0-2-2019

Alcalá Villagómez Mario; Murguia Chavez Nadia Sarahi

controladores logicos programables.

Pistón con sensores

Actividad 1

Índice.

Nombre de la práctica.

Objetivo.

Materiales.

Investigación previa.

Desarrollo.

Conclusión.

Nombre de la practica:

Cilindro con sensores.

Objetivo de la practica:

El alumno debe realizar y analizar una simulación de un PLC, mediante el siguiente ejercicio.

Materiales:

|  |  |
| --- | --- |
| Caja de operaciones | Software Axel |
| Raspberry | Jumpers tipo hembra y macho |
| Caimanes |  |

Investigación previa:

Un PLC es un Controlador Lógico Programable (Programmable Logic Controller), en sí es un sistema de control. Los PLC´s son dispositivos electrónicos o computadoras digitales de tipo industrial que permiten la automatización, especialmente de procesos de la industria, debido a que controlan tiempos de ejecución y regulan secuencias de acciones.

De acuerdo con la definición de la NEMA (National Electrical Manufacturers Association) un PLC es: “Un aparato electrónico operado digitalmente, que usa una memoria programable para el almacenamiento interno de instrucciones para implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas para controlar, a través de módulos de entrada/salida digitales (ON/OFF) o analógicos (1-5 VDC, 4-20 mA, etc.), varios tipos de máquinas o procesos”.

Funcionamiento básico

* Detecta diversos tipos de señales del proceso mediante la Memoria Programable.
* Elabora y envía acciones de acuerdo al programa en la Memoria de Datos.
* Recibe configuraciones de los operadores y da reportes a los mismos
* Admite modificaciones en el programa cuando son necesarias.

Principales aplicaciones

Ingeniería y producción en empresas, principalmente en la industria, en donde se aprovechan especialmente para los siguientes casos:

* Espacio reducido.
* Procesos de producción periódicamente cambiantes.
* Procesos secuenciales.
* Maquinaria de procesos variables.
* Instalaciones de procesos complejos y amplios.
* Chequeo de programación centralizada de las partes del proceso

En concreto los PLC´s son dispositivos que permiten automatizar y son empleados principalmente para procesos industriales, con múltiples ventajas y aplicaciones variadas.

Programación tipo ladder

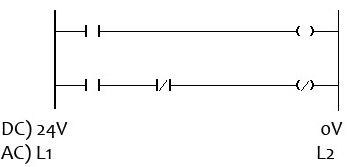
En esta nueva entrada de blog vamos a tratar el lenguaje más utilizado en la programación de autómatas; el denominado lenguaje en escalera, ladder o de contactos.

Este sistema, estandarizado en IEC 61131-3, es un lenguaje gráfico basado en los esquemas de control de conmutación clásicos siguiendo lógica matemática. La programación ladder se encuentra presente en la mayoría de autómatas o PLCs (Program Logic controller) debido a su simplicidad y amplio abanico de posibilidades.

El esquema eléctrico se representa con dos líneas verticales a derecha e izquierda representando:

* Las línea 24V y 0V en sistemas de corriente continua
* L1 y L2 en sistemas de corriente alterna.

La CPU realiza la lectura de izquierda a derecha y de arriba abajo, por lo que las instrucciones de entrada o contactos se introducen en el lado izquierdo y las de salida o bobinas en el lado derecho. Las condiciones de entrada determinan si las instrucciones de salida son activadas o no.



Al representar los contactos clásicos podemos encontrar cuatro posibilidades de condiciones de entrada:

– Contacto normalmente abierto: El contacto o entrada se encuentra en circuito abierto cuando su valor lógico es 0 y en circuito cerrado cuando su valor lógico es 1. Es el contacto más empleado en todo tipo de interruptores o selectores.

– Contacto normalmente cerrado: El contacto o entrada se encuentra en cortocircuito cuando su valor lógico es 0 y en circuito abierto cuando su valor lógico es 1. La utilidad de este tipo de equipos es la activación de los mismos si hay algún fallo en ellos lo que permite detectar fallos de alimentación, cableado…

– Flanco positivo: Se activa en el momento que el estado lógico de un contacto pasa de 0 a 1. Se suele emplear en pulsadores, donde lo importante no es el estado, sino el número de pulsaciones.

– Flanco negativo: Se activa en el momento que el estado lógico de un contacto pasa de 1 a 0. Se suele emplear igualmente en pulsadores, pero en el momento de soltar el pulsador.

Las representaciones gráficas de los elementos están normalizadas según normas NEMA. En particular los símbolos empleados para los contactos de entradas que hemos visto son los siguientes:



En cuanto a las salidas o bobinas también nos encontramos con cuatro posibilidades:

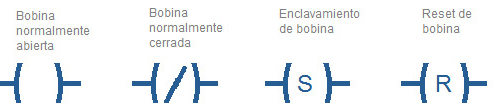
– Bobina normalmente abierta: al activarse pondrá a nivel lógico 1 la bobina.

– Bobina normalmente cerrada: al activarse pondrá a nivel lógico 0 la bobina.

– Enclavar bobina: al activarse pondrá a nivel lógico 1 la bobina y aunque las condiciones de entrada cambien, la bobina se quedará en este este estado.

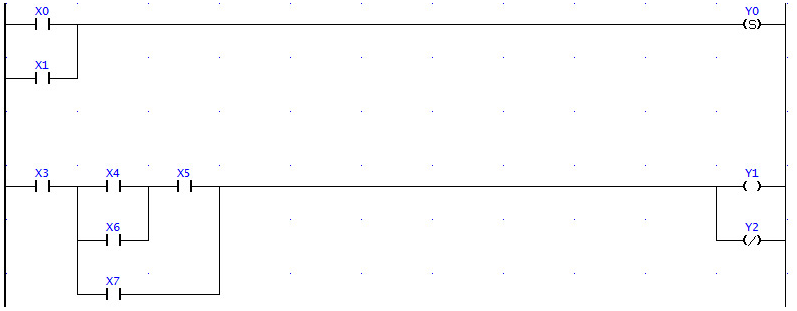
– Resetear bobina: al activarse pondrá a nivel lógico 0 la bobina y aunque las condiciones de entrada cambien, la bobina se quedará en este estado.

La representación de estas cuatro opciones es la que se puede ver a continuación:



En la parte superior de los elementos se suele indicar la entrada/salida/marca interna empleada.

Empleando estos elementos, podemos realizar combinaciones entre ellos para crear diferentes lógicas de activación/desactivación tales como elementos en serie o paralelo:



Además de estas bobinas, las condiciones de entrada pueden ejecutar funciones que nos permiten ejecutar diferentes operaciones, ya sean matemáticas (sumas, restas, multiplicaciones, divisiones…), de temporización, de contaje, etc.

Como hemos comentado este lenguaje se emplea en numerosos softwares de programación de PLCs como pueden ser el WinProLadder de Fatek o el TIA Portal de Siemens.

Desarrollo de la práctica:

Se tiene un cilindro en posición de Home (retraído) y con el sensor A activo, cuando el operador presiona el botón P el cilindro sale hasta el sensor C, y si el botón se deja de presionar regresa a Home.

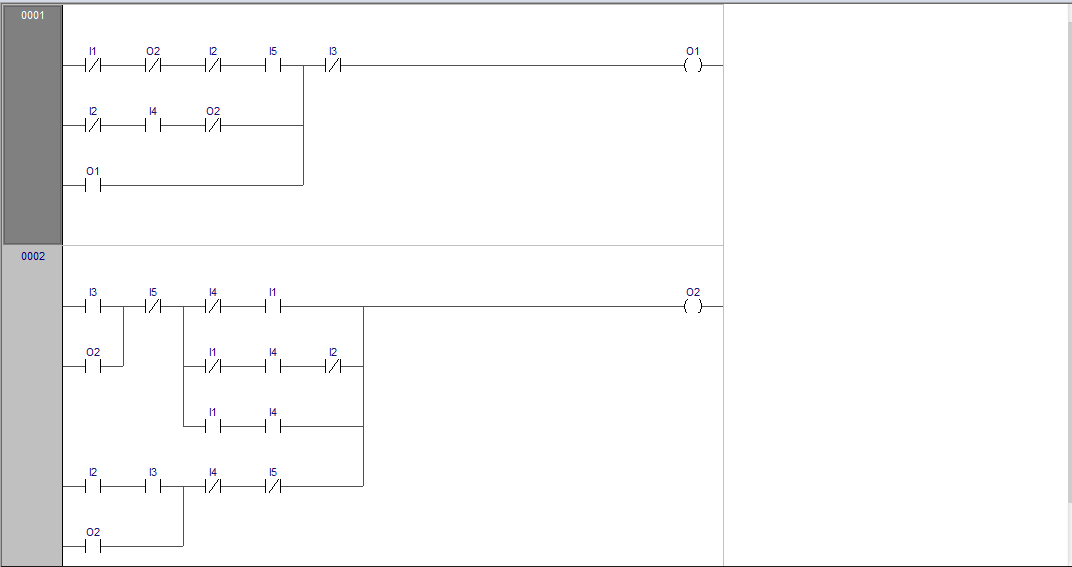
Pero si presiona el botón Q, el cilindro sale hasta B y cuando deja de presionarse se regresa a Home condiciones:

Home, cilindro hasta atrás, sensor A activo.

P y A activos = mover cilindro hasta C → P y C activos = regreso Home

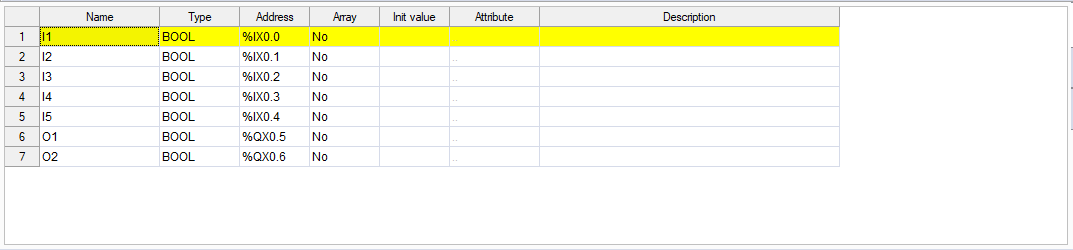
Q y A activos = mover cilindro hasta B → Q y B activos = regreso Home

Diagrama de escalera.



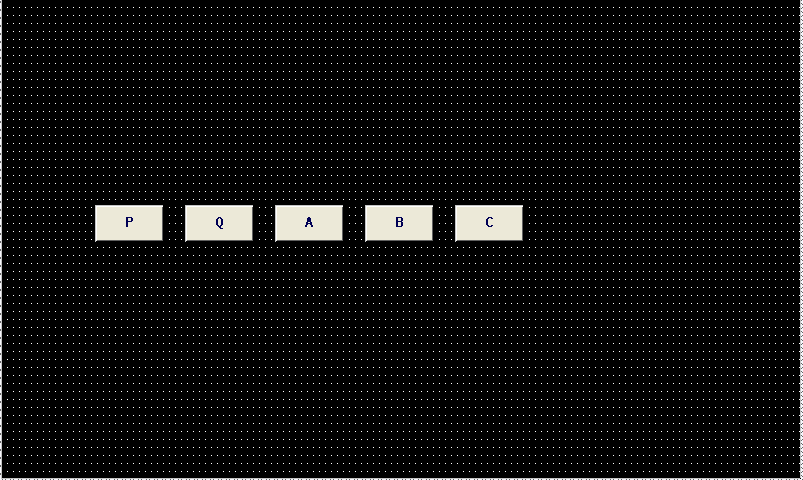
|  |  |  |
| --- | --- | --- |
| Variable | Entrada | Salida |
| P | 1 |  |
| Q | 2 |  |
| A | 3 |  |
| B | 4 |  |
| C | 5 |  |
| S |  | 1 |
| R |  | 2 |

Declaracion de variables.

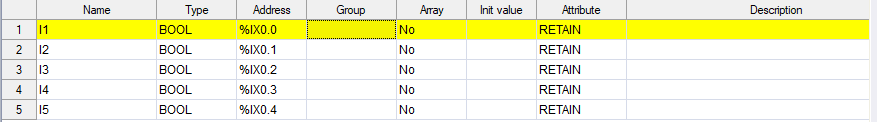


Simulación en Axel:

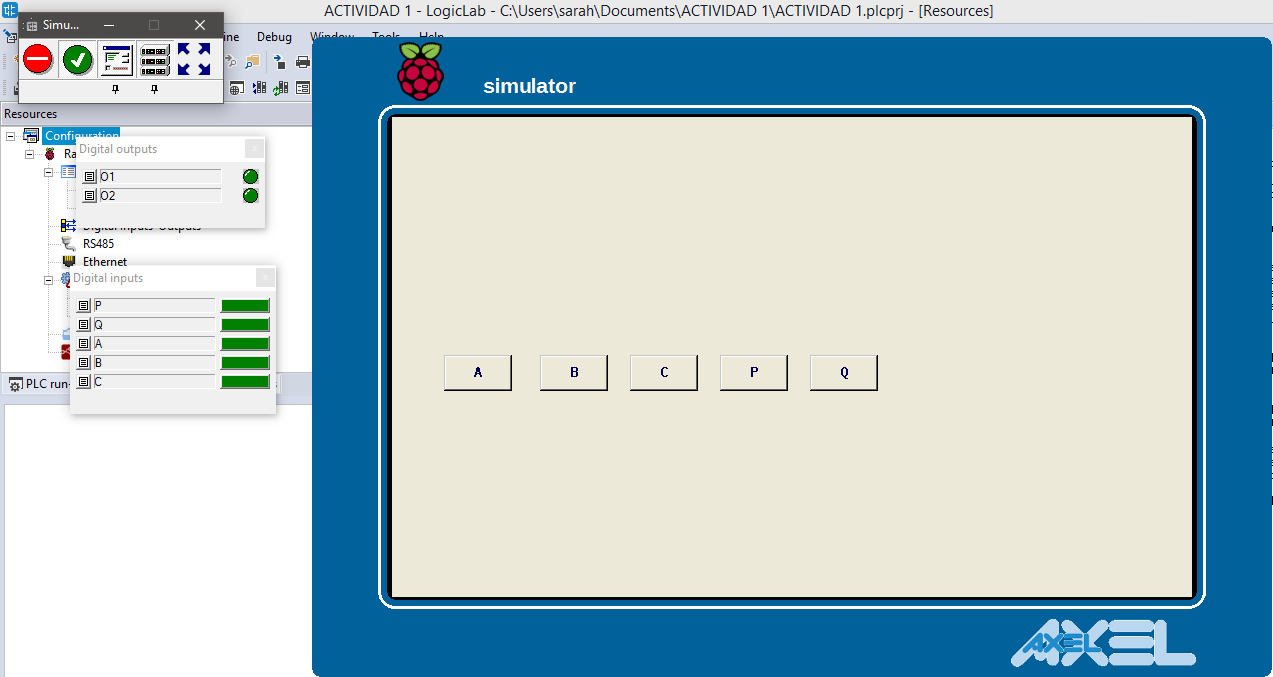
Creamos una pantalla en pageLab:



Volvemos a declarar nuestras variables



Depues tendremos la siguiente imagen para nuestra simulación:



Ya después cargaremos nuestro programa a la rapsberry para verificar que todo esto funciones correctamente.

Conclusiones

Tal como se muestra en la simulación el programa funciona correctamente sin embargo por problemas con la licencia de Axel para la raspberry no pudimos presentar las pruebas físicas de la actividad.