



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA
DEPARTAMENTO DE CIRCUITOS Y MEDIDAS

LABORATORIO DE SISTEMAS DE CONTROL

Introducción
al uso de los
Controladores Lógicos Programables (PLC).

Práctica # 1:

- ⌘ Conceptos Básicos
- ⌘ Introducción de Programas
- ⌘ Lógica Binaria “And” y “Nand”
- ⌘ Lógica Binaria “Or” y “Nor”
- ⌘ Operaciones Binarias con “And” y “Or”

INTRODUCCION DE PROGRAMAS:

Una vez que el equipo esta apropiadamente conectado se procede a la introducción de un programa para lo cual se deben tomar en cuenta ciertas indicaciones:

- 1) Observar en el panel del AG la luz verde indicadora "5V" la cual informa que el procesador y el simulador están energizados.
- 2) Asegurarse de que el switch (stop-run) se encuentre en la posición "stop", esto se visualizará a través de una luz indicadora roja.
- 3) Observar en la pantalla del PG la palabra ORDEN la cual indica que el programador esta listo para recibir un comando a través del teclado del AG,

1) Pasos requeridos para la introducción del programa

Ahora vamos a seguir paso por paso como las teclas presionadas afectan el display del programador:

- I) Presionar la tecla INPUT localizada en la parte inferior del tablero. El display cambia la lectura



- II) Presione la tecla previa y note el cambio presentado.



Las tres letras subrayadas en la parte inferior izquierda indican que el tablero esta ahora en modo "shift" y se activan las funciones segundas impresas en blanco encima de las teclas.

- III) Ahora presione la tecla marcada con "PB" la cual indica "PROGRAM BLOCK" o bloque de programa.



- IV) Para completar el comando se debe identificar con un numero el PB; el 101 U solo tiene un bloque de programa, por lo cual siempre se usará "PB1".



V) Para entrar el bloque de programa completo en la memoria se debe presionar la tecla ENTER.



VI) Si un programa existe anteriormente será borrado al presionar de nuevo la tecla ENTER.





Ahora el programador esta listo para aceptar la primera proposición del usuario de un programa.

CORRECCIONES


La corrección se puede realizar ya sea en la función de Entrada o de Salida y tiene 6 formas básicas.

- 1) Borrado de instrucciones:
 - a) Seleccionar con las teclas de posicionamiento o búsqueda la instrucción a borrar.
 - b) Presionar la tecla "XI" con la cual queda borrada la instrucción.
 - c) En la pantalla se visualizará la siguiente instrucción.


- 2) Inclusión de instrucciones:
 - a) Seleccionar con las teclas de búsqueda la instrucción delante de la que se debe incluir.
 - b) Escribir la instrucción que se desea incluir.
 - c) Presionar la tecla  incluir instrucción
 - d) En pantalla se observará la instrucción delante de la que se incluyó.

- 3) Sobre escritura de instrucciones:
 - a) Seleccionar con la tecla de búsqueda la instrucción que debe ser sobrescrita.
 - b) Introducir la instrucción.
 - c) Presionar la tecla  (sobre escribir instrucción)
 - d) En pantalla se observará la siguiente instrucción.

4) Corrección usando la tecla cancelar:

Cuando se presiona una tecla que es apropiada pero indeseada (por ejemplo: 4 en lugar de 5) esta puede ser corregida presionando la tecla  (cancelar), esta borra la proposición en pantalla lo que permite volver a la proposición de manera correcta, esta tecla se puede usar en caso de aparecer en pantalla el mensaje de error 50.

5) Corrección usando la tecla interrumpir:

La tecla interrumpir  no se usa mucho para corrección de errores, sino para regresar al estado básico. Al presionar por primera vez la tecla interrumpir el programador requiere verificación y en pantalla se observara.



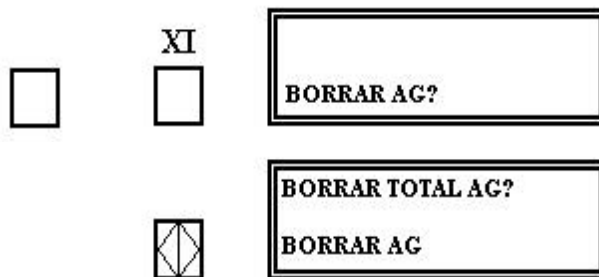
Si se presiona la tecla ENTER el programador retorna a donde estaba antes de presionar interrumpir. Si se presiona por segunda vez la tecla interrumpir el programador retorna su estado básico y se observa en pantalla:



6) Reseteando el AG 101U:

Cuando se necesite borrar completamente un bloque de programa existente en la memoria e introducir uno nuevo se debe resetear presionando la tecla <XI> (borrar)

- a) Colocar el swicht del programador en STOP.
- b) Presionar dos veces la tecla <INTERRUMPIR> para regresar al estado básico.
- c) Presionar las siguientes teclas:



- d) Presionar la tecla <ENTER> por segunda vez.



El programa en memoria ha sido borrado y el programador esta listo para un nuevo comando.

AJUSTE DEL ESTADO DE SERVICIO DEL AG:

1) Parada (STOP):

Método para parar un programa que este en ejecución.

- a) Verificar que el swicht (STOP-RUN) se encuentre en RUN.
- b) Seleccionar la función AG-STOP, mediante las siguientes teclas:

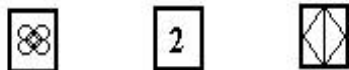


- c) Verificar que se enciende la luz roja indicadora STOP, aunque el swicht este en la posición RUN.

2) Arranque (START):

Para regresar al modo RUN se deben seguir los siguientes pasos.




- a) Seleccionar la función AG-START mediante las siguientes teclas:



- b) Verificar que se enciende la luz indicadora verde RUN.

SELECCION DE SALIDA


1) Salida del AG.

- a) Iniciar la salida del AG presionando la tecla  (SALIDA).
- b) Seleccionar el número de módulo  indica que se selecciona el módulo número 1.
- c) Realizar la ejecución de la orden mediante la tecla , se visualiza la primera instrucción del bloque de programa PB 1.

2) Salida del PG:

- a) Iniciar la salida de la siguiente manera:



Si por medio de la tecla  (INTERRUMPIR), se interrumpió la entrada o salida, se puede continuar la elaboración del módulo por medio de las instrucciones anteriormente descritas.

COMPROBACION DE UN PROGRAMA:

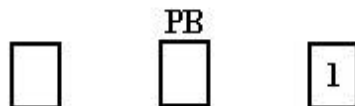
Señalización del estado de serial en dependencia del programa:

Condiciones previas :

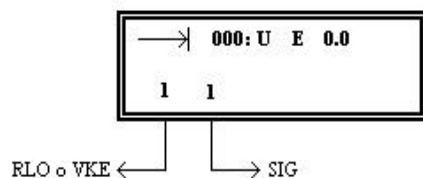
- a) Estado de servicio del AG : RUN.
- b) Iniciar la señalización del estado en dependencia del programa con la tecla STATUS



- c) Especificar la clase y número de módulo:



- d) Efectuar la orden mediante la tecla  (ENTER), se observará la primera instrucción en el módulo, y se puede ver la serial STATUS de la entrada particular o la salida especificada en la proposición, en la parte inferior izquierda de la pantalla:



RLO o VKE: resultado de la concatenación.

SIG: estado de la señal.

- e) Con la tecla  (INTERRUPCION), se puede volver al estado básico.

PRINCIPALES OPERACIONES REALIZADAS MEDIANTE EL USO DEL PLC:

Usando el lenguaje de programación STEP5 se pueden introducir las diferentes proposiciones que conforman un programa que cumple una función específica mediante un listado de instrucciones.

Las principales operaciones del PLC son las siguientes :

- 1) Lógica binaria AND/OR.
- 2) Set y Reset.
- 3) Contadores.
- 4) Temporizadores.

LOGICA BINARIA "AND":

Cada instrucción en el bloque de programas es llamada una proposición lógica. Aquí por ejemplo una proposición lógica AND debe ser entrada de la siguiente manera:

U E 0.0

donde U significa AND
 E significa Entrada
 0.0 es el número de la entrada.

Esto corresponde a la traducción de el Alemán, que es el idioma utilizado por el lenguaje STEP5.

EJEMPLO #1:

Una cierta salida será activada si y solo si dos entradas especificadas están ambas en "ON".

Solución:

a) Diagrama en escalera:



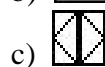
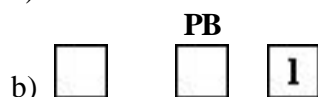
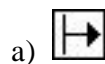
b) Lista de instrucciones:

U	E	0.0
U	E	0.1
=	A	1.0

La última línea de la lista de instrucciones corresponde a la función de asignación en la cual se asigna el valor del RLO al SIG.

La letra A corresponde a la palabra QUIT (salida) .

Pasos:



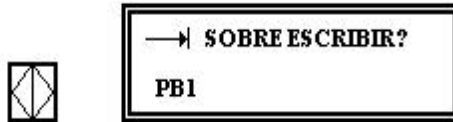
- d) Introducir el programa.
 e) Transferir a la memoria.

- 1) Asegúrese de que el swicht STOP-RUN del AG esta en la posición STOP.
- 2) Presionar la tecla < ENTER > una vez y observar el display:



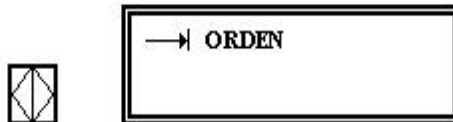
Como una precaución el programador preguntará para verificar antes de efectuar ciertos comandos críticos.

3) Si en verdad se desea transferir el bloque de programa, presione la tecla <ENTER > otra vez. Si existe otro programa en PB I el programador responderá con otra pregunta:



Esto causará que el viejo PB1 sea sobre escrito. Si el PB1 no existe, la pregunta sobre escribir no aparecerá.

4) Ahora presione < ENTER > una tercera vez y observe el display:



El programador ha completado la transferencia y está listo para recibir nuevas instrucciones.

5) Una vez transferido el programa al AG se procede a correr el programa a través de las teclas:



con lo cual se coloca el AG en la posición RUN.

6) Para probar el programa se varía el estado de las diferentes entradas (0.0 y 0.1) cuando cualquiera de ellas esté en OFF la salida se debe apagar y cuando ambas estén en ON la salida se debe encender, si esto sucede indica que el programa está funcionando correctamente.

7) El programa debe cumplir con la siguiente tabla de la verdad.

E 0.0	E 0.1	A 1.0
0	0	0
0	1	0
1	0	0
1	1	1

LOGICA BINARIA "NAND"

Se estudiará como introducir una proposición lógicamente "NAND" en un bloque de programa. Para introducir una proposición NAND es necesario escribirla de la siguiente manera:

UN E 0.0

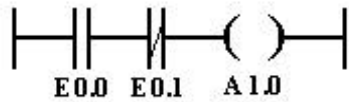
donde UN significa NAND.

EJEMPLO #2:

La salida es controlada por dos entradas. Para que la salida este en ON, la primera entrada debe estar en ON y la segunda entrada debe estar en OFF.

Solución:

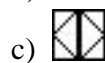
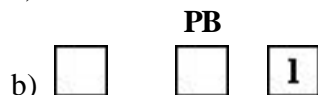
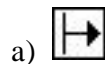
a) Diagrama en escalera:



b) Lista de instrucciones:

U	E	0.0
UN	E	0.1
=	A	1.0

Pasos:



d) Introducir el programa.

e) Transferir a la memoria. (Asegúrese de que el swicht STOP-RUN del AG esta en la posición STOP).

Si el programa está bien configurado, debe cumplir con la siguiente tabla de la verdad.

E 0.0	E 0.1	A 1.0
0	0	0
0	1	0
1	0	1
1	1	0

LOGICA BINARIA "OR":

Las reglas que gobiernan las operaciones lógicas OR son las mismas de las AND, facilitando de esta manera su comprensión. Para introducir una proposición lógica OR se debe hacer de la siguiente manera:

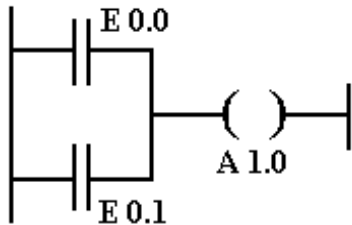
O.	E	0.0
----	---	-----

EJEMPLO #3:

Una salida es controlada por 2 entradas, se quiere que cuando cualquiera, la primera o a segunda entrada, esté en ON la salida debe ser ON.

Solución:

a) Diagrama en escalera:

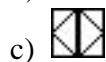
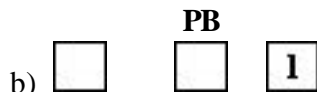
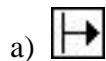


b) Listado de instrucciones:

```

O.    E    0.0
O.    E    0.1
=     A    1.0
  
```

Pasos:



d) Introducir el programa.

e) Transferir a la memoria. (Asegúrese de que el swicht STOP-RUN del AG esta en la posición STOP).

Si el programa está bien configurado, debe cumplir con la siguiente tabla de la verdad.

E 0.0	E 0.1	A 1.0
0	0	0
0	1	1
1	0	1
1	1	1

LOGICA BINARIA "NOR":

Para introducir al AG una proposición lógica "NOR" se debe hacer de la siguiente manera

```

O.N   E    0.0
  
```

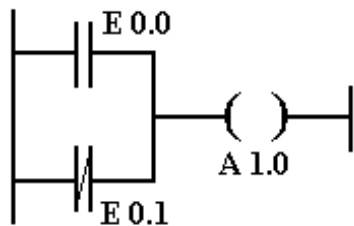
Esta proposición dirige el PLC a buscar la entrada 0.0 y si la señal STATUS es cero, coloca el RLO a "1".

EJEMPLO #4:

Una salida es controlada por 2 entradas; la salida debe ser ON cuando una de las entradas este en ON o la otra entrada este en OFF.

Solución:

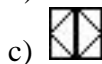
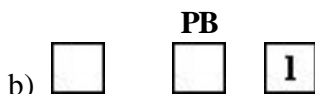
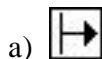
a) Diagrama en escalera:



b) Listado de instrucciones:

O.	E	0.0
O.N	E	0.1
=	A	1.0

Pasos:



d) Introducir el programa.

e) Transferir a la memoria. (Asegúrese de que el swicht STOP-RUN del AG esta en la posición STOP).

Si el programa está bien configurado, debe cumplir con la siguiente tabla de la verdad.

E 0.0	E 0.1	A 1.0
0	0	1
0	1	0
1	0	1
1	1	1

OPERACION BINARIA “AND ANTES QUE OR”:

Ahora se estudiará un tipo de programación mas sofisticada que combina lógica AND y lógica OR. La primera técnica que se aplicará es la llamada "AND antes que OR", en la cual el AG combina grupos de operadores lógicos AND a través de una operación lógica OR.

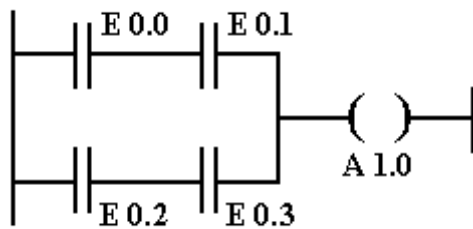
Las operaciones AND son procesadas primero, antes que la operación OR sea ejecutada.

EJEMPLO #5:

Una salida debe ser activada cuando la entrada 0.0 y la entrada 0.1 o la entrada 0.2 y la entrada 0.3 están en ON.

Solución:

a) Diagrama en escalera:



b) Listado de instrucciones:

U	E	0.0
U	E	0.1
O		
U	E	0.2
U	E	0.3
=	A	1.0

OPERACION BINARIA “OR ANTES QUE AND”:

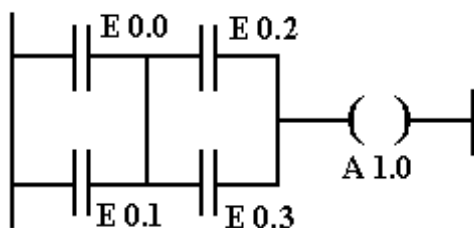
En esta configuración la técnica de programación cubre las aplicaciones en las cuales el AG combina grupos de operaciones lógicas OR a través de una operación lógica AND. En este caso las operaciones OR son procesadas primero, antes que la operación AND sea ejecutada.

EJEMPLO #6:

La salida debe ser activada cuando la corriente fluya a través de las entradas 0.0 ó 0.1 y las entradas 0.2 ó 0.3.

solución:

a) Diagrama en escalera:



b) Listado de instrucciones:

O.	E	0.0
O.	E	0.1
U(
O.	E	0.2
O.	E	0.3
=	A	1.0

Se debe hacer notar que en este listado se han utilizado 2 nuevas teclas como son: "U(" y la tecla ")", las cuales son usadas exclusivamente con lógica OR antes de AND. Para observar su mejor funcionamiento se recomienda hacer un seguimiento a todos los problemas realizados a través de la función STATUS y observar como varían los estados del RLO y el SIG en cada instrucción.

OPERACION “OR” Y “AND” MEZCLADAS:

En la lógica mezclada, las operaciones OR pueden tener lugar entre operaciones AND; y operaciones AND entre operaciones OR; además pueden ocurrir en niveles variantes. Esto en realidad no es mas que combinaciones de "AND antes de OR" y "OR antes que AND".

La tecla para programar operaciones lógicas mezcladas es reconocida cuando se usa "U(" y "O".

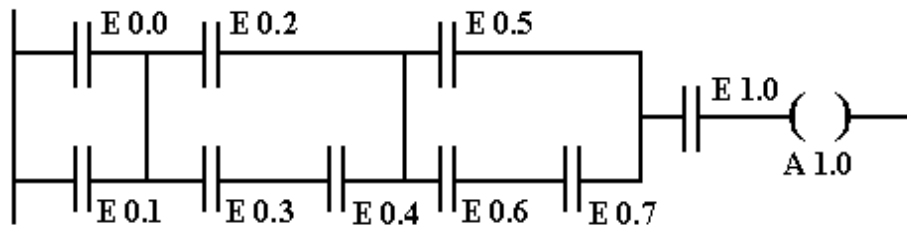
EJEMPLO #7:

Una salida puede ser controlada por, varias combinaciones de nueve entradas. La salida es activada si alguna de las siguientes condiciones ocurre:

- (las entradas 0.0 ó 0.1 están en ON) y
- (la entrada 0.2 esta en ON ó las entradas 0.3 y 0.4 están en ON) y
- (la entrada 0.5 esta en ON ó las entradas 0.6 y 0.7 están en ON y
- (la entrada 1.0 está en ON).

Solución:

a) Diagrama en escalera:



b) Listado de instrucciones:

O.	E	0.0
O.	E	0.1
U(
O.	E	0.2
O		
U	E	0.3

U	E	0.4
)		
U(
O.	E	0.5
O		
U	E	0.6
U	E	0.7
)		
U	E	1.0
=	A	1.0

Ahora se procede a entrar la lista de instrucciones con el 605U:

- 1) Hágale un Reset a el AG y entre un nuevo PB1.
- 2) Arrancando de la dirección de proposición 000, introduzca el programa entero, proposición por proposición.
- 3) Una vez introducido el programa se debe transferir el PB1 al AG
- 4) Colocar el swith del AG en el modo RUN.
- 5) Trate varias combinaciones de entradas y note que la entrada 1.0 debe siempre estar en ON para que la salida este en ON.
- 6) Es importante realizar el monitoreo del programa de lógica mezclada que se ha introducido y observar el estado del RLO y el SIG para los distintos valores de las entradas.