

Circuitos Digitales con **MyOpenLab**



Build 3.0.8.0

José Manuel Ruiz Gutiérrez

j.m.r.gutierrez@gmail.com

PRÁCTICAS DE CIRCUITOS DIGITALES

Circuitos digitales básicos

1. Simulación de operadores lógicos básicos.

Realizar la simulación de los operadores lógicos OR, AND, NOR, NAND y rellenar las tablas de verdad correspondientes. *circuito1.vlogic*

Función OR		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Función AND		
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Función NOR		
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Función NAND		
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



Utilizar interruptores y diodos LED para simular las entradas y las salidas

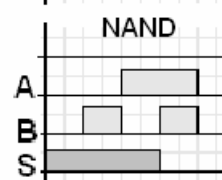
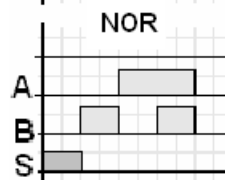
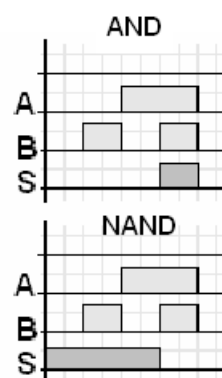
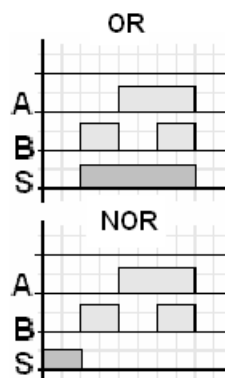
respectivamente pertenecientes a la librería boolean del panel frontal



Utilizar la librería de circuitos digitales del panel de circuito

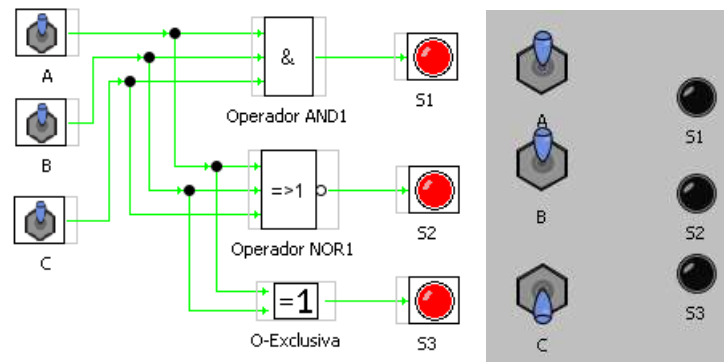


Rellenar el cronograma de cada función.



2. Deducción de tabla de verdad dado un circuito.

Dado el circuito de la figura observa que se han colocado dos operadores AND y NOR de tres entradas. Realiza la simulación y cumplimenta la tabla de verdad.

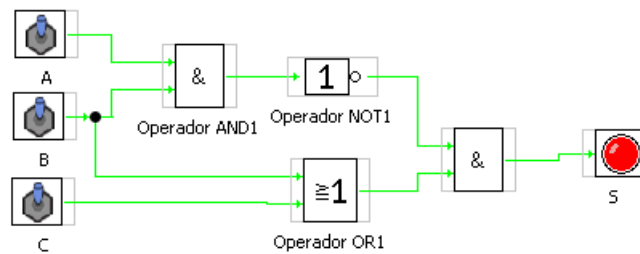


Solución: *circuito2.vlogic*

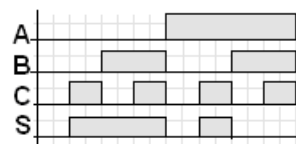
A	B	C	S1	S2	S3
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	1	0	0

3. Deducción de tabla de verdad y el cronograma dado un circuito.

Dado el circuito de la figura, monta el circuito y rellena la tabla de verdad y el cronograma haciendo uso del simulador.



Solución: *circuito3.vlogic*

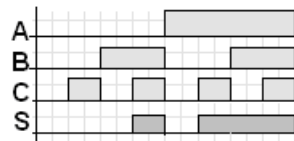
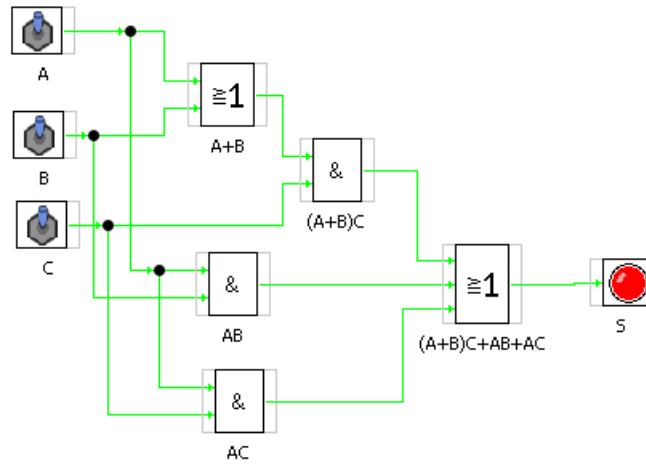


A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

4. Dada la ecuación averiguar el circuito, simularlo y comprobar su funcionamiento.

Dada la ecuación $S = (A+B)C + AB + AC$ dibuja el circuito con ayuda del simulador y prueba su comportamiento escribiendo la correspondiente tabla de verdad y cumplimenta el cronograma


Solución: *circuito4.vlogic*




A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Alarma doméstica básica.

Queremos realizar un circuito automático para que active una alarma que nos indique si se ha producido en nuestra casa un escape de agua, un escape de gas o si ha entrado alguien por la ventana.

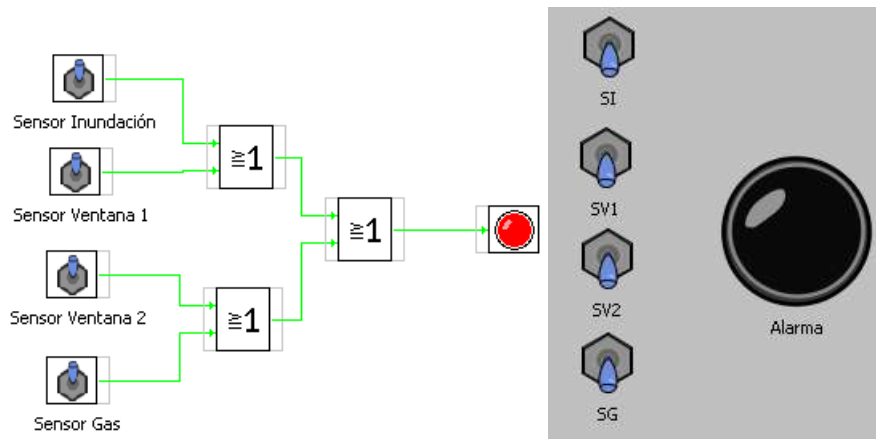
Disponemos de los siguientes sensores :

Sensor de inundación	SI	Sensor de ventana1	SV1
Sensor de escape de gas	SG	Sensor de ventana 2	SV2

La salida queremos que sea una lámpara  que se encienda cuando cualquiera de estos sensores se active (se ponga en “1”).

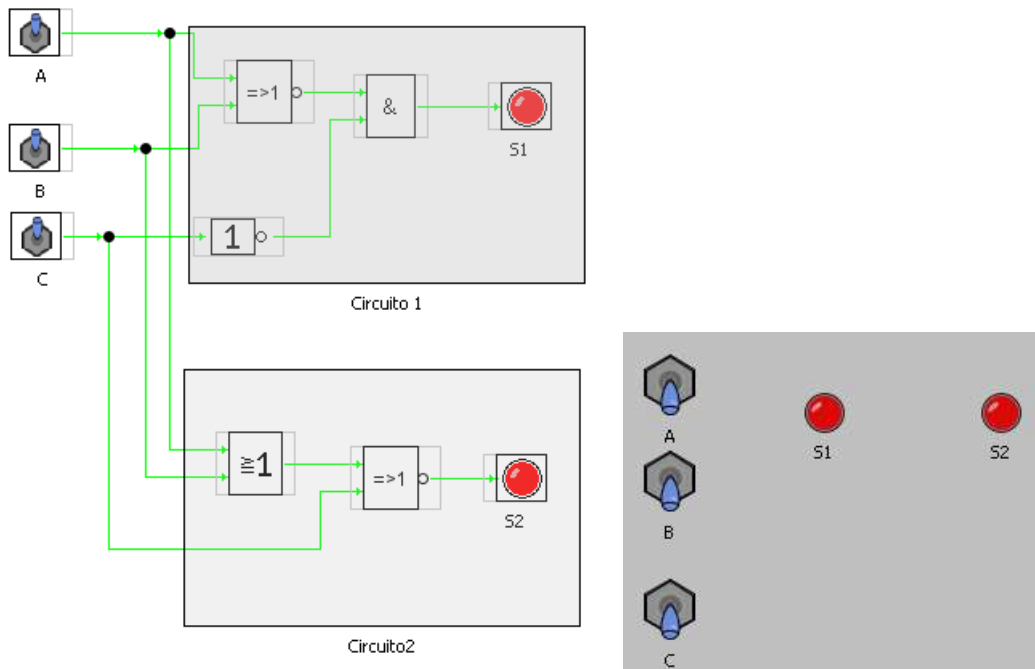
Realizar el circuito y simular su comportamiento.

Solución: *circuito5.vlogic*



5. Circuitos equivalentes.

Comprobar si los siguientes circuitos son equivalentes.



Nota: Para que dos circuitos sean equivalentes tienen que tener las mismas tablas de verdad

Solución: *circuito6.vlogic*

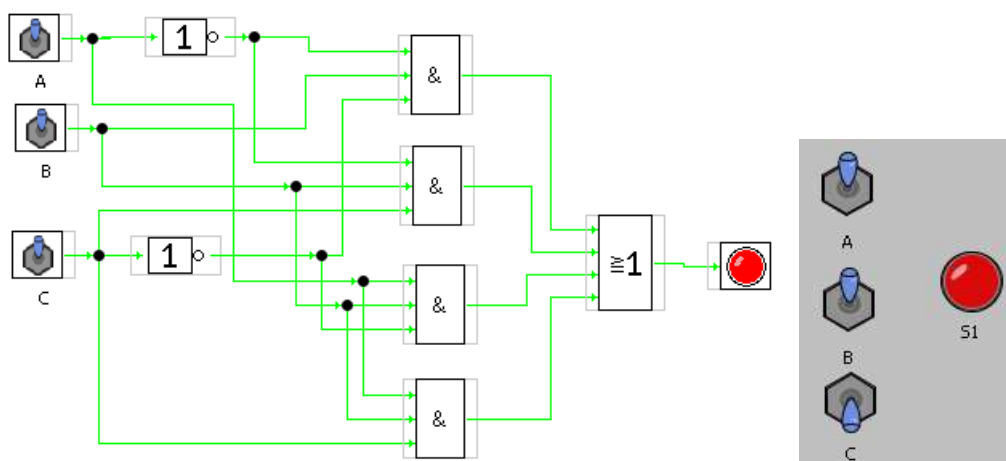
A	B	C	S1	S2
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

6. Dada la tabla diseñar el circuito e intentar simplificarlo.

Dada la tabla de verdad diseña el circuito correspondiente y simúlalo. Después intenta simplificar la función y si se puede simplifica prueba a simular el nuevo circuito simplificado y comprueba que has realizado bien la simplificación

A	B	C	S1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Solución: *circuito7.vlogic*

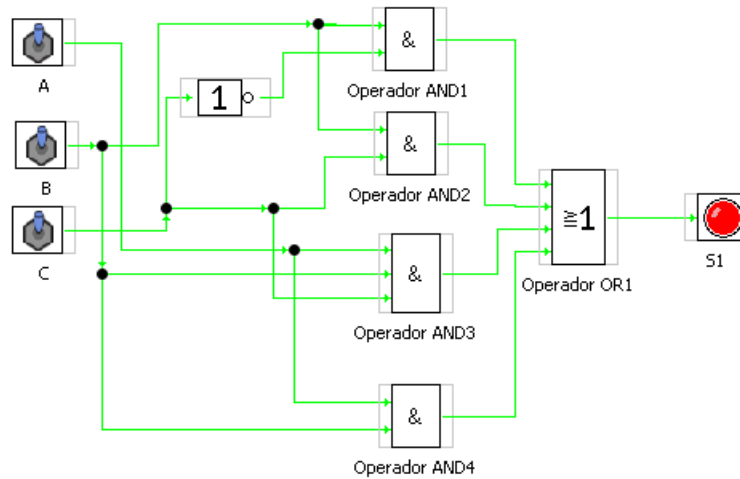


Simplificado



7. Averiguar la tabla de verdad y simplificar.

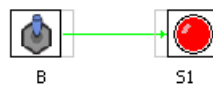
Monta el circuito de la figura simúlalo y rellena su tabla de verdad. Intenta simplificar la ecuación del circuito y si lo consigues realiza el nuevo circuito de la función simplificada y simúlalo también.



Solución: *circuito8.vlogic*

A	B	C	S1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Circuito simplificado



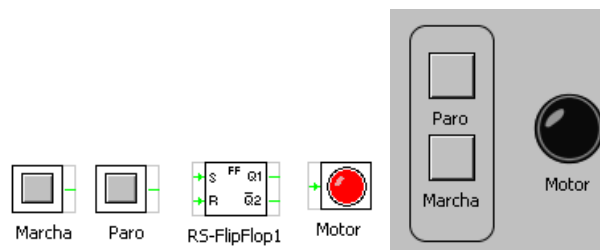
Circuitos SECUENCIALES

8. Realización de un circuito Paro-Marcha.

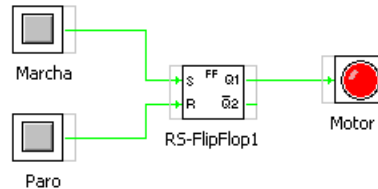
Realiza el circuito de mando de un motor mediante dos pulsadores “paro” y “marcha” utilizando como elemento básico un biestable tipo RS.

Funcionamiento: Cuando pulsamos sobre el botón marcha se activará el motor, si pulsamos mas veces sobre el botón no ocurrirá nada (ya está en marcha el motor), si pulsamos el botón paro se desactivará el motor.

Elementos a utilizar en el montaje:



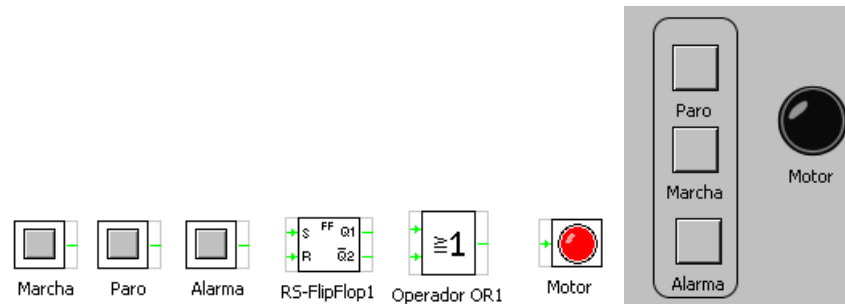
Solución: *circuito9.vlogic*



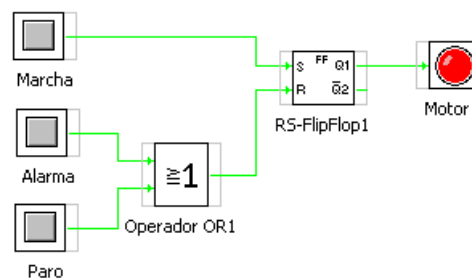
9. Realizar un circuito Paro-Marcha con Alarma.

Realizar el montaje anterior pero añadiendo una señal de alarma que cuando se active también se pare el motor

Elementos a utilizar en el montaje:

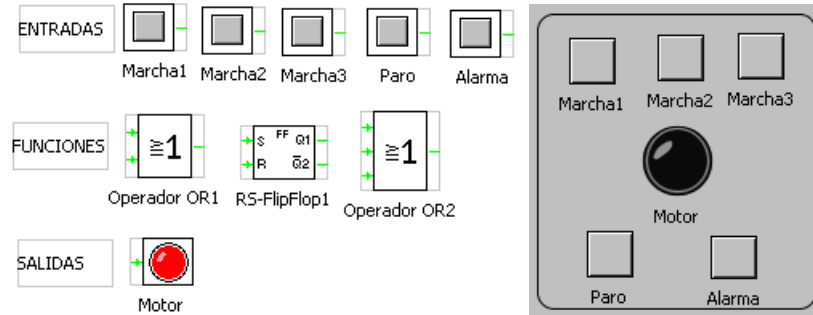


Solución: *circuito10.vlogic*

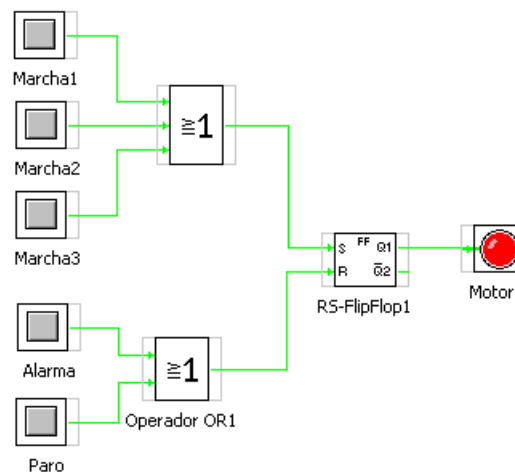


10. Gobierno de un motor desde 3 sitios diferentes.

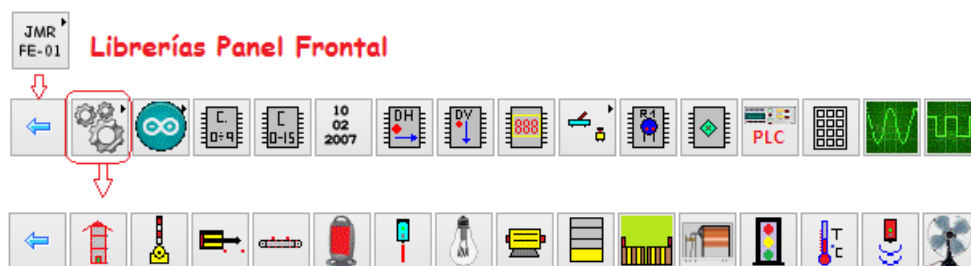
Realizar el montaje anterior pero con la posibilidad de poder activar el motor desde tres sitios diferentes.



Solución: *circuito11.vlogic*



Utilizando las librerías de “panel frontal” realiza los siguientes ejercicios.



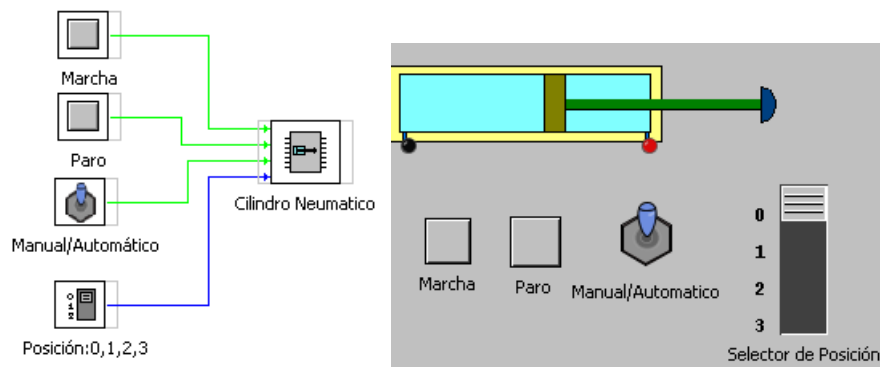
11. Gobierno de un cilindro neumático.

Se trata de investigar el funcionamiento de un cilindro neumático que puede trabajar de dos formas: Manual o Automático.

En Modo manual se controlara mediante un número de 0 a 3 que le indicara en que posición se deberá colocar.

En modo automático se activara y moverá cíclicamente parándolo mediante el pulsador de paro.

Esquema de montaje del panel de Circuito y del panel frontal.: *circuito12.vlogic*



12. Posicionamiento de un cilindro neumático.

Realizar un circuito para controlar el cilindro neumático en forma manual de tal manera que este se mueva de una posición a otra con un tiempo determinado $T=0,8$ seg. cíclicamente. Realizar la aplicación mediante diagrama de flujo.

Variables:



posición (variable de tipo integer que varia de 0 a 3 y que indica la posición en la que se debe situara el cilindro.

MAUT (variable de tipo booleano que indica el modo de funcionamiento del cilindro Manual o Automático, en nuestro caso deberá ponerse a valor “1” -manual-)

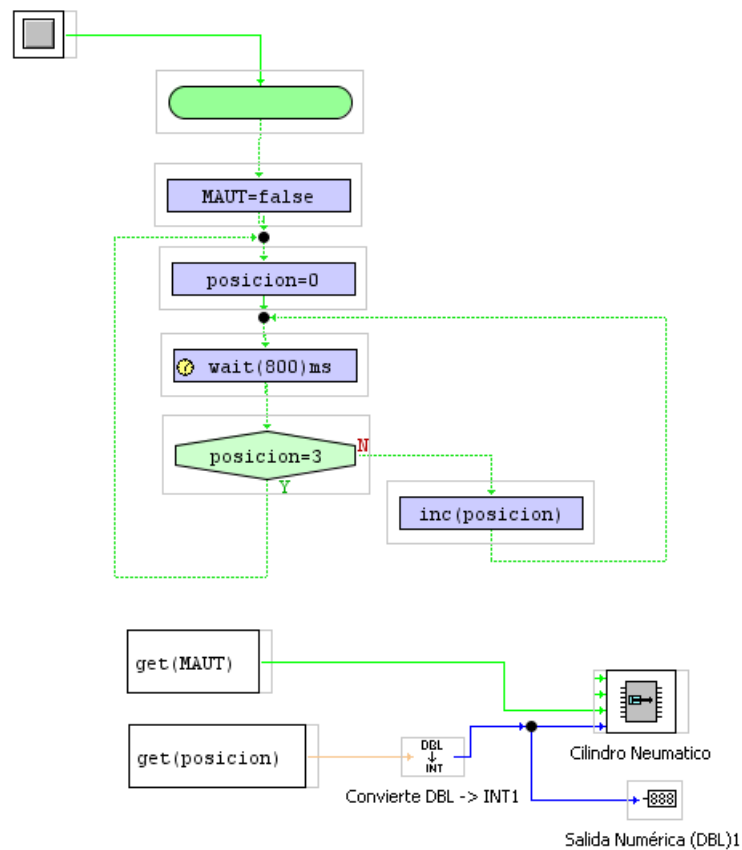
Recomendaciones: La variable “posición” debemos dársela al objeto “Cilindro” de tipo “integer” y como en los diagramas de flujo solo se puede definir de tipo “double tenemos que auxiliarnos de un bloque de librería que convierte una

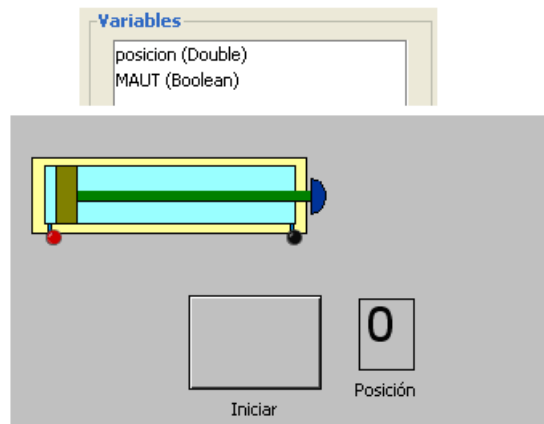
dbl-> int  -> 

Se sugiere también visualizar el estado de la variable “posición” haciendo uso de un

a caja de visualización de texto  -> 

Solución: *circuito13.vlogic*

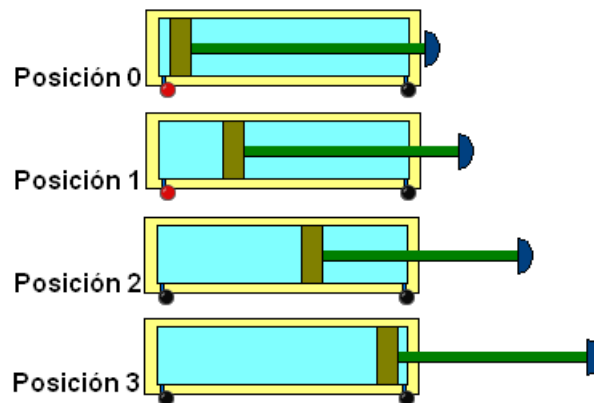




13. Mando temporizado y cíclico de un cilindro neumático

Realizar un circuito que control del cilindro neumático en el modo manual de tal manera que se cumpla la siguiente secuencia de forma cíclica

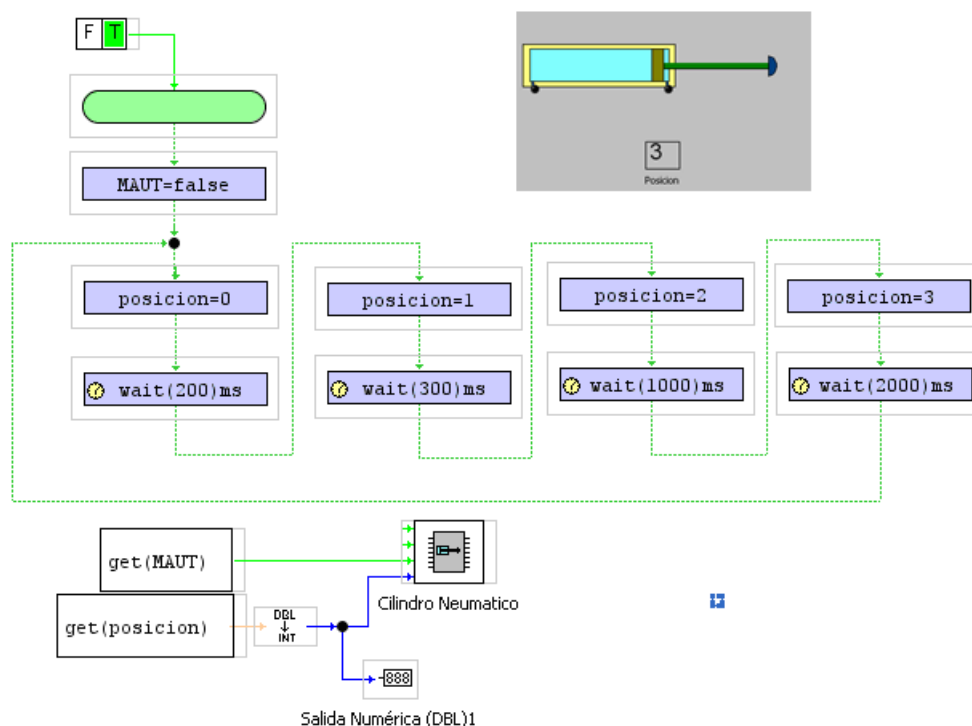
Posición	Tiempo
0	0,2 seg.
1	0,3 seg.
2	1 seg.
3	2 seg.



Variables: “**posición**” (variable de tipo double que indica la posición en la que queremos colocar el cilindro)

Parámetros: **Tp0, Tp1, Tp2, Tp3** (los tiempos que se indican en la tabla y que son los que permanece el cilindro quieto en cada posición) estos no hace falta definirlos como variables ya que son los que se asignan directamente en el bloque de temporización.

Solución: *circuito14.vlogic*



ELEMENTOS CODIFICADORES Y DECODIFICADORES DIGITALES

En Este apartado vamos a estudiar aquellos elementos que permiten codificar y decodificar de código binario a decimal y viceversa. También se hablará de los decodificadores utilizados para activar los displays de 7 segmentos.

Trabajaremos especialmente con las siguientes librerías de funciones digitales.



Conocimientos previos para poder realizar estas practicas:

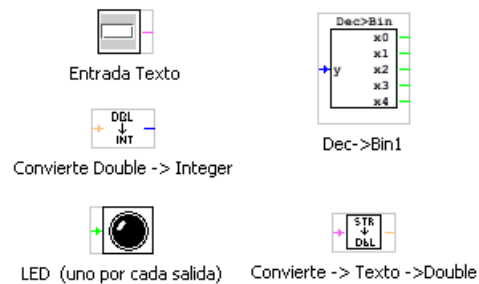
Codificación de números decimales a binarios.

Decodificacion de números binarios a decimales

Código Binario puro y Código Decimal codificado en Binario (BCD)

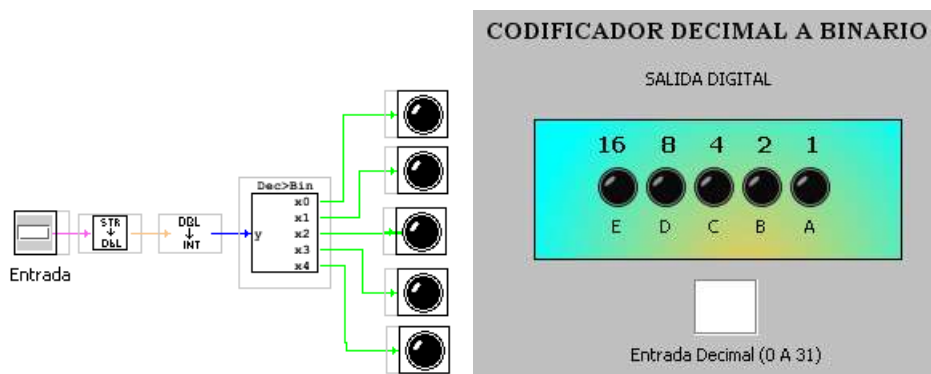
14. Codificador Decimal a Binario

Realizar un montaje en el que dado un número decimal se convierta en binario puro. Se trata de que colocando un numero decimal que varíe de 0 a 31 podamos ver su equivalente en binario. Para ello debemos utilizar el bloque de función



Téngase en cuenta que el numero de entrada que hay que suministrar al codificador debe ser de tipo Integer (entero) por lo que se hace necesario convertir el numero de entrada (decimal) de tipo double a integer mediante el bloque de función

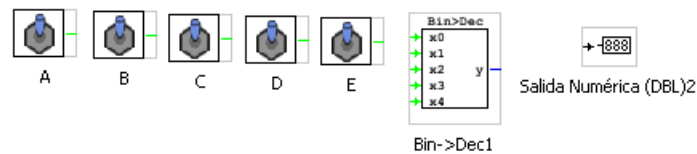
Solución: *circuito15.vlogic*



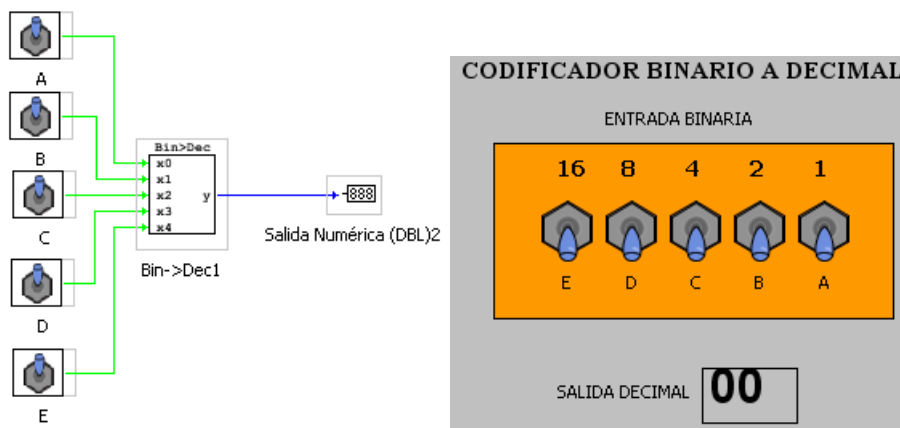
15. Codificador Binario a Decimal

Convertir un número binario de 5 dígitos (A,B,C,D,E) en su equivalente en decimal. Se trata de realizar la operación inversa a la anterior.

Librerías a utilizar:

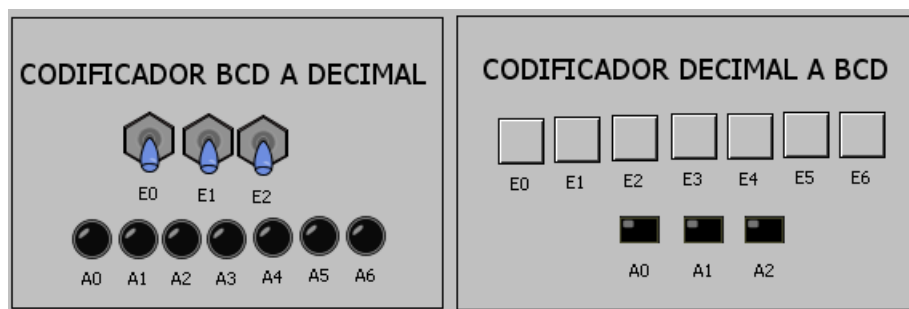
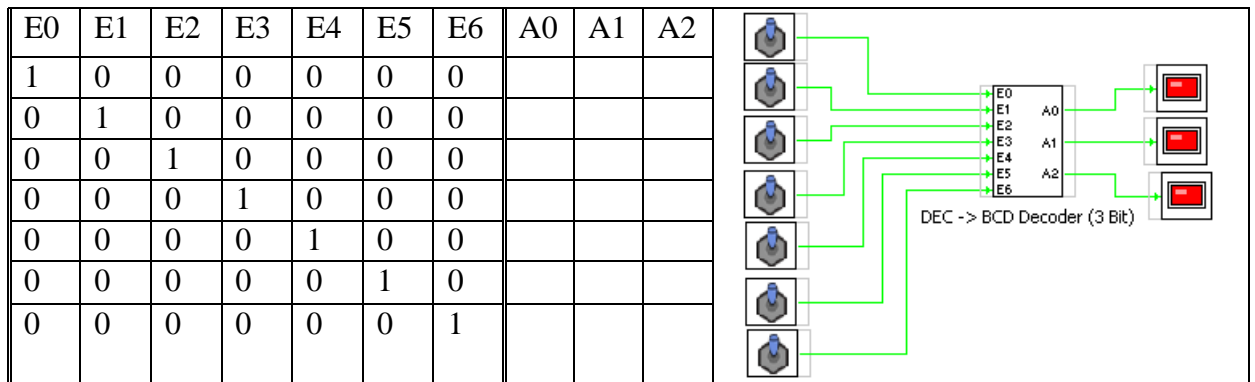
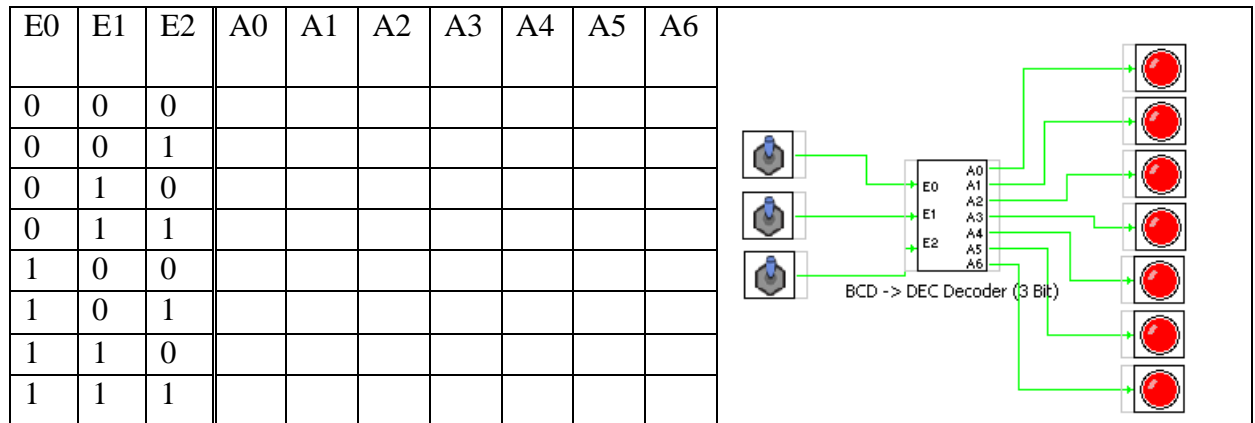


Solución: *circuito16.vlogic*



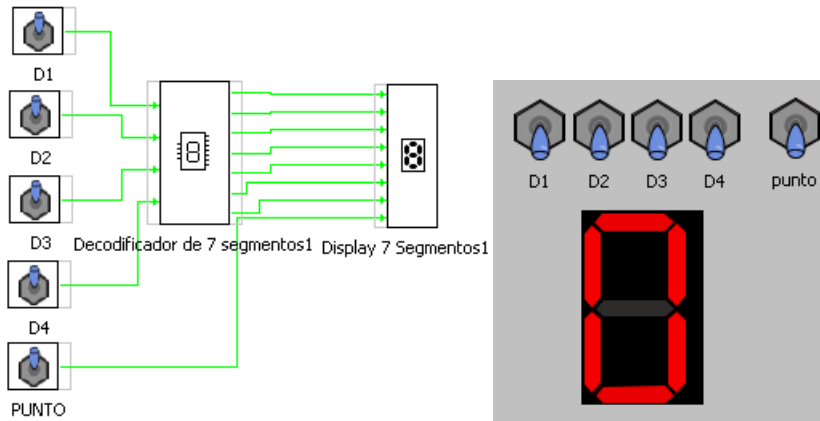
16. Codificadores de BCD a Decimal y Decimal a BCD de 3 bits.

Con este ejemplo se trata de comprender la codificación que se realiza en Binario y Decimal con códigos de 3 bits. Se trata de montar el circuito de la figura y rellenar la tabla de verdad. *circuito17.vlogic*



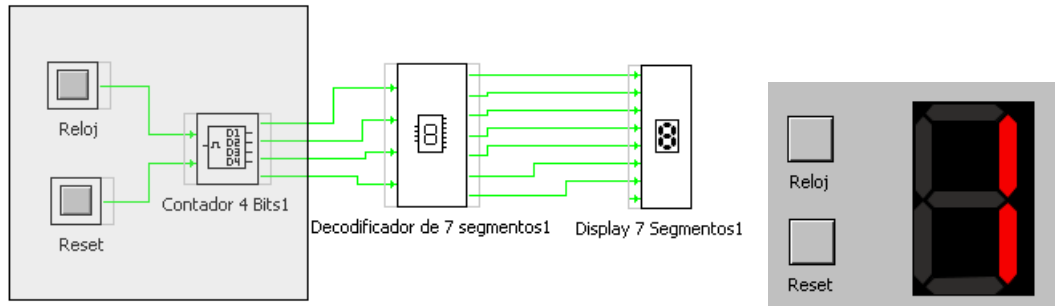
17. Gobierno de un display de 7 segmentos.

Realizar el montaje de la figura y comprobar el funcionamiento de un display de 7 segmentos. *circuito18.vlogic*



18. Contador de 4 bits con display

Tomando como base el esquema anterior colocar en lugar de los interruptores D1 a D4 un contador de 4 bits y estudiar el comportamiento del circuito. *circuito19.vlogic*

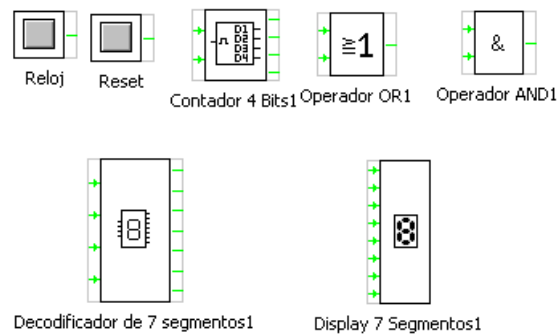


Observar que ahora los códigos de entrada al decodificador los genera un contador. Este contador cuenta desde 0 hasta 15 es decir sus salidas van desde 0000 a 1111 lo cual hace que en el display a partir de 9 aparezcan los caracteres correspondientes al código hexadecimal A,B,C,D,E,F

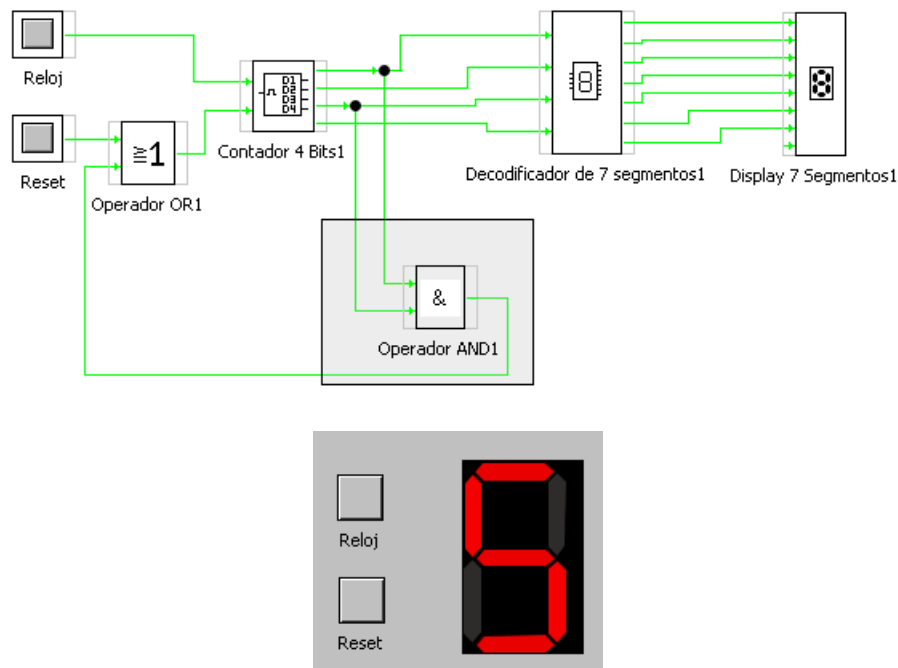
19. Contador decimal

Partiendo del esquema anterior realizar un montaje que cuente desde 0 hasta 9 y que luego vuelva a empezar en cero (0,1,2,3,4,5,6,7,8,9). Se sugiere utilizar un operador AND que recoja el código del 10 (1010) y la salida se lleve a la entrada reset del contador.

Elementos a utilizar en el montaje:



Solución: *circuito20.vlogic*



MULTIPLEXORES Y DEMULTIPLEXORES

En estas practicas se trata de realizar pruebas con los elementos Multiplexores y

demultiplexores digital. Que se muestran a continuación .

Estos elementos pertenecen a los llamados circuitos combinacionales y su utilización esta muy extendida en los automatismos.

20. Multiplexor y demultiplexor,

Realizar el montaje con el multiplexor y el demultiplexor y comprobar como funcionan.

MULTIPLEXOR:

Es importante observar como en función del código de entrada (S0,S1,S2) seleccionada así podemos dirigir el contenido de una de las 8 entradas (X0 , X1... X7) a la salida Y o Y negada

DEMULTIPLEXOR

En el caso del demultiplexor debe observarse que al revés de cómo lo hace el multiplexor en su caso se trata de dirigir una sola entrada a la salida seleccionada mediante el código de selección (S0,S1,S2)

Solución: *circuito21.vlogic*

