



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

*Práctica 1: SUBVIS EL CONVERTOR DE BITS A  
SIETE SEGMENTOS.*

Presenta:

Mendoza Meza Manuel Everardo

Profesor:

Ramirez Huitron Erick

Grupo:

3MV5

Periodo:

2022/2



29/04/2022

## Objetivo

El objetivo de la siguiente práctica consta de la implementación, ocupación y realización de los diferentes elementos con los que cuenta el programa LabVIEW; el cual, como ambiente de programación gráfica, es de mucha utilidad el conocimiento de cada uno de sus elementos con el fin de implementarlos para realizar algo más grande; por lo que, se pretende hacer uso de ciertos elementos que hagan el desarrollo completo de la siguiente práctica. A su vez, se pretende retomar conceptos y conocimientos de materias previamente cursadas, las cuales se retoman para la realización de esta práctica

## Cálculos

**Convertidor (4-bits)**

Basándonos en la siguiente tabla de verdad para el convertidor:

Decimal	Entradas (A,B,C,D)	Salidas (W,X,Y,Z)
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Teniendo en cuenta la tabla ocupamos el siguiente de circuitos lógicos para realizar Mapas de Karnaugh de forma que se relacionen las entradas con las salidas y pueda ser realizado en LabVIEW.

**W**

A\B	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$W = A + BC + BD$

**X**

A\B	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	x	x	x	x
10	0	1	x	x

$X = BE\bar{D} + AD$

**Y**

A\B	00	01	11	10
00	0	0	1	1
01	0	0	1	0
11	x	x	x	x
10	1	0	x	x

$Y = A\bar{D} + DE + B\bar{C}A$

**Z**

A\B	00	01	11	10
00	0	1	1	0
01	10	0	0	1
11	x	x	x	x
10	1	0	x	x

$Z = A\bar{D} + \bar{A}BD + B\bar{C}D$

**Convertidor DEBA a display 7 segmentos**

Realizaremos una tabla de verdad para la obtención de las propiedades de Karnaugh para el display de 7 segmentos, nos permite visualizar numéricamente de esta forma:

Entradas (DEBA)	Salidas (h,a,b,c,d,e,g)
0000	00111111
0001	00001110
0010	01011011
0011	01001111
0100	01100110
0101	01101101
0110	01111101
0111	01101111
1000	01101111
1001	01100111

Donde  $h = 0$

**a**

D\B	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$a = \bar{A}B + B\bar{C} + BE + D$

**b**

D\B	00	01	11	10
00	1	0	0	0
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$b = AB + \bar{A}B + \bar{C}D + \bar{E}$

**c**

D\B	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$c = A\bar{B} + \bar{A}E$

**d**

D\B	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$d = A\bar{B} + \bar{A}E + B$

**e**

D\B	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$e = A\bar{B} + \bar{A}E + B$

**f**

D\B	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	x

$f = A\bar{B} + \bar{A}E + B$

Imagen 1: Cálculo para obtener Arquitectura SUBVI y convertidor de 4 bits a display 7 segmentos respectivamente

## Desarrollo

## SUBVI Principal (1)

Para el desarrollo de esta SUBVI se propone el uso de un array vacío, en el que podemos ingresar elementos booleanos, esto con el fin de tener nuestro comportamiento de 7 bits, el cual, genera una señal capaz de convertirse en elementos de display de 7 segmentos que posteriormente se hablará de él. A su vez se implementa el uso de la herramienta conocida como Cluster, el cual es un elemento que puede almacenar en su interior y contener diferentes tipos de datos, esto es, booleanos, numéricos, entre otros. Esto nos ayuda con la observación de un display como se muestra en la imagen 2.

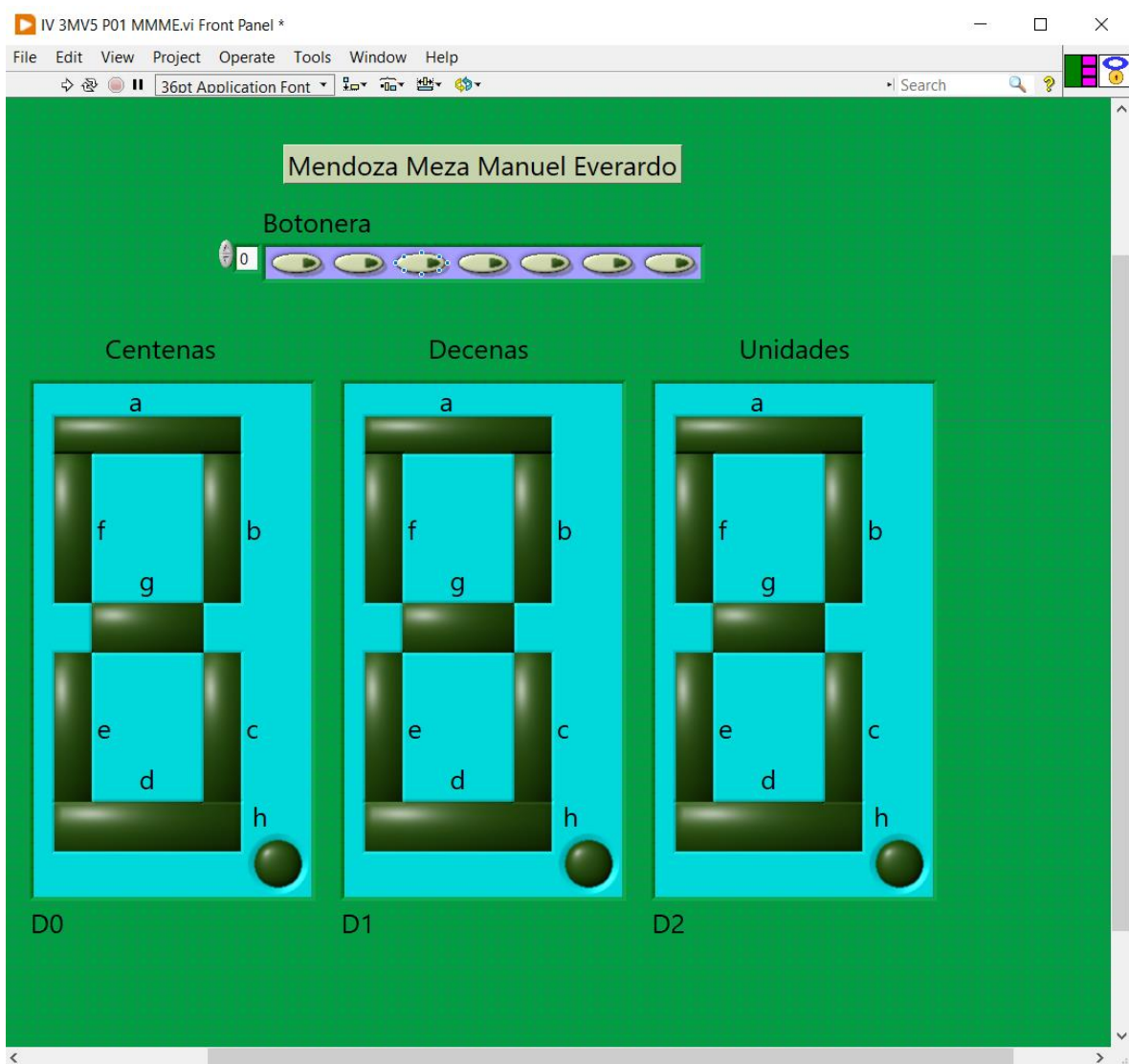


Imagen 2: Panel frontal

Esta SUBVI podemos notar la presencia del Cluster, el cual contiene elementos de tipo Booleanos representados como Leds rectangulares y uno circular para la representación de punto decimal, exactamente igual que los displays físicos.

Botonera, muestra un arreglo de elementos booleanos, que procederán a realizar la numeración correspondiente al siguiente arreglo:

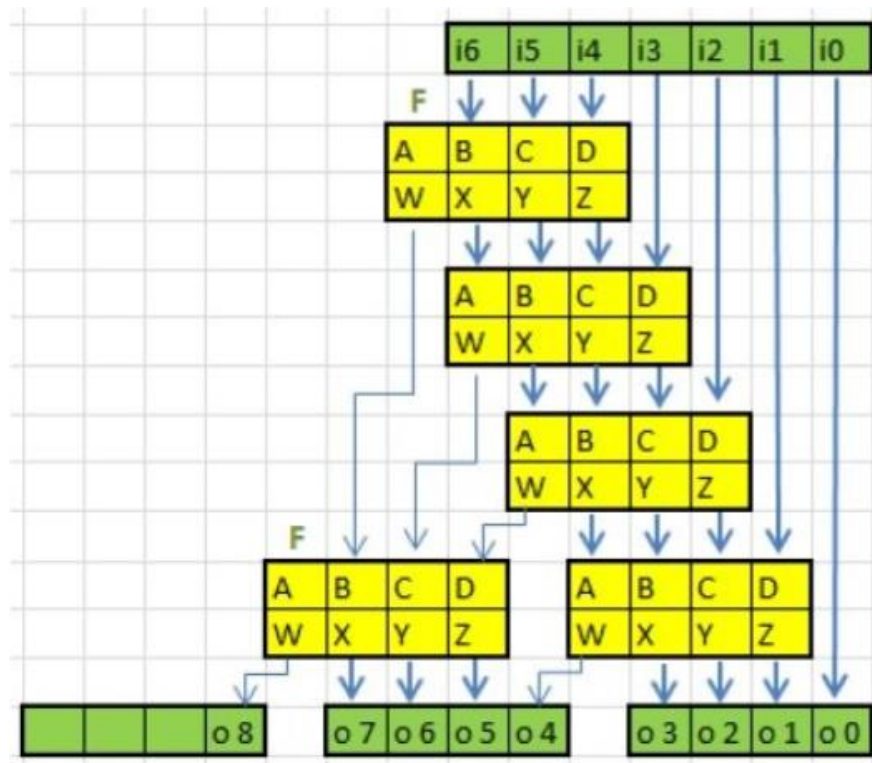


Imagen 3: Arquitectura de SUBVI

Como podemos observar en la imagen 3, el comportamiento reflejado es el mismo que se muestra en la SUBVI botonera, sin embargo, podemos observar que necesitamos realizar un arreglo de 4 entradas y 4 salidas, el cual fue realizado por medio de cálculos, como se puede observar en la imagen 1 (lado izquierdo). Una vez realizado el cálculo necesario procedemos con la obtención del programa para esta conversión y a su vez aplicarlo para la SUBVI Principal.

## SUBVI Interno

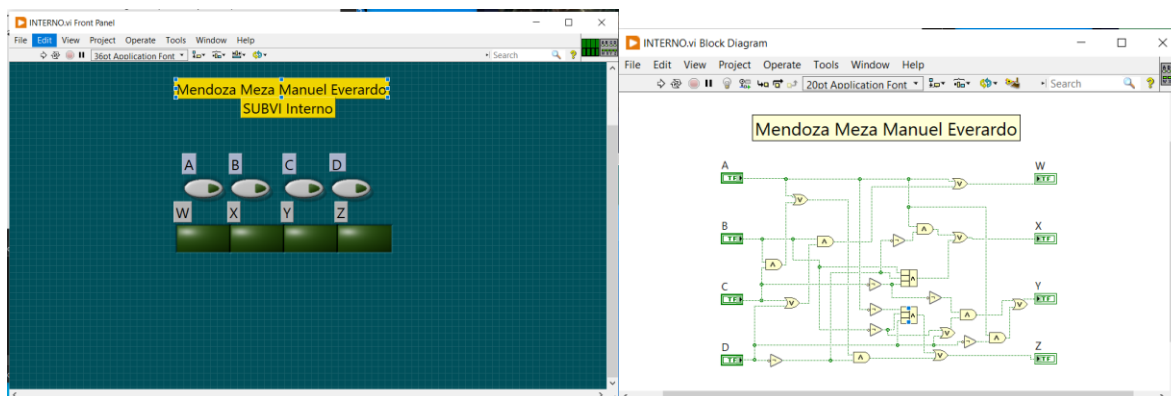


Imagen 4: SUBVI Interno

El desarrollo del SUBVI Interno corresponde a lo mencionado en la botonera dónde necesitamos crear un SUBVI que nos proporcione la tabla mostrada en la imagen 1 (izquierda) dónde se hacen los pasos necesarios para obtener las salidas “WXYZ” requeridas, esto con la finalidad de implementarlo al SUBVI Principal. Al determinar nuestras funciones, procedemos con la implementación en código, esto es, ocupar funciones booleanas que están predefinidas en LabVIEW para generar las funciones que obtuvimos gracias a los cálculos.

## SUBVI PRINCIPAL (2)

Una vez realizada la SUBVI lo podemos colocar en el SUBVI Principal con el objetivo de obtener la imagen 3, esto es para la aplicación de la estructura, intencionalmente obtenemos el siguiente resultado en la SUBVI Principal:

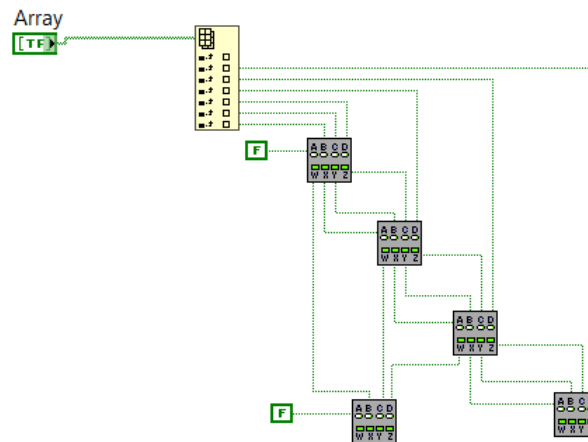


Imagen 5: SUBVI Principal conectado a SUBVI Interno y mismo conectado a clusters

Ahora, hasta este punto podríamos mencionar que el comportamiento deseado es el correcto y que funcionaría sin problemas, sin embargo, necesitamos analizar el resultado deseado, es decir, buscamos representar un display por medio de un cluster, que esta guiado por una botonera (array de booleanos) de tal forma que, al mover uno de los switch contenidos en botonera, este genere la secuencia observada en la imagen 3.

Es importante mencionar que, la estructura observada en la imagen 3 tiene un orden de LSB a MSB de derecha a izquierda, por lo que, si yo presiono el switch i0 corresponde al valor de “a” dentro del display, siguiendo el ordenamiento de LSB y MSB (Véase imagen 1 lado derecho) es por ello que, retomando la imagen 3, hay demasiadas ocupaciones del SUBVI interno para reflejar el comportamiento del display, por lo que necesitamos emparejar el SUBVI Interno en conjunto con un ordenamiento de display de 7 segmentos. Es por ello que, se propone el siguiente SUBVI.

## SUBVI DCBA a Display 7 segmentos

Como se menciona en el SUBVI Principal, buscamos que el comportamiento de la imagen 3 sea reflejado en el display de 7 segmentos, por lo que es necesario hacer la conversión de 4 bits al display, esto es realizado con la imagen 1 (lado derecho) dónde comprobamos que, para obtener el arreglo de 4 bits y que nos muestre la numeración deseada, es necesario cumplir con las funciones obtenidas, de esta forma:

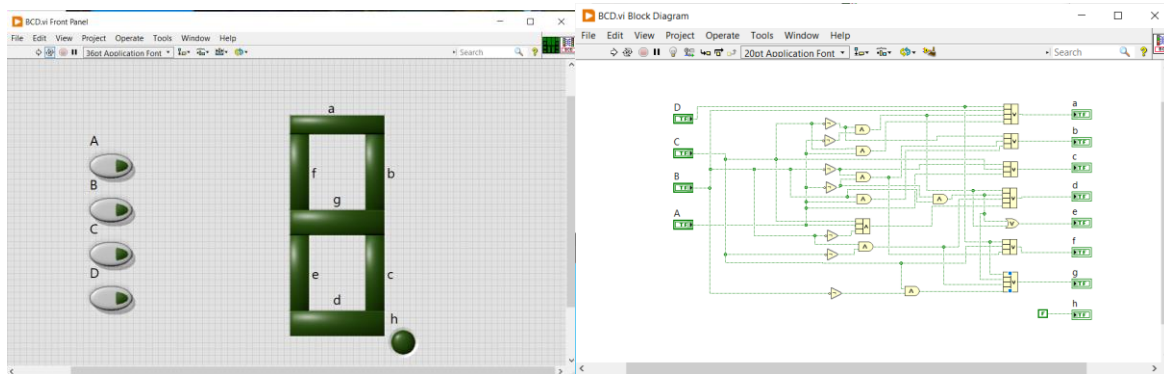


Imagen 6: SUBVI BCD

De esta forma, podemos observar con la imagen 6, el desarrollo de lo obtenido con lo calculado en la imagen 1, mostrando una representación de 4 booleanos “ABCD” que llevan a la obtención de todos los números seleccionados (0-9).

### SUBVI PRINCIPAL (3)

Una vez desarrollada la SUBVI BCD procedemos con la unión de esta SUBVI a la Principal como es requerido para la obtención de lo observado en la imagen 3 (arquitectura)



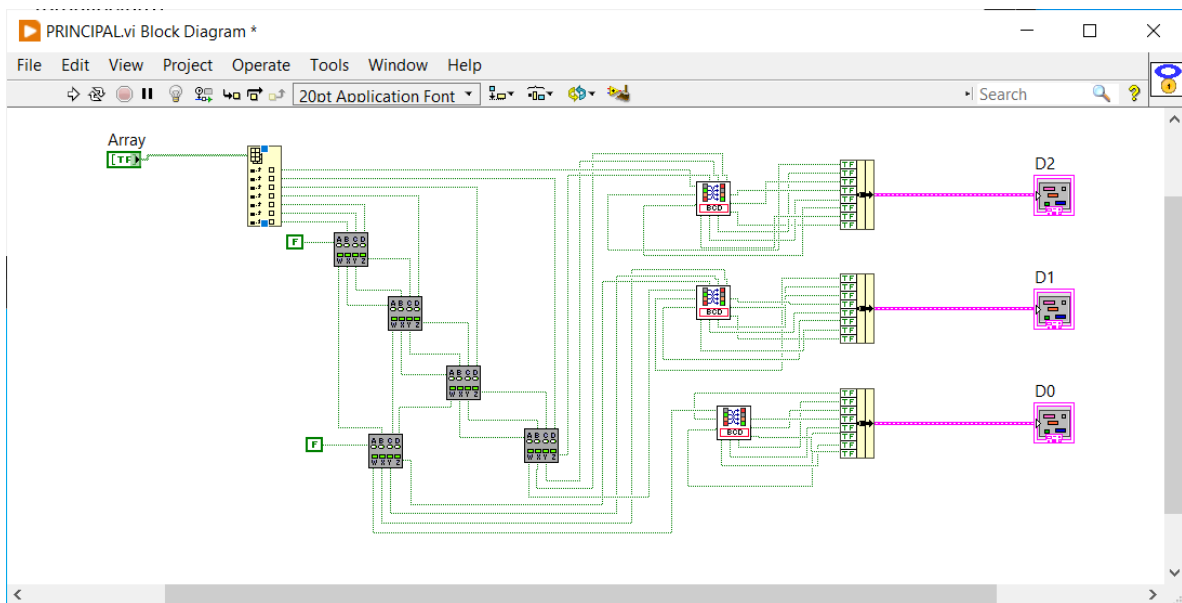


Imagen 7: SUBVI Principal

Este complemento, proporciona lo observado en la arquitectura, ya que, se nos pide como entrega que se realicen al menos 3 displays, D0-D2 como se muestra en la siguiente imagen, también es necesario obtener los elementos individuales que están contenidos en el array de la botonera, es por ello que se ocupa de un elemento "Index Array".

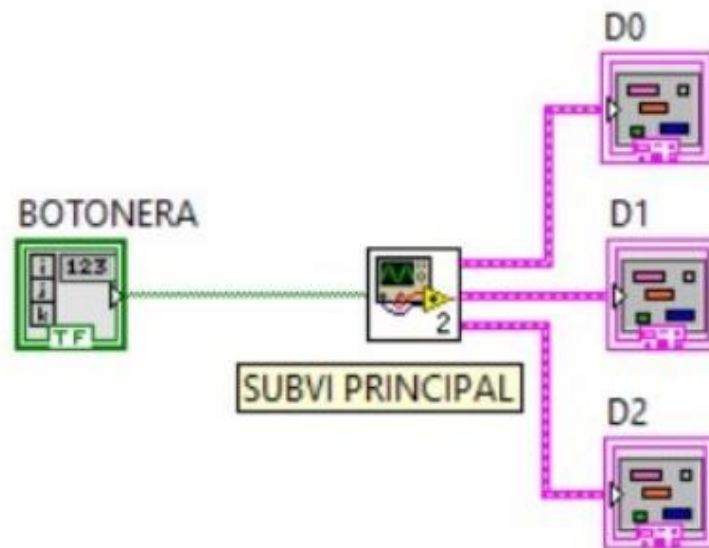


Imagen 8: Esquema de revisión

Es por ello que, requerimos juntar 3 SUBVI BCD para poder realizar el SUBVI Principal (imagen 7), posteriormente las salidas de a-h se deben de relacionar con los clusters dónde se encuentran los displays es por ello que necesitamos de la

función Bundle, la cual permite juntar las señales necesarias y llevarlas a la salida de cada cluster, de esta forma, obtenemos el resultado siguiente completo de la SUBVI Principal

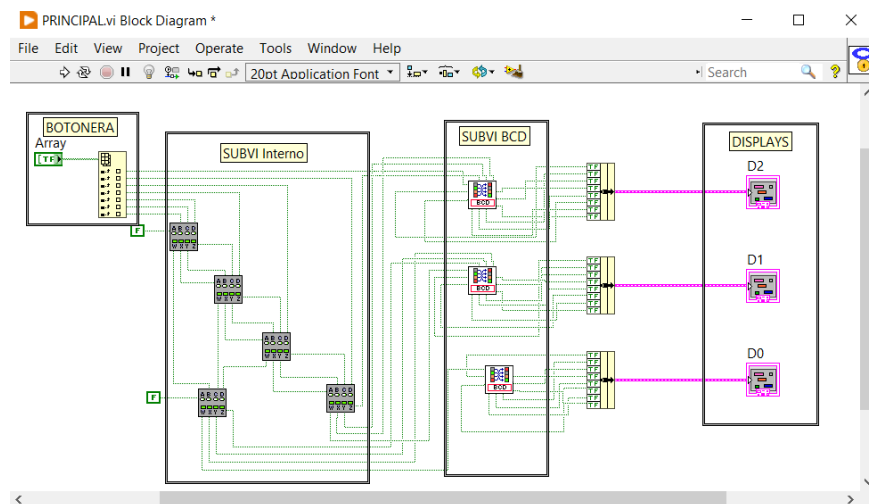


Imagen 8: SUBVI Principal completo

### Conclusiones

El desarrollo de este proyecto nos proporciona una visión amplia para la ocupación de los diferentes elementos que son posibles de localizar y ocupar en el programa LabVIEW, teniendo en cuenta características como descripciones, cambios de logos, implementación de entradas y salidas para el uso del propio SUBVI, entre otras características.

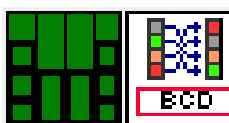


Imagen 9: Selección de entradas y salidas, ícono modificado para SUBVI BCD



Imagen 10: Selección de entradas y salidas, ícono modificado para SUBVI Principal



Imagen 11: Selección de entradas y salidas, ícono modificado para SUBVI Interno



De esta forma, con lo realizado y obtenido a lo largo de esta práctica, podemos observar y analizar los siguientes puntos que, ahora más que nunca, reflejan el conocimiento adquirido con el curso de Instrumentación Virtual.

1. SUBVI
2. Uso de SUBVI en otra SUBVI
3. Descripción del SUBVI
4. Modificación de íconos
5. Selección de elementos de entrada y salida
6. Elementos Booleanos
7. Elementos Cluster
8. Elementos tipo Array
9. Constante booleana
10. Elemento Bundle