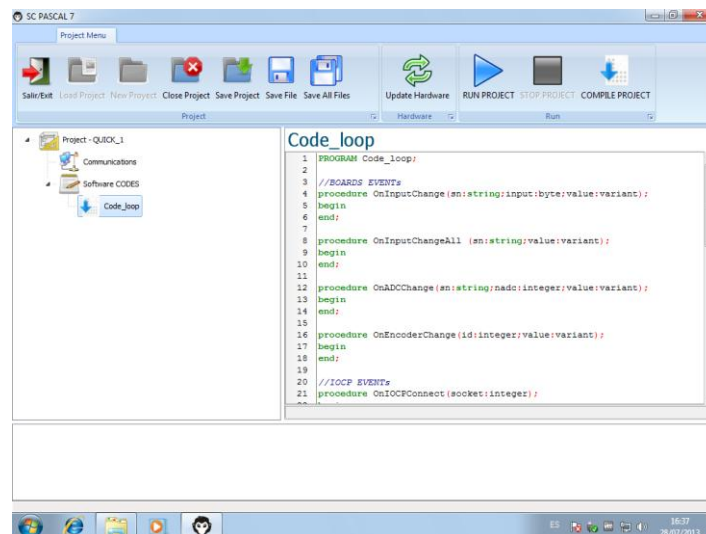




2. Pulsamos el botón derecho del ratón encima de "Software CODES" de la ventana de objetos y seleccionamos "New CODE".



3. En la ventana de New Code escribimos el nombre del código que queremos crear, en este caso "Code_loop", y pulsamos "CREATE".
4. Una vez creado el fichero de código, pulsamos sobre él y nos aparece en el área de trabajo.



5. Nos desplazamos por el código hasta encontrar esta estructura:

//MAIN

Begin

//WRITE HERE YOUR CODE

End.



6. Esta estructura nos indica que es el procedimiento MAIN, o principal desde el que empieza la ejecución de este código.
7. Dentro de esta estructura vamos a crear una estructura repetitiva FOR (ver capítulo 6.3 del Manual del Programador de SC-PASCAL7).

Begin

for x:=1 to 10 do begin

end;

//WRITE HERE YOUR CODE

End.

8. Como vemos hemos creado una estructura donde se ejecuta lo que hay dentro desde x=1 hasta x=10, por lo que se ejecutará 10 veces.
9. Ahora dentro de la estructura FOR, vamos a crear un condicional IF (ver capítulo 5.1 del Manual del Programador de SC-PASCAL7). En este condicional vamos hacer una acción cuando x=5, que en este caso será poner un mensaje en el debug que diga que x ya vale 5.

Begin

for x:=1 to 10 do begin

if x=5 then begin

debug('X ya vale 5');

end;

end;

End.



10. Ahora como final vamos a poner otro mensaje cuando la estructura FOR termine diciendo que el programa ha terminado.

Begin

for x:=1 to 10 do begin

if x=5 then begin

debug('X ya vale 5');

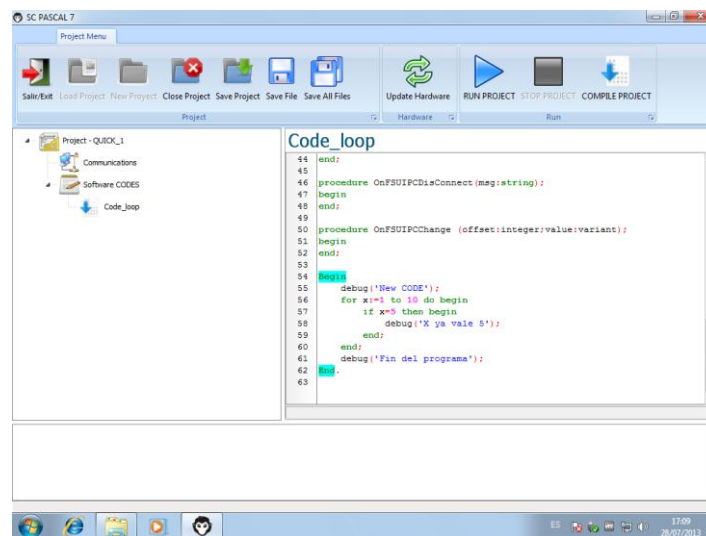
end;

end;

debug('Fin del programa');

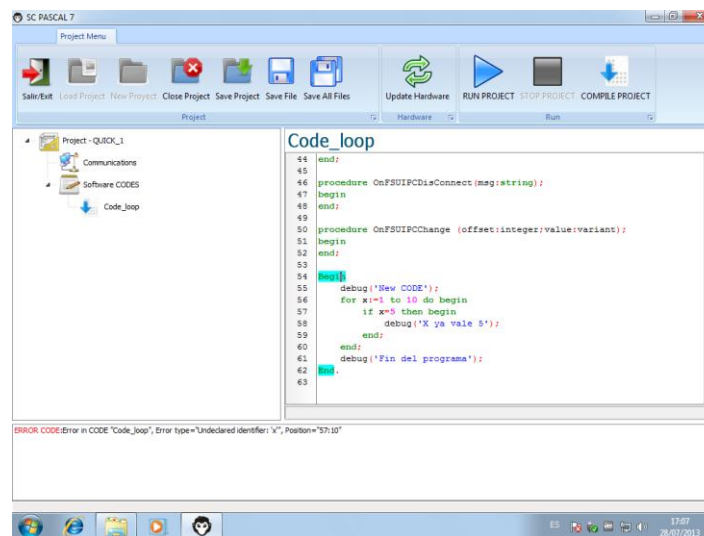
End.

11. Ahora compilamos el código. Para ello primero pulsamos sobre el icono del “Save File” que tenemos en la barra de menú y luego pulsamos de nuevo botón derecho del ratón sobre el nombre del código que aparece en la ventana de objetos.





12. Como podemos comprobar tenemos un error de compilación.

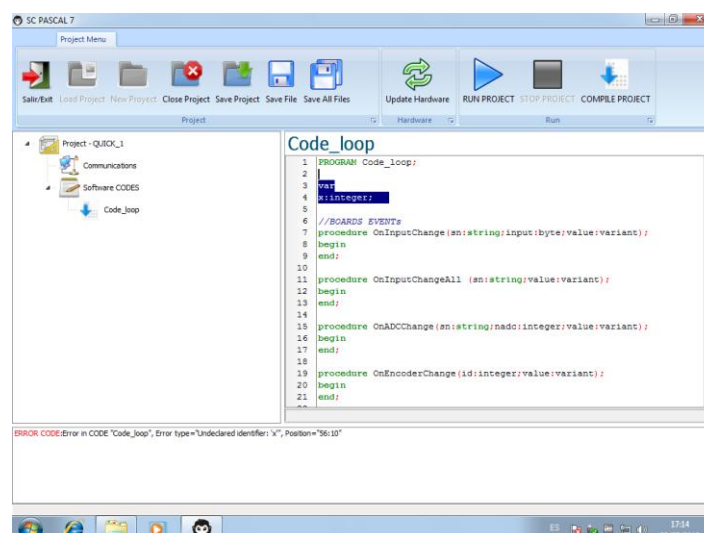


13. El error nos indica la línea 57, fila 10 del código. Como podemos comprobar no hemos declarado qué es X, por lo que lo hacemos en la parte superior, debajo de la clausula “PROGRAM”. Para ello tenemos que poner esta declaración.

var

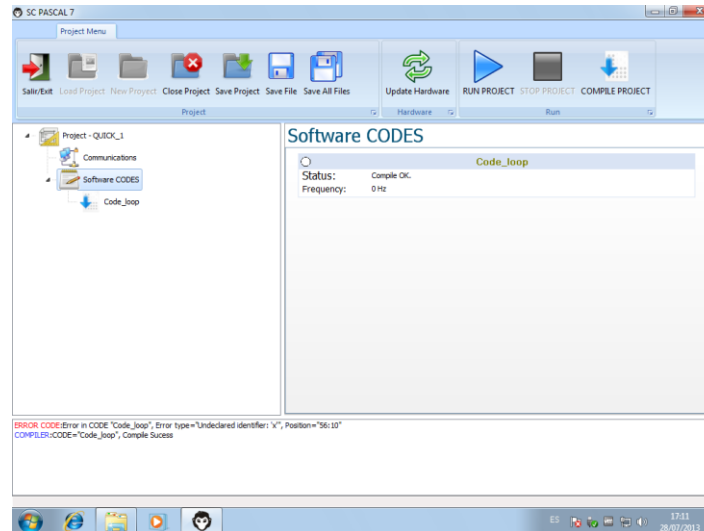
x:integer;

14. Una vez declarada la variable x, volvemos a pulsar “Save File” antes de volver a compilar. Es conveniente en este punto salvar el proyecto pulsando sobre “Save Project”, que se encuentra igualmente en la barra de menú.

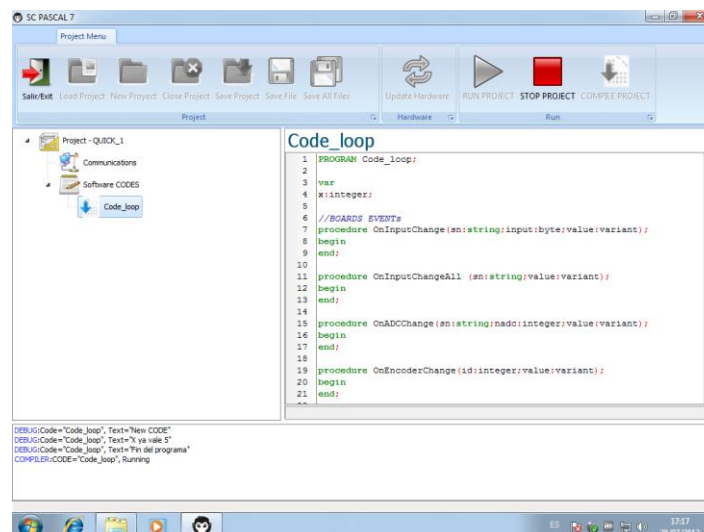




15. Volvemos a compilar con el botón derecho del ratón sobre el nombre del código en la ventana de objetos. Ahora tenemos un “Compile OK”, eso quiere decir que no hay errores y podemos ejecutarlo.

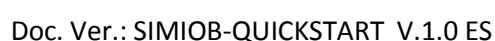
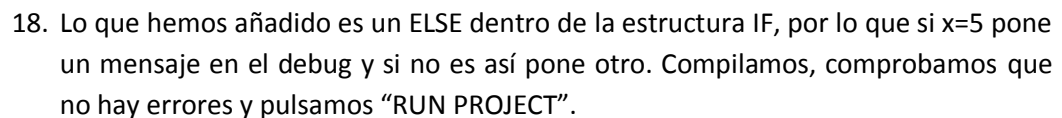


16. Ahora pulsamos “RUN PROJECT”.





End.

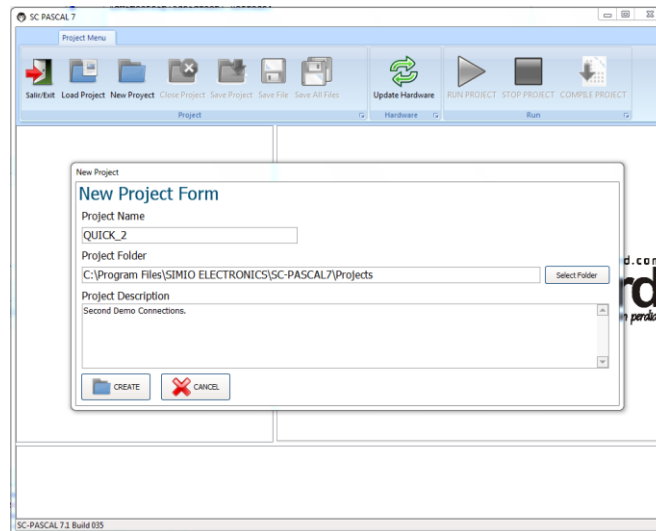




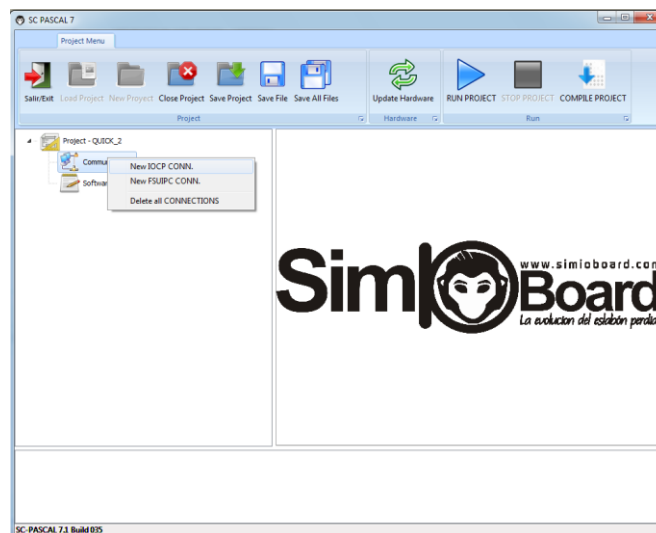
4. Ejemplo Conexión

El siguiente ejemplo muestra cómo conectarse a FSX mediante FSUIPC y mediante IOCP y obtener el valor de un offset. Primero haremos las dos conexiones por separado y luego el mismo ejemplo con diferente conexión.

1. Abrimos SC-PASCAL7 y creamos un nuevo proyecto.

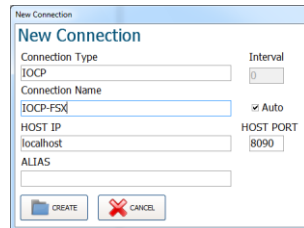


2. Pulsamos con el botón derecho del ratón encima de Communications en la ventana de objetos y pulsamos "New IOCP Conn".

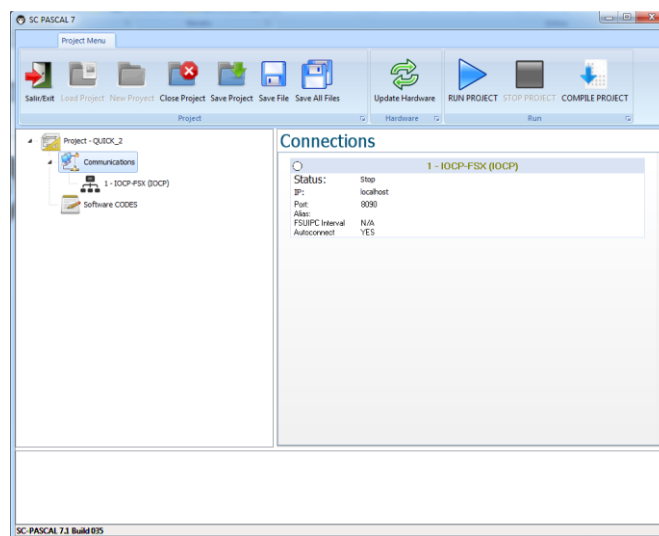




3. Completamos los datos de conexión. En este caso nombramos la conexión como "IOCP-FSX" y en el campo "HOST IP" ponemos localhost, ya que el simulador se ejecuta en el mismo PC que SC-PASCAL7.



4. Pulsamos "CREATE" y comprobamos la creación de la conexión.

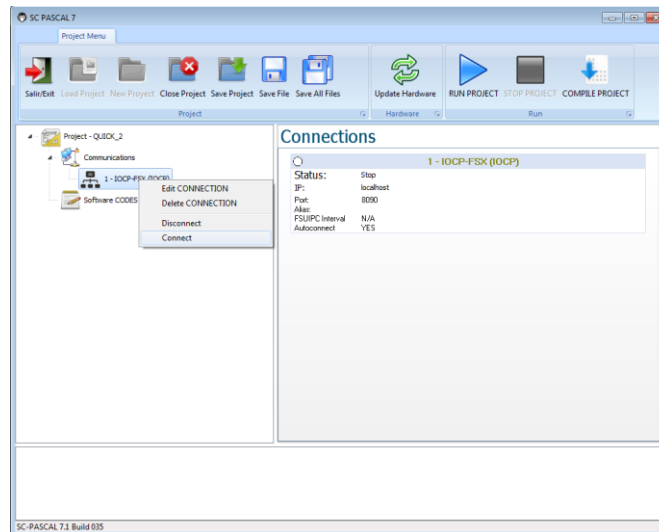


5. Una vez en este punto, arrancamos FSX.

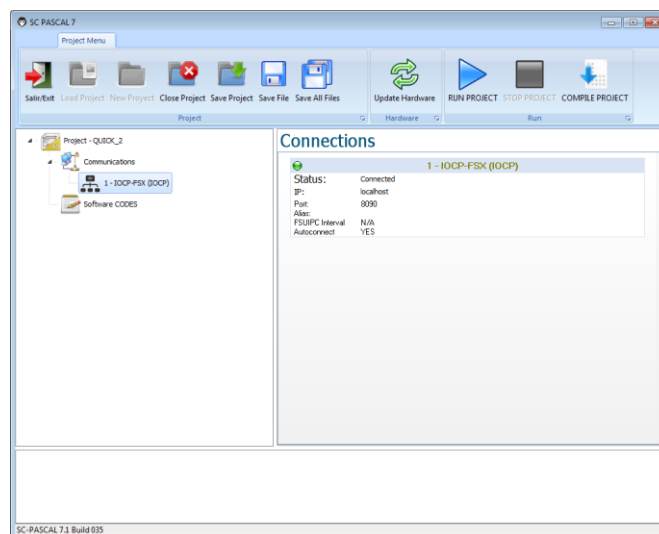




- Ahora pulsamos botón derecho sobre el nombre de la conexión en la ventana de objetos y pulsamos "Connect".



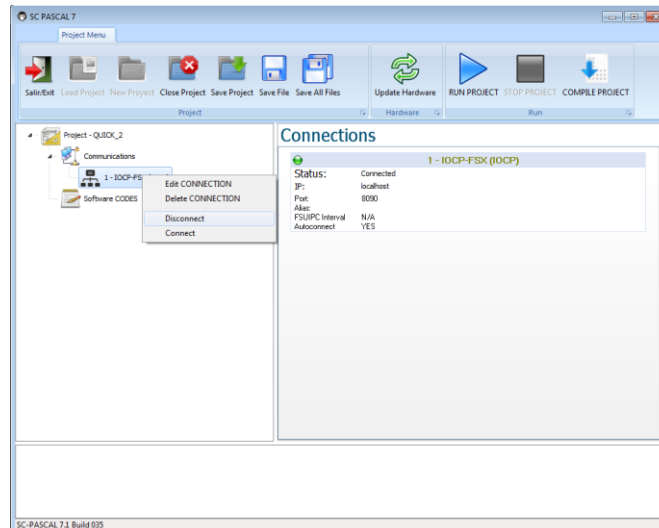
- Comprobamos que se conecta correctamente.



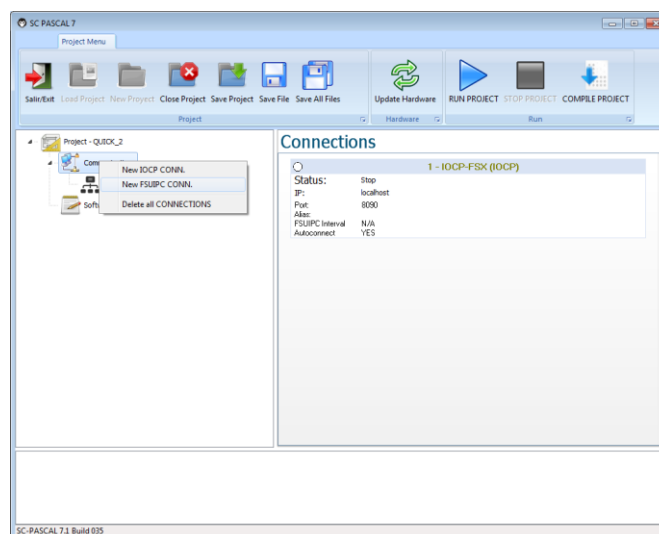
- Como podemos ver el icono de la conexión está en verde y en "STATUS" tenemos "Connected", por lo que la conexión es correcta.



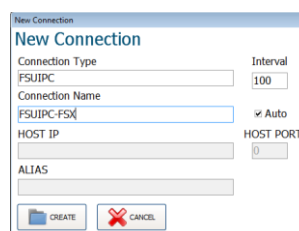
9. Volvemos a pulsar botón derecho sobre la conexión y pulsamos “Disconnect”



10. Ahora volvemos a pulsar botón derecho encima de “Communications” en la ventana de objetos y pulsamos “New FSUIPC Conn.”

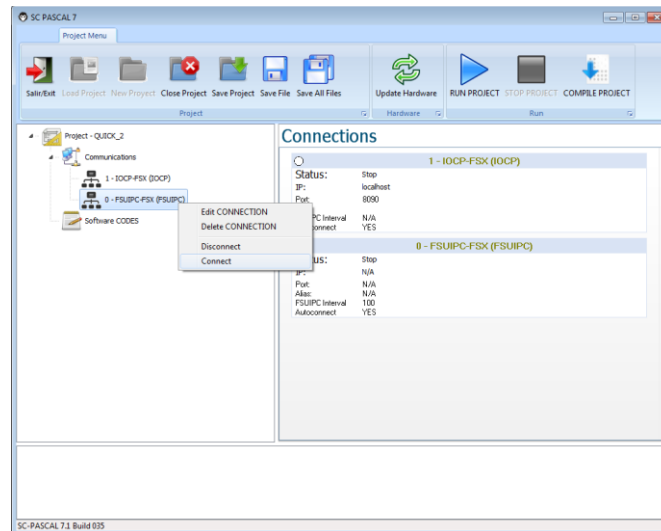


11. Rellenamos los datos de la conexión. En este caso, al ser FSUIPC, sólo necesitamos poner un nombre y dejar en 100 milisegundos el campo “Interval”.

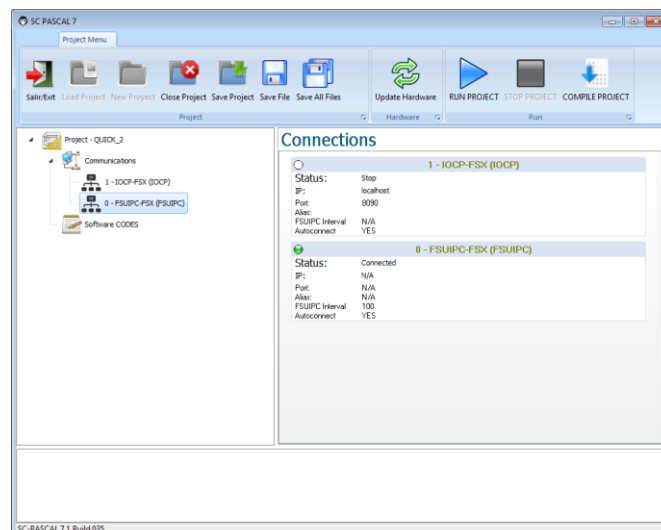




12. Pulsamos "CREATE" y pulsamos sobre "Communications" para comprobar que están las dos conexiones. En este punto y teniendo en FSX instalado FSUIPC, podemos probar dicha conexión.
13. Pulsamos Botón derecho sobre la conexión FSUIPC en la ventana de objetos y pulsamos "Connect".



14. Comprobamos que la conexión se ha realizado correctamente al cambiar el icono a color verde y en "STATUS" aparece como "Connected". Eso quiere decir que la conexión es correcta.

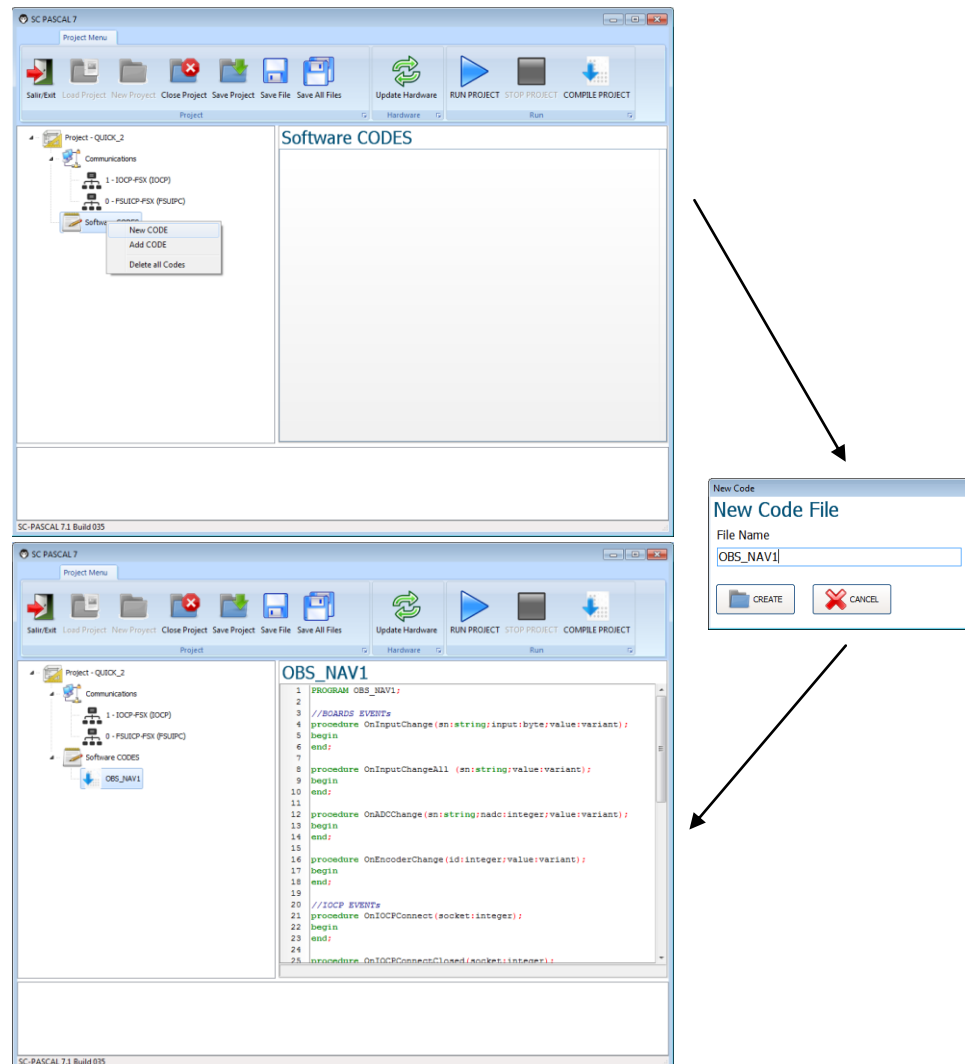


15. Ahora podemos pulsar botón derecho del ratón encima del nombre de la conexión FSUIPC y pulsamos "Disconnect". En este punto tenemos comprobado los dos tipos de conexión, así que vamos a leer un offset mediante las dos conexiones. Para este ejemplo vamos a usar OBS_NAV1 en las dos conexiones. Para IOCP es



offset es el 43 y para FSUIPC es \$0C4E. También sabemos que los valores tanto de lectura como de escritura van desde 0 a 359 ó desde 1 a 360.

16. Creamos un nuevo código pulsado botón derecho del ratón sobre “Software CODES” en la ventana de objetos y pulsamos “New Code”. Introducimos un nombre para el fichero, por ejemplo “OBS_NAV1” y pulsamos “CREATE”.



17. Lo primero que tenemos que hacer es registrar cada uno de los offsets. Primero lo hacemos con IOCP (ver punto 7.3 Manual de Programación de SC-PASCAL7).

`RegIOCPOffset(1,43);`

18. Tenemos que colocar esta línea en el evento OnIOCPConnect ya que una vez conectado le decimos qué offsets vamos a querer. Hemos cambiado el “1” por “socket” ya que este contiene el identificador de la conexión.



```
//IOCP EVENTS
```

```
procedure OnIOCPConnect(socket:integer);
```

```
begin
```

```
    RegIOCPOffset(socket,43);
```

```
end;
```

19. Una vez registrado el offset, esperamos que si en el simulador cambia ese valor, automáticamente y mediante un evento SC-PASCAL7 nos comuniqué que ha cambiado. Eso se produce en el evento “OnIOCPChange”, por lo que en dicho evento pondremos un debug para ver qué valor nos está llegando.

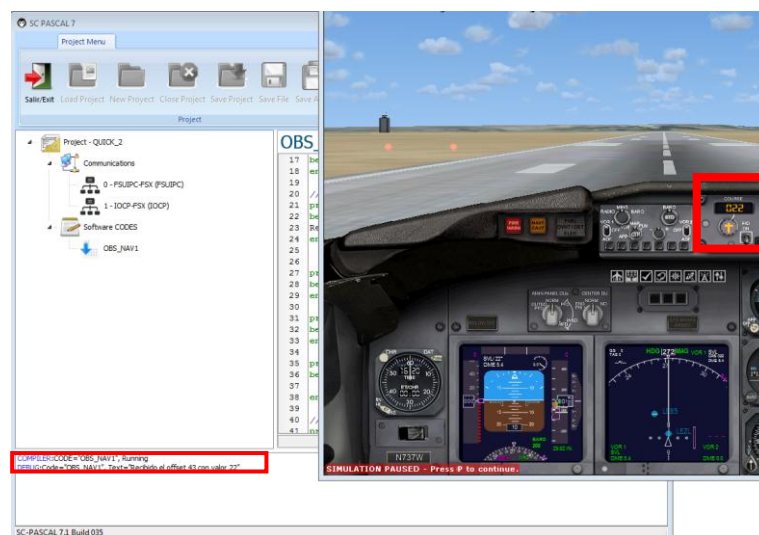
```
procedure OnIOCPChange(socket,offset,value:integer);
```

```
begin
```

```
    debug('Recibido el offset '+inttostr(offset)+' con valor '+inttostr(value));
```

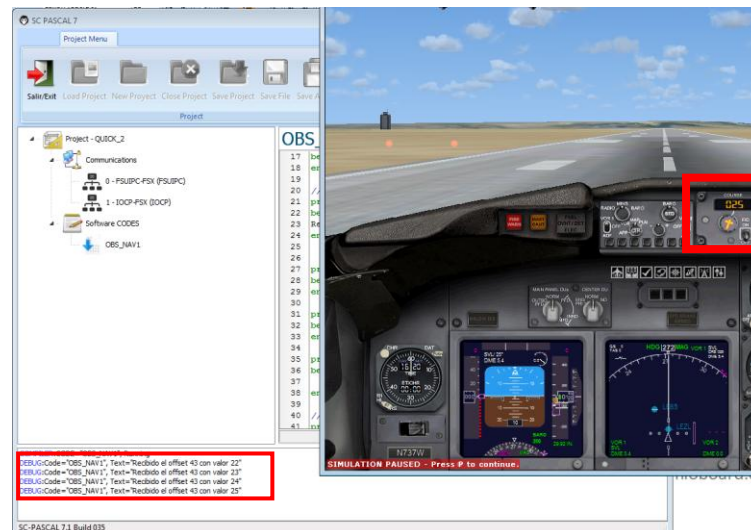
```
end;
```

20. Evaluemos qué hemos hecho antes de compilar y ejecutar. Queremos que el simulador nos envíe el valor del OBS_NAV1. Para ello registramos el offset 43 en el evento OnIOCPConnect. Luego, cuando en el simulador cambie ese valor, SC-PASCAL7 hará que salte el evento OnIOCPChange. En sólo dos pasos recibimos el valor del OBS_NAV1. Ahora ejecutamos el proyecto y el resultado es el siguiente:





21. Como vemos, en la ventana de debug, hemos recibido el offset 43 que tiene el valor 22 que coincide con el valor en FSX. Ahora pulsamos en FSX en el control del OBS para que cambie el valor y vemos como van cambiando los valores recibidos en la ventana de mensajes.



22. Desde aquí podemos usar este valor para por ejemplo enviarlo a una tarjeta de displays.
23. Ahora vamos a hacer lo mismo pero con FSUIPC. Al igual que con IOCP, tenemos que registrar el offset en FSUIPC para que recibamos su valor. En este caso el offset es el \$0C4E de 2 bytes. Primero tenemos que registrarlo cuando se haya producido la conexión, y usar InitOffsetFSUIPC (ver punto 7.4 del Manual de Programación de SC-PASCAL7). El evento es OnFSUIPCConnect.

```
procedure OnFSUIPCConnect (msg:string);
```

```
begin
```

```
InitOffsetFSUIPC($0C4E,2)
```

```
end;
```

24. Ahora en el evento, cuando recibimos el valor lo mostramos en la ventana de debug.

```
procedure OnFSUIPCChange (offset:integer;value:variant);
```

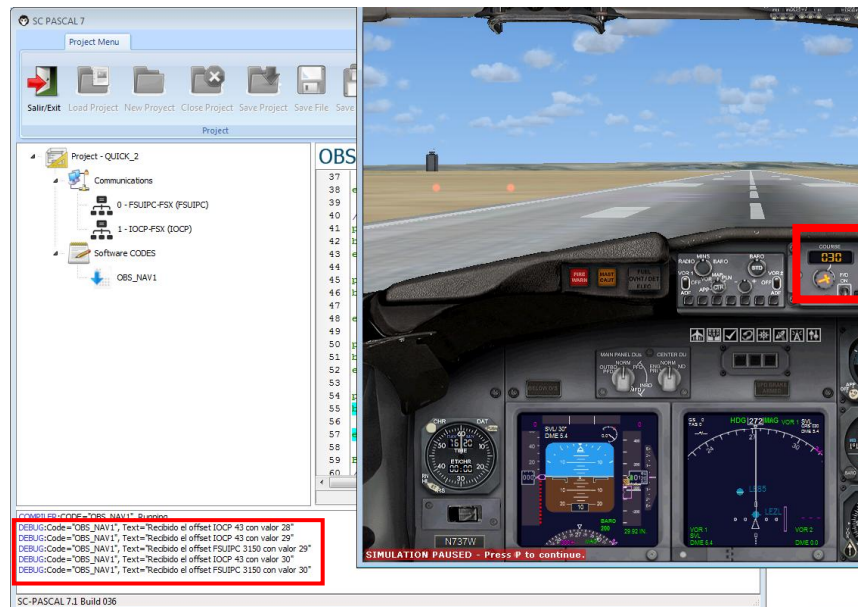
```
begin
```

```
debug('Recibido el offset FSUIPC '+inttostr(offset)+' con valor '+inttostr(value));
```

```
end;
```



25. Para diferenciar el offset de IOCP y de FSUIPC, ponemos el nombre de la conexión en el debug.

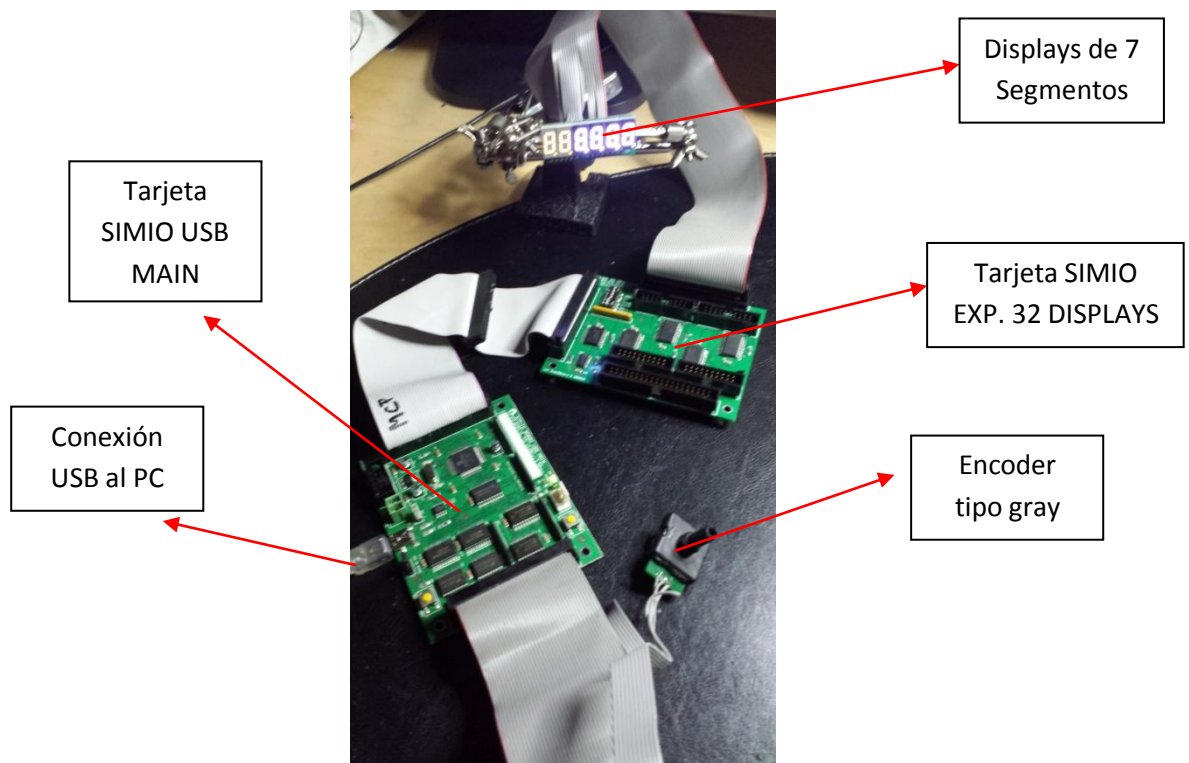


26. Como se puede ver, cuando cambiamos el OBS_NAV1 o COURSE en el simulador, en SC-PASCAL7 recibimos ese valor por las dos conexiones.
27. En este punto estamos listos para pasar al ejemplo 3, donde usaremos el hardware.
28. Fin ejemplo de Conexiones.



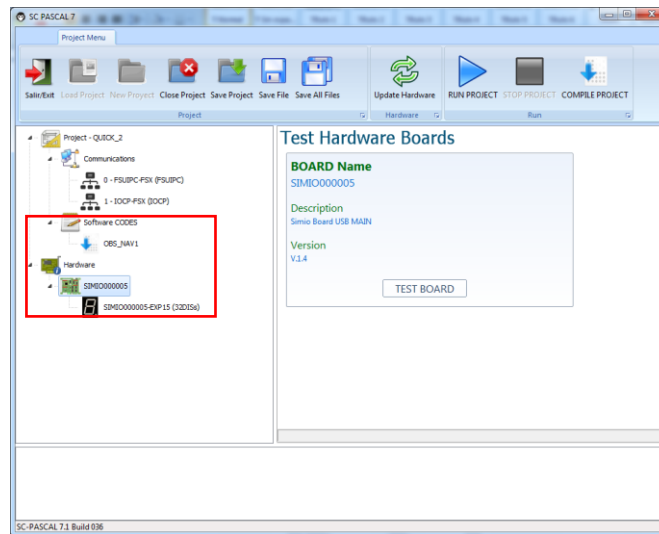
5. Ejemplo Conexiones y Hardware.

Para este ejemplo vamos a usar el mismo proyecto anterior al que vamos a añadir el hardware. Para ello vamos a usar una tarjeta SIMIO BOARD USB MAIN y una expansión SIMIO BOARD 32 DISPLAYS. En la tarjeta Main vamos a conectar un encoder tipo gray, y en la de displays 6 displays de 7 segmentos de los que solo usaremos los 3 primeros. Una vez comprobado el hardware, crearemos el código necesario para que al mover el encoder mande ese valor al simulador y posteriormente, al recibir el valor en SC-PASCAL7 desde el simulador, lo mostremos en los displays. A continuación un esquema del hardware conectado al ordenador.

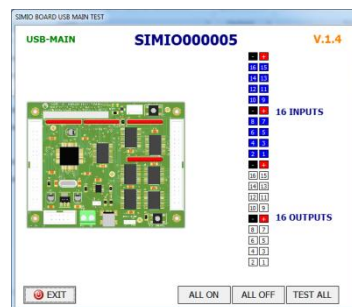




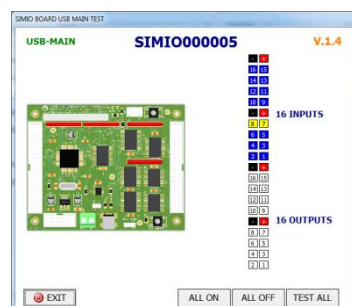
1. Una vez conectado todo al ordenador tenemos que tener el proyecto del ejemplo anterior y además debe aparecer en la ventana de objetos, el hardware.



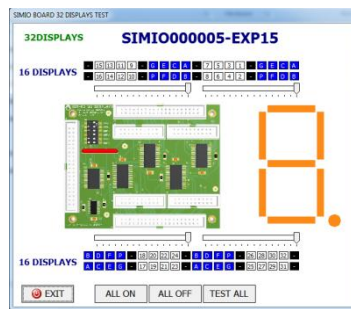
2. Pulsamos sobre la tarjeta Main SIMIO000005 y en la ventana de trabajo pulsamos sobre el botón "TEST BOARD".



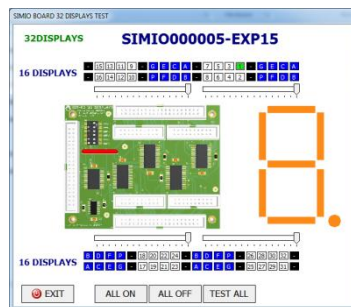
3. Ahora movemos el encoder para ver donde está conectado.



4. Como vemos, han cambiado de color las entradas 7 y 8, por lo que ya sabemos en qué entradas está conectado el encoder.
5. Cerramos la ventana de testeo de la tarjeta Main.
6. Ahora pulsamos sobre la expansión de 32 displays "SIMIO000005-EXP15" y en la ventana de trabajo pulsamos sobre el botón "TEST BOAD".



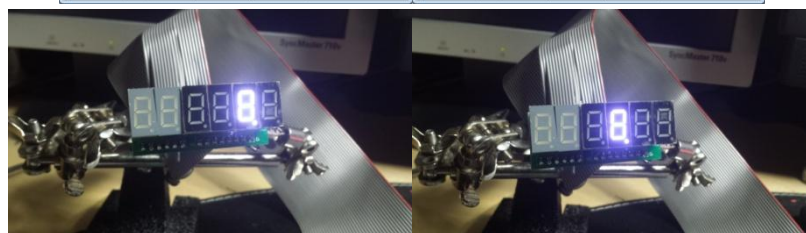
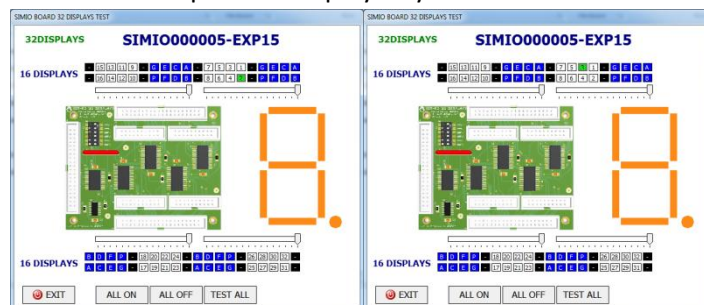
7. Ahora testamos los displays. Para esto pulsamos sobre el digito 1.



8. Comprobamos si se enciende el digito.



9. Ahora repetimos lo mismo para los displays 2 y 3.

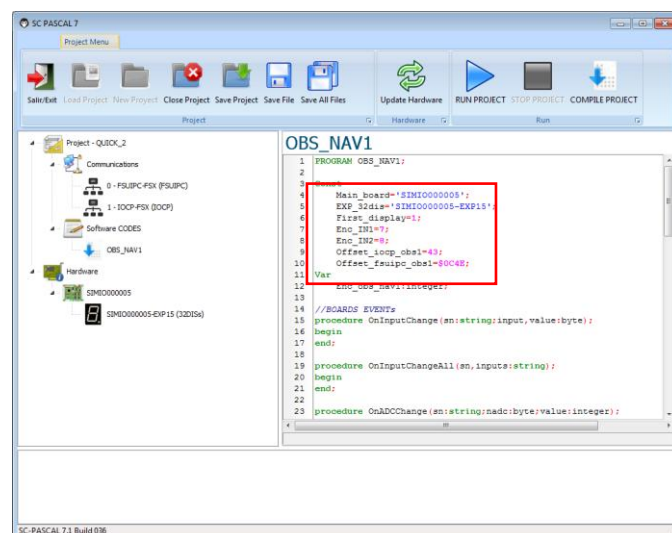


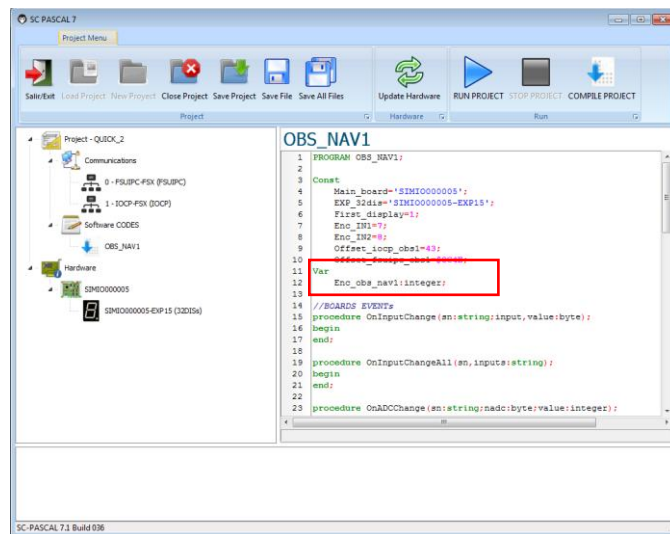


10. Ya tenemos todo lo necesario. Por un lado las conexiones, por otro el hardware y solo nos queda completar el código para que funcione todo.
11. Tenemos que tener muy claro el esquema hardware que tenemos:
 - a. 1 Tarjeta SIMIO USB MAIN: SIMIO000005.
 - b. 1 Tarjeta SIMIO EXP. 32 DISPLAYS: SIMIO000005-EXP15.
 - c. 3 Displays en la tarjeta de displays que empieza por el display 1.
 - d. 1 Encoder conectado a la tarjeta Main en las entradas 7 y 8.
12. Lo primero es definir todo este hardware en el código. Lo haremos en la clausula CONST.

Const

```
Main_board='SIMIO000005';  
  
EXP_32dis='SIMIO000005-EXP15';  
  
First_display=1;  
  
Enc_IN1=7;  
  
Enc_IN2=8;  
  
Offset_iocp_obs1=43;  
  
Offset_fsuipc_obs1=$0C4E;
```





14. Ahora en el procedimiento MAIN definimos el encoder (ver capítulo 7.7 del Manual de Programación de SC-PASCAL7).

Begin

```
Enc_obs_nav1:=RegEncoder(Main_board, Enc_IN1, Enc_IN2,0,360,1,0,false,100);
```

End.

15. Ya tenemos registrado el encoder, por lo que ahora cuando movemos dicho encoder el evento OnEncoderChange nos notificará que el valor del encoder ha cambiado. Es en ese evento donde cogeremos el valor del encoder y lo enviaremos al simulador. Como estamos usando IOCP, para enviar ese valor al simulador usaremos WriteIOCP (ver capítulo 7.7 del Manual de Programación de SC-PASCAL7).

```
procedure OnEncoderChange(id_encoder:integer;value:integer);
```

begin

```
if id_encoder=Enc_obs_nav1 then begin
```

```
WriteIOCP(1,Offset_iocp_obs1,value);
```

```
end;
```

end;

16. Como podéis ver hemos introducido una condicional IF dentro del evento. Este condicional nos asegura que la información que nos llega es la del encoder que queremos y no de otro. Cuando tengamos más encoders veréis que es imprescindible.



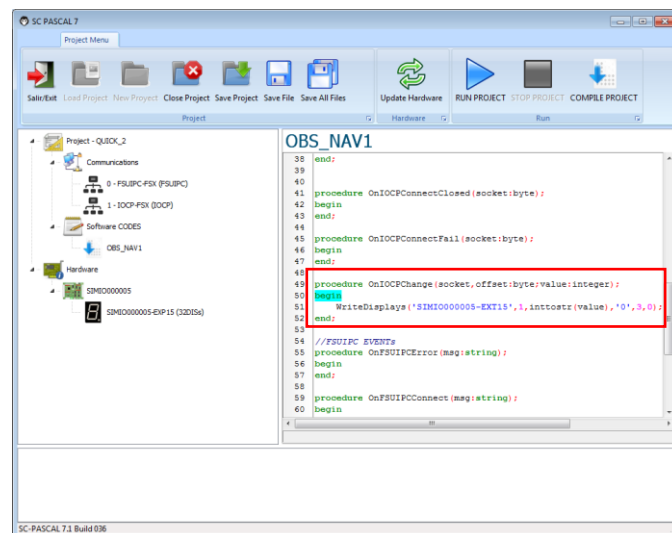
17. Ejecutamos el proyecto y veremos que al mover el encoder se mueve el COURSE del simulador.
18. Por último, solo nos queda que el valor del simulador se represente en los displays. Para ello en el evento OnIOCPChange recibiremos el valor del OBS_NAV1 o COURSE, y es aquí donde lo enviaremos a los displays con el procedimiento WriteDisplays (ver capítulo 7.8 del Manual de Programación de SC-PASCAL7).

```
procedure OnIOCPChange(socket,offset:byte;value:integer);
```

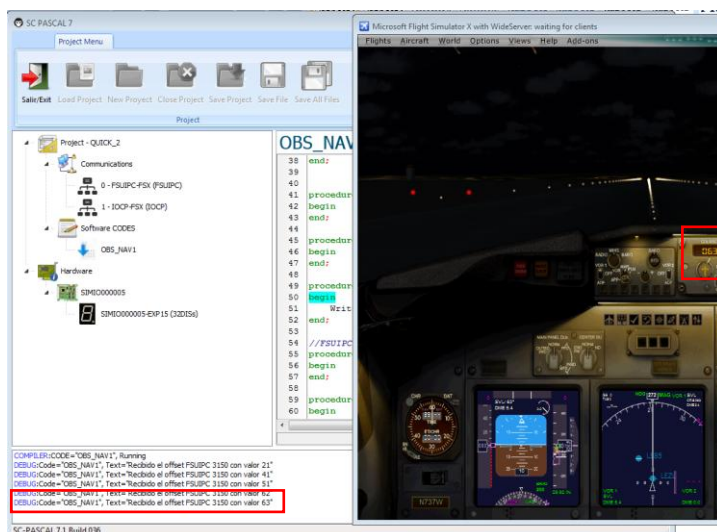
```
begin
```

```
WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
```

```
end;
```



19. Ejecutamos el proyecto y comprobamos que funciona todo correctamente.





20. Todo el código quedaría así eliminando lo que sobra del proyecto por defecto.

```
PROGRAM OBS_NAV1;
Const
  Main_board='SIMIO000005';
  EXP_32dis='SIMIO000005-EXP15';
  First_display=1;
  Enc_IN1=7;
  Enc_IN2=8;
  Offset_iocp_obs1=43;
  Offset_fsuipc_obs1=$0C4E;
Var
  Enc_obs_nav1:integer;

//BOARDS EVENTS
procedure OnEncoderChange(id_encoder,value:integer);
begin
  if id_encoder=Enc_obs_nav1 then begin
    WriteIOP(1, Offset_iocp_obs1,value);
  end;
end;

//IOP EVENTS
procedure OnIOCPConnect(socket:byte);
begin
  RegIOCPOffset(socket, Offset_iocp_obs1);
end;

procedure OnIOCPChange(socket,offset:byte;value:integer);
begin
  WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
end;

//MAIN
Begin
  Enc_obs_nav1:=RegEncoder(Main_board, Enc_IN1, Enc_IN2,0,360,1,0,false,100);
End.
```



21. Este ejemplo se ha hecho con IOCP. Para hacer lo mismo en FSUIPC deberemos cambiar muy pocas cosas, sólo donde se escribe el valor del encoder y cuando recibamos el offset por el evento OnFSUIPCChange, enviar el valor a los displays. Así quedaría el mismo proyecto pero por FSUIPC.

```
PROGRAM OBS_NAV1;
Const
  Main_board='SIMIO000005';
  EXP_32dis='SIMIO000005-EXP15';
  First_display=1;
  Enc_IN1=7;
  Enc_IN2=8;
  Offset_iocp_obs1=43;
  Offset_fsuipc_obs1=$0C4E;
Var
  Enc_obs_nav1:integer;

//BOARDS EVENTS
procedure OnEncoderChange(id_encoder,value:integer);
begin
  if id_encoder=Enc_obs_nav1 then begin
    WriteFSUIPC(Offset_fsuipc_obs1,2,value);
  end;
end;

//FSUIPC EVENTS
procedure OnFSUIPCConnect(msg:string);
begin
  InitOffsetFSUIPC(Offset_fsuipc_obs1,2)
end;

procedure OnFSUIPCChange(offset:integer;value:variant);
begin
  WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
end;

//MAIN
Begin
  Enc_obs_nav1:=RegEncoder(Main_board, Enc_IN1, Enc_IN2,0,360,1,0,false,100);
End.
```



22. Como mejora del código y como introducción de una nueva estructura de Pascal, en los eventos OnIOCPChange y OnFSUIPCChange, vamos a diferenciar lo que hacemos según el offset que nos llega.
23. En OnIOCPChange llegarán todos los cambios de todos los offsets que hemos registrado, por lo que el código que hemos hecho no nos funcionaria correctamente si registrásemos más de un offset.

```
procedure OnIOCPChange(socket,offset:byte;value:integer);
begin
    WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
end;
```

24. En este caso usaremos una estructura CASE (ver capítulo 5.2 del Manual de Programación de SC-PASCAL7).

```
procedure OnIOCPChange(socket,offset:byte;value:integer);
begin
    case offset of
        Offset_iocp_obs1:WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
    end;
end;
```

25. El mismo ejemplo pero con FSUIPC en el evento OnFSUIPCChange.

```
procedure OnFSUIPCChange(offset:integer;value:variant);
begin
    WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
end;
```

Quedando

```
procedure OnFSUIPCChange(offset:integer;value:variant);
begin
    case offset of
        Offset_fsuipc_obs1:WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
    end;
end;
```

26. Como ejemplo final añadiremos otro offset, la velocidad indicada IAS que se mostrará en los displays 4 a 6 de la tarjeta de 32 displays. Para esto buscamos el offset tanto en IOCP como en FSUIPC. El código de más abajo mezcla los dos códigos de IOCP y FSUIPC con color azul lo que hay que añadir.

```
PROGRAM OBS_NAV1;
```

```
Const
```

```
    Main_board='SIMIO000005';
    EXP_32dis='SIMIO000005-EXP15';
```



```
First_display=1;
Enc_IN1=7;
Enc_IN2=8;
Offset_iocp_obs1=43;
Offset_fsuipc_obs1=$0C4E;
Offset_iocp_IAS=114;
Offset_fsuipc_IAS=$07E2;
Second_display=4;
Var
  Enc_obs_nav1:integer;

//BOARDS EVENTS
procedure OnEncoderChange(id_encoder,value:integer);
begin
  if id_encoder=Enc_obs_nav1 then begin
    WriteIOCP(1,Offset_iocp_obs1,value);
    WriteIOCP(1,Offset_iocp_IAS,value);
    //WriteFSUIPC(Offset_fsuipc_obs1,2,value);
    //WriteFSUIPC(Offset_fsuipc_IAS,2,value);
  end;
end;

//IOCP EVENTS
procedure OnIOCPConnect(socket:byte);
begin
  RegIOCPOffset(socket,Offset_iocp_obs1);
  RegIOCPOffset(socket,Offset_iocp_IAS);
end;

procedure OnIOCPChange(socket,offset:byte;value:integer);
begin
  case offset of
    Offset_iocp_obs1:WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);
    Offset_iocp_IAS:WriteDisplays(EXP_32dis, Second_display,inttostr(value),'0',3,0);
  end;
end;

//FSUIPC EVENTS
procedure OnFSUIPCConnect(msg:string);
begin
  InitOffsetFSUIPC(Offset_fsuipc_obs1,2);
  InitOffsetFSUIPC(Offset_fsuipc_IAS,2);
end;
```



```
procedure OnFSUIPCChange(offset:integer;value:variant);
begin
    //case offset of

    //Offset_fsuipc_obs1:WriteDisplays(EXP_32dis, First_display,inttostr(value),'0',3,0);

    //Offset_fsuipc_IAS:WriteDisplays(EXP_32dis, Second_display,inttostr(value),'0',3,0);

    //end;
end;

//MAIN
Begin
    Enc_obs_nav1:=RegEncoder(Main_board, Enc_IN1, Enc_IN2,0,360,1,0,false,100);
End.
```