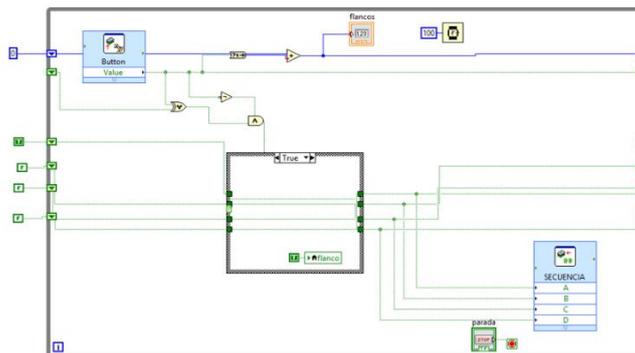
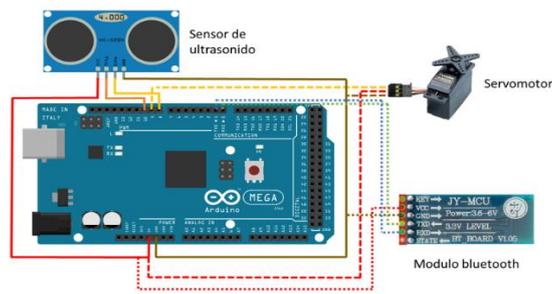
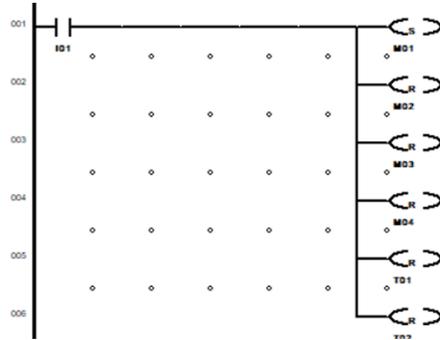


GUÍA DE LABORATORIO VIRTUAL DE MECATRÓNICA I LMEC 432



Febo Flores

Contenido

PRÁCTICA 1.....	3
Simulación de inversión de giro de motor trifásico con automatismos	3
PRÁCTICA 2.....	6
Simulación de inversión de giro motor trifásico con micro PLC	6
PRÁCTICA 3.....	10
Simulación de arranque secuencial de motores con micro PLC	10
PRÁCTICA 4.....	13
Simulación con GRAFCET de secuencias de circuitos electro-neumáticos	13
PRÁCTICA 5.....	18
Simulación de arranque estrella triángulo con micro PLC	18
PRÁCTICA 6.....	27
Simulación control motor paso a paso con autómata programable	27
PRÁCTICA 7.....	33
Aplicaciones de simulación con señales analógicas.....	33
PRÁCTICA 8.....	44
Simulación de control combinacional con circuitos integrados.....	44
PRÁCTICA 9.....	47
Simulación con circuitos integrados bi-estables	47
PRÁCTICA 10.....	51
Programas en Arduino	51
PRÁCTICA 11.....	58
Introducción Arduino (UNO) entradas-salidas digitales y conexión interfaz.....	58
PRÁCTICA 12.....	61
Control de motores paso a paso con Arduino	61
PRÁCTICA 13.....	63
Proyectos con Arduino	63
PRÁCTICA 14.....	65
Programación básica con LabVIEW	65
PRÁCTICA 15.....	66
Integración Arduino (UNO) y LabVIEW, almacenamiento de datos	66

PRÁCTICA 1

Simulación de inversión de giro de motor trifásico con automatismos

OBJETIVO: Simulación de control combinacional ejecutado con automatismos aplicado a inversión de giro de un motor trifásico jaula de ardilla.

COMPETENCIA: Aplica simulador CadeSimu o uno similar en control combinacional para inversión de giro de un motor trifásico jaula de ardilla, ensamblando el circuito con automatismos en base a contactores.

SOFTWARE DE SIMULACIÓN RECOMENDADO: CadeSimu

FUNDAMENTO TEÓRICO: Los automatismos están siempre compuestos de circuitos de control o mando (lado izquierdo de la figura) y de potencia o fuerza (lado derecho de la figura).

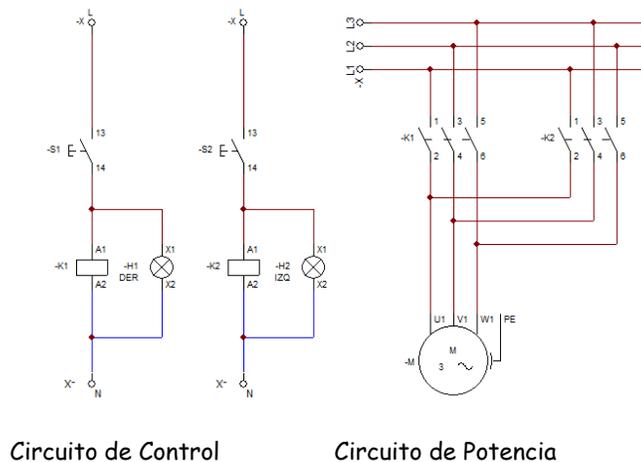


Figura 1: Diagrama de conexión inversión de sentido de rotación motor 3 ϕ

En el circuito de control se aprecia que dependiendo si se pulse S1 o S2, se accionará la bobina K1 o K2 que tienen accionamiento en sus contactos auxiliares normalmente abiertos mostrados en el circuito de potencia.

Mientras se mantiene activado K1;

- L1 conecta U1,
- L2 conecta V1,
- L3 conecta W1.

Mientras que si se activa K2;

- L1 conecta U1,
- L2 conecta W1,
- L3 conecta V1.

Es como si con las manos estuviera cambiando de: L2 – V1 y L3 – W1 a: L2 – W1 y L3 – V1

Precisamente para evitar riesgos de hacer cortos circuitos, anexar a los contactores, contactos auxiliares NC (Normally Closed) o normalmente cerrados y hacer la siguiente modificación:

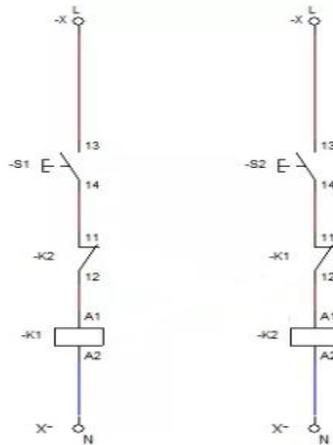


Figura 2: Circuito de control con enclavamiento eléctrico

Los contactos auxiliares normalmente cerrados NC (11 – 12) cortan la alimentación a la bobina opuesta, de manera de evitar que ambas se energicen simultáneamente. Existen también relés para enclavamiento mecánico que pueden adicionarse para brindar mayor seguridad al presente circuito. Para protección del motor contra sobrecargas débiles y prolongadas suele incluirse en la instalación, un relé térmico que contiene contactos auxiliares a adicionarse al circuito de control. Al adicionarse un relé térmico deben emplearse los terminales (NC) 95 y 96 en el circuito de control, de manera que al incrementarse la temperatura, dicho contacto se abre y desenergiza a las bobinas de los contactores. Contactos 97 y 98 pueden usarse para encender una lámpara de alarma.

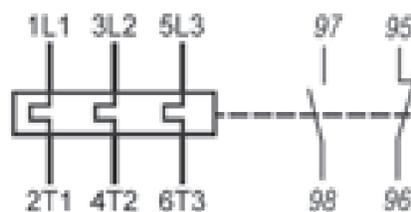
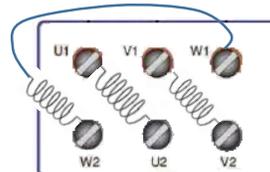


Figura 3: Relé Térmico, Extraído del catálogo de WEG Automatización-Contactores y relés de sobrecarga

Bornera de un motor de inducción jaula de ardilla trifásico:



Vista de la distribución interna de las bobinas

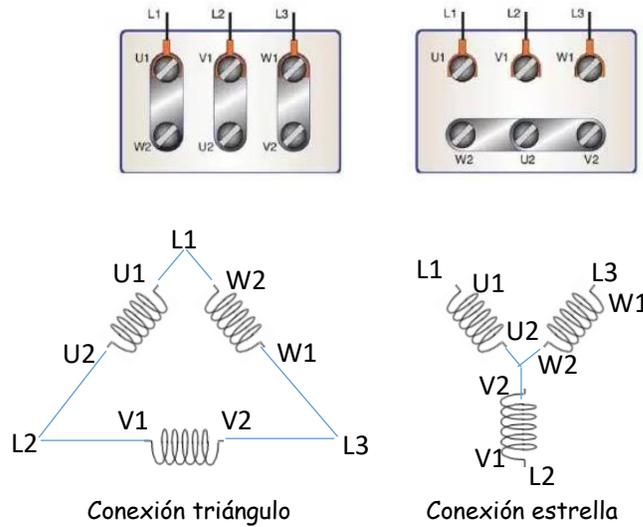


Figura 4: Bornes del motor trifásico de inducción o jaula de ardilla

ACTIVIDADES A REALIZAR:

1. Simular el sistema de la figura 1 para inversión de giro combinacional en CadeSimu, emplee contactores de 220 VAC.
2. Simular un corto circuito pulsando ambos pulsadores al mismo tiempo, entonces añada un enclave eléctrico que evite ese incidente con contactos auxiliares N.C. como se muestra en la figura 2 y simule qué sucede si se pulsan ambos pulsadores.
3. Simular el mismo circuito pero añadiendo un relé térmico como el mostrado en la figura 3.
4. Busque en la red la forma física (fotos) de los siguientes componentes de automatismos eléctricos: Pulsadores, Interruptores, finales de carrera, minicontactor, contactor, magnetotérmico, relé térmico y preséntelos con descripción propia.
5. Dibujar la representación física, es decir de cómo se vería la instalación eléctrica, si gusta, puede hacer uso de fotos de los componentes empleados.

PRÁCTICA 2

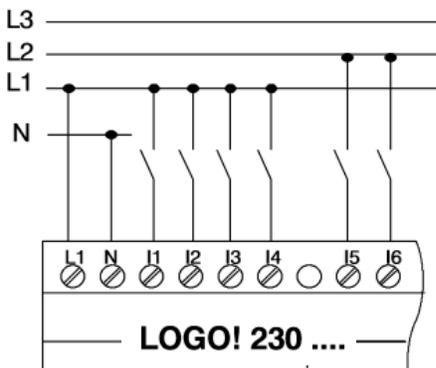
Simulación de inversión de giro motor trifásico con micro PLC

OBJETIVO: Simulación de control combinacional ejecutado con micro autómatas aplicado a inversión de giro de un motor trifásico de inducción o jaula de ardilla.

COMPETENCIA: Emplea simulador aplicado a operación de control combinacional para inversión de giro de un motor trifásico jaula de ardilla, ensamblando y programando contactores controlados por micro autómatas programables.

SOFTWARE DE SIMULACIÓN RECOMENDADO: CADeSIMU, LOGO! Soft Control, eSMS Config, EASY-SOFT.

FUNDAMENTO TEÓRICO

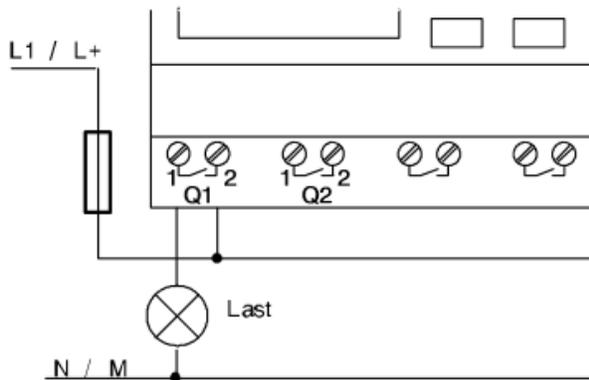


Los modelos LOGO! 230... cuentan con entradas divididas en dos grupos de cuatro entradas.

Estado operacional 0: < 40 V CA o < 30 V CC

Estado operacional 1: > 79 V CA o > 79 V CC

Figura 1: Entradas en modelo LOGO!, extraído de manual LOGO!



Los modelos con salida de relé para operación en CA o CC, se recomienda instalación de un fusible de hasta 16 A en la línea L1/L+.

La ventaja de salidas de relé es que pueden manejarse actuadores que con CA o CC y cualquier voltaje, siempre y cuando la corriente de operación no exceda la permitida por los contactos del autómatas

Figura 2: Salidas de relé, extraído de manual LOGO!

Revise el video realizado por un antiguo estudiante de la asignatura en:

<https://youtu.be/9abWngomMpk>



El modelo EASY Messenger con salida de relé para operación en CA o CC, se recomienda instalación de un fusible de hasta 16 A.

La novedad de este modelo es la posibilidad de enviar y recibir mensajes SMS. Es decir, realizar control mediante teléfono móvil con un alcance y cobertura en función del chip de la empresa de telefonía móvil.

Figura 3: Aspecto físico de micro autómatas EASY-Messenger!

EASY Messenger puede programarse tanto en sitio empleando su pantalla y teclado como con el software eSMS Config, mismo que también permite hacer simulación.

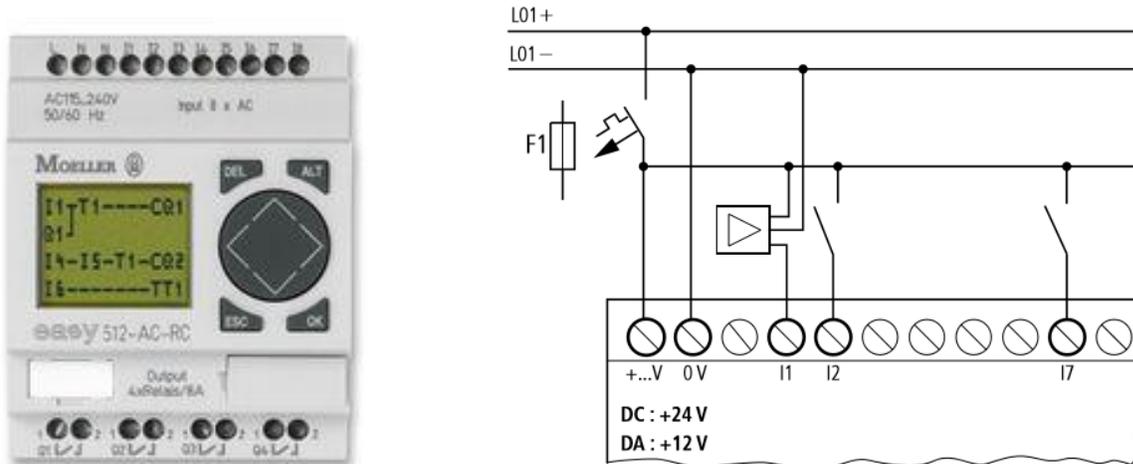


Figura 4: Micro autómatas EASY de MOLLER y sus conexiones físicas de entradas PNP

En la figura 4 se muestra el micro autómatas EASY de MOLLER y conexiones físicas en su puerto de entrada.

En su pantalla se aprecia su programación en diagrama de contactos, mismo que se puede hacer en el sitio haciendo uso de su teclado o a través de su software EASY-SOFT con una computadora personal. EASY-SOFT permite también hacer simulaciones.

Existen dos tipos de entrada de autómatas programables PNP o NPN o del inglés “sourcing” o “sinking” respectivamente.

Cuando la entrada es del tipo PNP (sourcing), las entradas son alimentadas de la línea viva o positiva, como se ve en la figura con entradas I2 e I7, mientras que en I1 está conectado un sensor de proximidad de tres hilos también configurado PNP.

Aplicación con finales de carrera: El autómata programable tiene cuatro entradas; I01, I02, L01 y L02, que son respectivamente; pulsador para rotación sentido horario, pulsador para rotación sentido anti horario, final de carrera horario y final de carrera anti horario.

Se emplearán dos salidas del autómata programables; Q01 y Q02 para mando sentido horario y sentido antihorario respectivamente.

Para el mando combinacional y a fin de asegurar operación segura se realiza una tabla de verdad para verificar todas las combinaciones posibles de las entradas y evaluar en que condiciones se obtiene operación segura.

I01	I02	L01	L02	Q01	Q02
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	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
1	1	1	1	0	0

Fig. 5: Tabla de verdad operación de motor

$$Q01 = I01 \times \overline{I02} \times \overline{L01} \times \overline{L02} + I01 \times \overline{I02} \times L01 \times \overline{L02} = I01 \times \overline{I02} \times \overline{L02}$$

$$Q02 = \overline{I01} \times I02 \times \overline{L01} \times \overline{L02} + \overline{I01} \times I02 \times \overline{L01} \times L02 = \overline{I01} \times I02 \times \overline{L01}$$

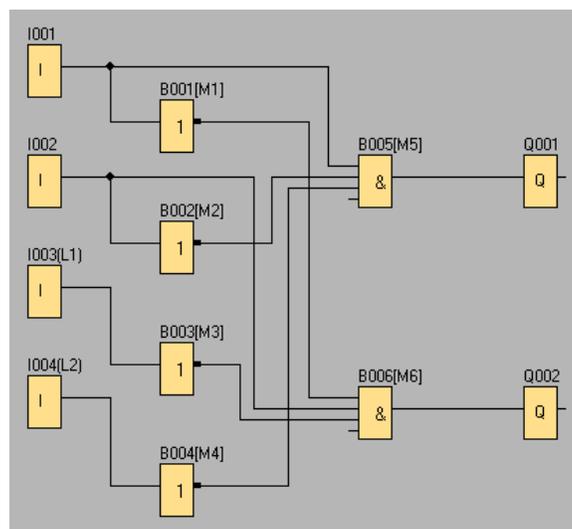


Fig. 6: Diagrama de bloques correspondiente a salida PLC Q01 (LOGO! Soft Control, eSMS Config)

Con el autómata “Messenger”, se dispone de un cable de conexión a la PC con conector RS232, del que ya no disponen las computadoras, por lo tanto se emplea un convertor RS232 a USB como se muestra en la figura 7, mismo que incluye un driver a instalar en la PC. “Messenger” se comunica con la computadora mediante MODBUS RTU, que permite hacer monitoreo con la computadora al autómata.



Figura 7: USB a RS232

ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

1. Armar en CADeSimu el mismo circuito de potencia empleado en la práctica anterior. Seguidamente, conectar los pulsadores y finales de carrera a entradas I1, I2, I3 e I4 y salidas Q1 y Q2 que alimentan K1 y K2 respectivamente en los autómata programables LOGO! Y SIMATIC S7.
2. Programar el automatismo según las ecuaciones obtenidas de la tabla de verdad de la figura 5. La figura 6 muestra la lógica para Q1. Completar el diagrama de la figura 6 con salida Q2 para cambio del sentido de rotación en CADeSimu:
 - LOGO! Soft Control (Esquema de funciones o bloques)
 - SIMATIC S7-1200, CPU1214C (Diagrama de contactos o escalera)
3. Repetir la experiencia pero en los siguientes simuladores de micro PLCs:
 - eSMS Config (Esquema de funciones o bloques).
 - EASY-SOFT (Diagrama de contactos o escalera).
4. Dibuje cómo conectaría señales todo o nada en autómatas con entrada NPN (sinking).
5. Dibuje la apariencia física de toda la instalación eléctrica, de los circuitos de control como de potencia.

PRÁCTICA 3

Simulación de arranque secuencial de motores con micro PLC

OBJETIVO: Simulación de control secuencial en función a eventos (pulsadores) aplicado en arranque de motores de inducción.

COMPETENCIA: Simula el arranque de motores de inducción trifásicos en secuencia y comandados por pulsadores, realizando el cableado y programación en un micro autómatas programable con diagrama de bloques y de contactos.

SOFTWARE DE SIMULACIÓN RECOMENDADO: CAdESIMU, LOGO! Soft Control, eSMS Config, EASY-SOFT.

FUNDAMENTO TEÓRICO: Grafo que muestra las secuencias:

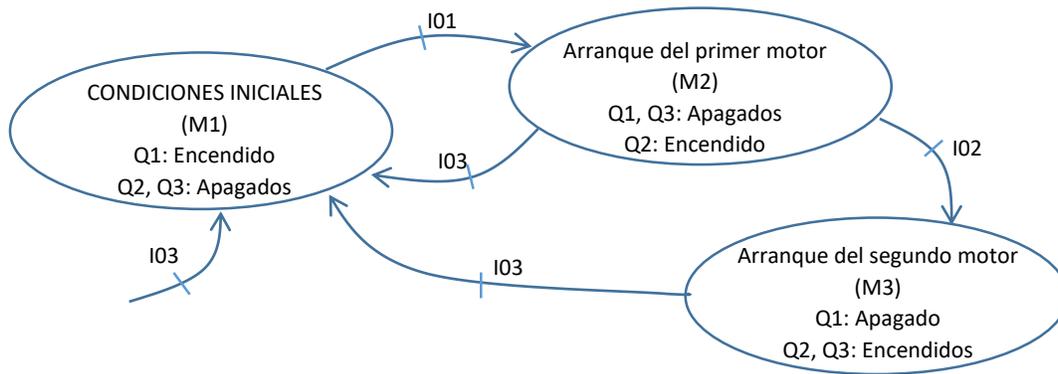


Figura 1: Parte secuencial del programa

Donde:

I01: Pulsador para arranque del primer motor

I02: Pulsador para arranque del segundo motor

I03: Pulsador para regresar a condiciones iniciales

Q1: Relé que activa a una lámpara que indica que el sistema está en condiciones iniciales

Q2: Relé que activa al primer motor

Q3: Relé que activa al segundo motor

	M1	M2	M3	Q1	Q2	Q3
1	0	0	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	0	1	1

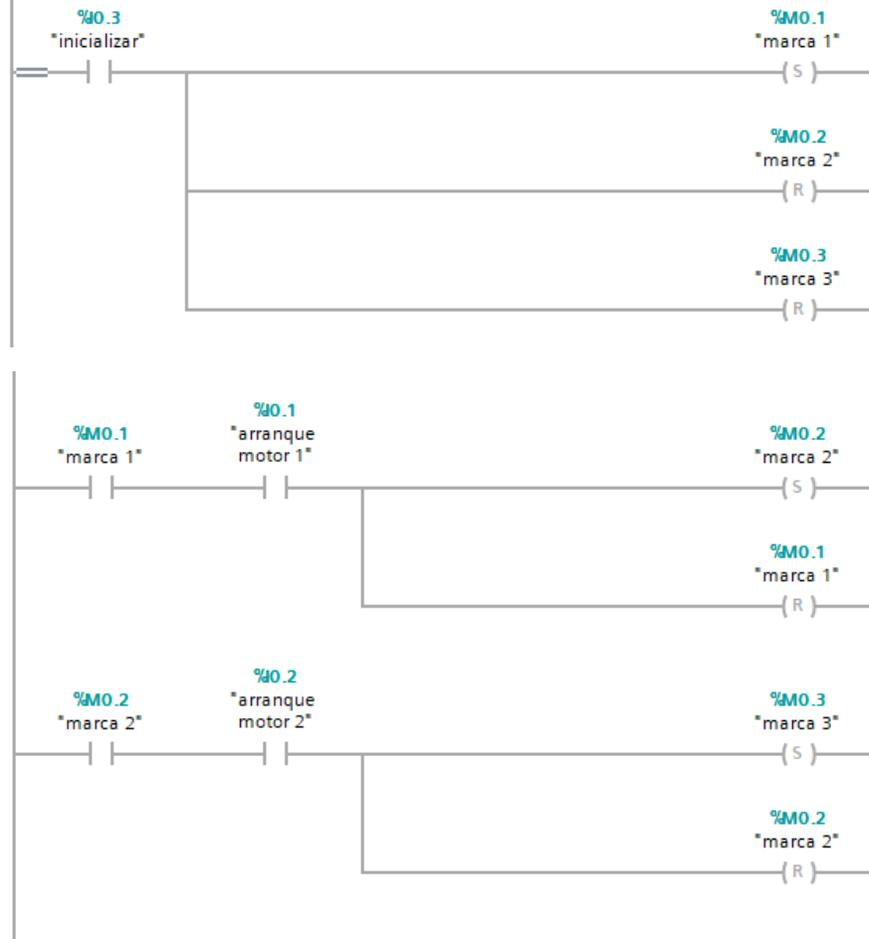
Tabla 1: Tabla de verdad que relaciona marcas con salidas, suponiendo que sólo una marca se activa por vez

Las ecuaciones para Q1, Q2 y Q3 suponiendo que sólo se activa una marca M_i por vez:

$$\begin{aligned}
 Q1 &= M1 \\
 Q2 &= M2 + M3 \\
 Q3 &= M3
 \end{aligned}$$

Segmento 1: Secuencia de marcas

Comentario



Segmento 2: Salidas físicas

Comentario

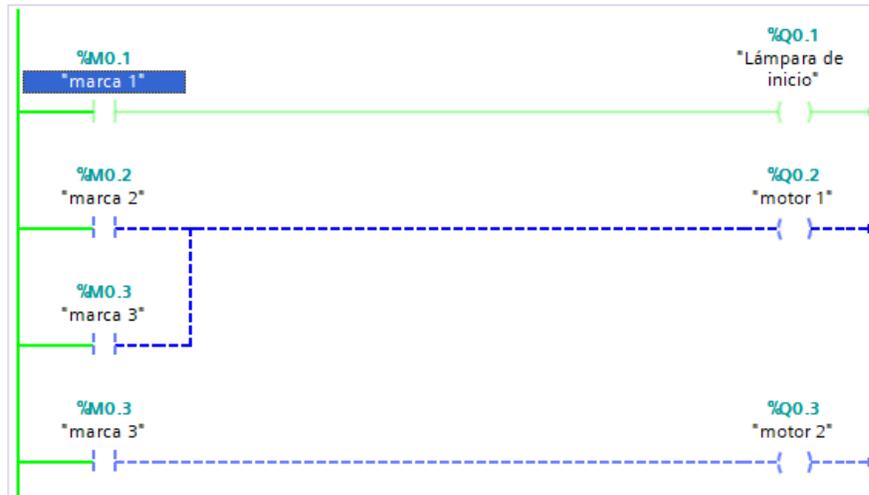


Figura 2: Programa en diagrama de contactos (escalera, "ladder" o "KOP") SIMATIC S7-1200

OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

1. Armar en CAdESimu el circuito de potencia para arranque de dos motores de inducción jaula de ardilla trifásicos, incluyendo conexiones de los pulsadores a entradas I1, I2 e I3 y salidas Q1, Q2 y Q3 que alimentan K1, K2 y la lámpara en los autómatas programables:
 - LOGO!
 - SIMATIC S7.
2. Programar el control secuencial según el grafo de estados mostrado en la figura 1 en los autómatas de CAdESimu de la pregunta anterior y simular verificando todas las transiciones de la secuencia:
 - LOGO! Soft Control (Esquema de funciones o bloques)
 - SIMATIC S7-1200, CPU1214C (Diagrama de contactos o escalera)
3. Repetir la experiencia de programación pero en los siguientes simuladores de micro PLCs:
 - eSMS Config (Esquema de funciones o bloques).
 - EASY-SOFT (Diagrama de contactos o escalera).
4. Dibuje o represente con fotos, las conexiones físicas de la práctica (como se verían realmente) que tendría que hacer en un PLC que guste.
5. Indique las modificaciones que tendría que hacer en el programa (escalera) para el grafo mostrado en la siguiente figura 3 con respecto al de la figura 1.

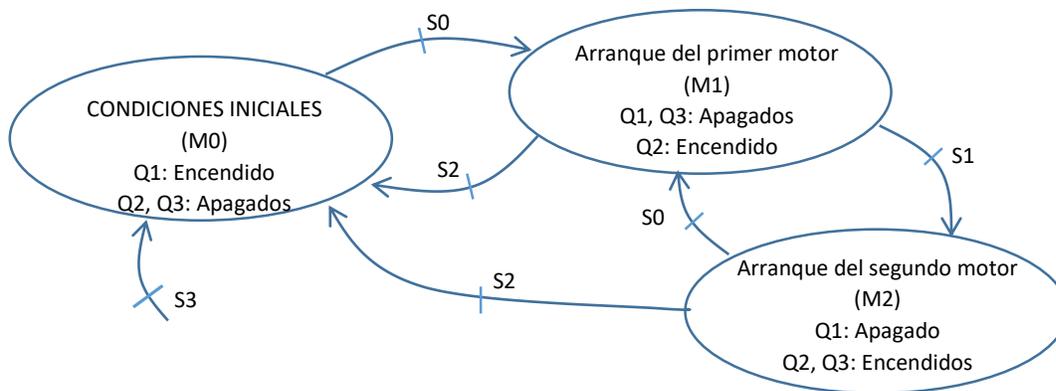


Figura 3: Grafo para representar en diagrama de contactos

PRÁCTICA 4

Simulación con GRAFCET de secuencias de circuitos electro-neumáticos

OBJETIVO: Aplicación de la metodología GRAFCET para programar secuencias en circuitos electro-neumáticos.

CAPACIDADES: Emplea GRAFCET para programar secuencias de circuitos electro-neumáticos y los simula.

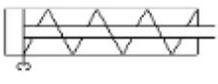
FUNDAMENTO TEÓRICO: Los circuitos electroneumáticos son muy empleados en actividades de manipulación industrial automatizada.

SOFTWARE DE SIMULACIÓN RECOMENDADO: CADeSIMU, eSMS Config, EASY-SOFT y opcionalmente LOGO! Soft Control, FLUIDSIM de Festo, xLogicsoft.

En la electroneumática los actuadores siguen siendo neumáticos, los mismos que en la neumática básica, pero las válvulas de gobierno mandadas neumáticamente son sustituidas por electroválvulas activadas con electroimanes en lugar de pilotadas con aire comprimido. Las electroválvulas son convertidores electroneumáticos que transforman una señal eléctrica en una actuación neumática. Por otra parte los sensores, fines de carrera y captadores de información son elementos eléctricos, con lo que la regulación y la automatización son, por tanto, eléctricas o electrónicas.

Las ventajas de la electroneumática sobre la neumática pura son obvias y se concretan en la capacidad que tienen la electricidad y la electrónica para emitir, combinar, transportar y secuenciar señales, que las hacen extraordinariamente idóneas para cumplir tales fines. Se suele decir que la neumática es la fuerza y la electricidad los nervios del sistema.

En la figura 1, se muestra la representación gráfica, denominación y símbolo de los componentes electroneumáticos.

DENOMINACIÓN	SÍMBOLO
<p>Cilindro simple efecto</p> 	
<p>Cilindro doble efecto</p> 	

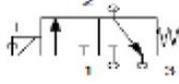
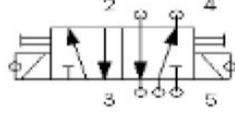
DENOMINACIÓN	SÍMBOLO
<p>Electrovalvula 3x2 monoestable</p> 	
<p>Electrovalvula 5x2 biestable</p> 	
<p>Pulsador</p> 	
<p>Relé</p> 	
<p>Detector de posición de avance</p> 	
<p>Detector de posición de retroceso</p> 	
<p>Alimentación de Tensión</p> 	
<p>FRL</p> 	
<p>Alimentación Neumática</p> 	

Figura 1: Representación gráfica, denominación y símbolo de componentes electropneumáticos

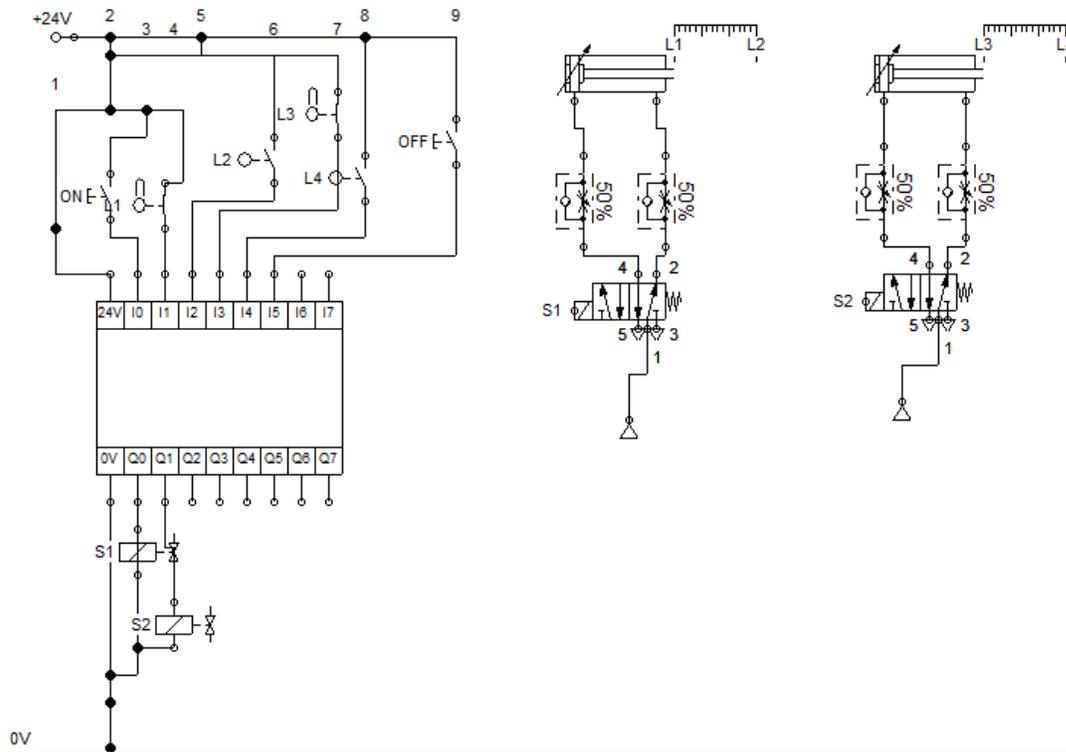


Figura 2: Representación esquemática del cableado electroneumático hecha en FLUIDSIM de FESTO

En la figura 2 se aprecia una aplicación hecha con el simulador FESTO FLUIDSIM en la que un microautómata programable es empleado como controlador.

Se puede escoger la secuencia de movimientos de los actuadores lineales neumáticos que se requiera para una aplicación, por ejemplo una de manipulación como la mostrada en la figura 3

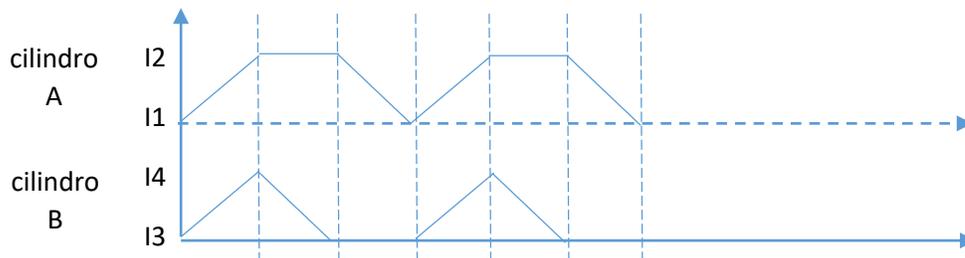


Figura 3: Secuencia cíclica de movimientos de cilindros A y B representada en el tiempo

El beneficio de FLUIDSIM de FESTO, es que es especializado para aplicaciones con actuadores neumáticos, sin embargo su autómata programable (Figura2) sólo se puede programar en esquema de funciones o de bloques (FUP).

A continuación se ve la simulación de la secuencia en CDe SIMU, controlado con un LOGO y programado con GRAFCET, nótese que en este autómata los bornes de entrada empiezan con I1 a diferencia del simulador de FESTO, en el que sus entradas empiezan de I0.

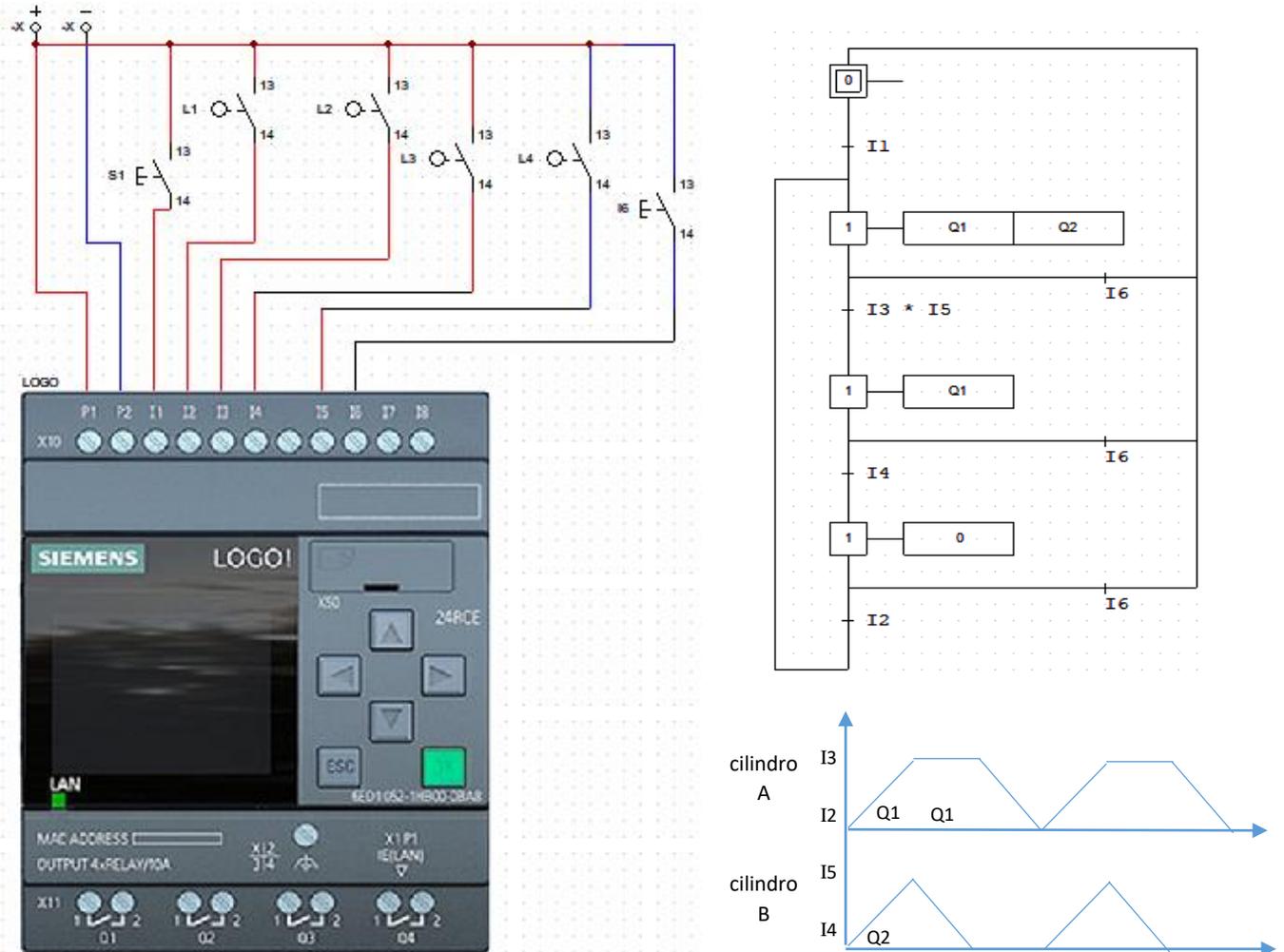


Figura 4: Simulación realizada en CADe SIMU

Dispositivo	Borne en el LOGO
Pulsador de inicio S1	Entrada I1
Final de carrera L1	Entrada I2
Final de carrera L2	Entrada I3
Final de carrera L3	Entrada I4
Final de carrera L4	Entrada I5
Pulsador de parada S2	Entrada I6
Conexión a solenoide válvula 1	Salida Q1
Conexión a solenoide válvula 2	Salida Q2

Tabla 1: Relación de las denominaciones de dispositivos y bornes del LOGO

Sin embargo cuando se realiza la simulación o programación en un autómata, ya sea con diagrama de contactos, esquema de funciones, lista de instrucciones o GRAFCET, no suele disponerse de los dispositivos neumáticos y sus correspondientes finales de carrera para hacer la simulación como con: FLUIDSIM de FESTO o con CADe SIMU, entonces debe conocerse la

representación de conexión y desconexión de las señales eléctricas de entrada como las señales eléctricas de salida, como se muestra en la siguiente figura:

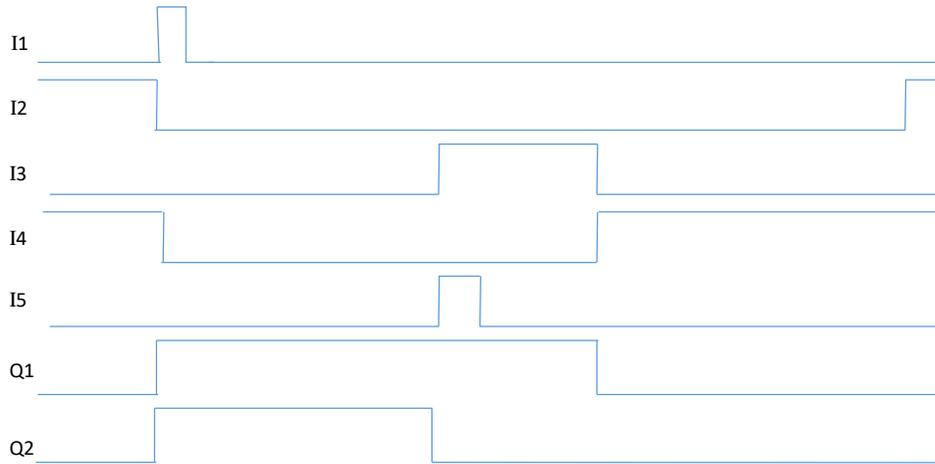


Figura 5: Representación de conexión/desconexión en el tiempo de las señales de entrada y salida del LOGO

ACTIVIDADES A REALIZAR

1. Simular en CADE SIMU el circuito mostrado en la figura 4, para ello tiene que dibujar dicho circuito y hacer la programación en lenguaje GRAFCET e incluya las conexiones a los actuadores neumáticos en las salidas de:
 - LOGO
 - SIMATIC
2. En los mismos diagramas de la pregunta anterior, programar y simular en:
 - LOGO (Esquema de funciones)
 - SIMATIC (Diagrama de contactos)
3. Para el mismo circuito mostrado en la figura 4, en CADE SIMU, cambiar los finales de carrera mecánicos por sensores de proximidad de tres hilos y verificar funcionamiento, indique si dichos sensores de proximidad son del tipo PNP o NPN, justifique su respuesta, incluya fotos de sensores de proximidad de tipo a) inductivo, b) capacitivo y c) óptico.
4. Realice el mismo programa en los programas en los simuladores de PLCs
 - eSms de Messenger
 - EASY-SOFT de Moeller
5. Compruebe operación basándose en representación de la figura 5, siguiendo la secuencia de conexiones para I1, I2, I3, I4 e I5, deben coincidir con las conexiones y desconexiones de Q1 y Q2 mostrado en la figura 5.

Bibliografía: <https://docplayer.es/36105020-6-electroneumatica-6-1-introduccion.html>

PRÁCTICA 5

Simulación de arranque estrella triángulo con micro PLC

OBJETIVO: Simulación del arranque estrella-triángulo mediante un micro autómatas programable basado en secuencia de condiciones de tiempo.

COMPETENCIA: Arranca un motor de inducción trifásico, realizando el cableado y programación con funciones de temporización en un micro autómatas programable con diagrama de bloques y de contactos.

SOFTWARE DE SIMULACIÓN RECOMENDADO: CADeSIMU, eSMS Config, EASY-SOFT.

FUNDAMENTO TEÓRICO: El arranque estrella triángulo se practica con el fin de reducir los niveles de corriente al momento de arrancar un motor de inducción trifásico, ya que la conexión estrella alimenta con menor voltaje a cada bobina del motor. La programación de este arranque, nos permitirá realizar secuencias en base a tiempo transcurrido, a diferencia de la programación de la práctica anterior con secuencia en base a eventos (pulsadores y finales de carrera), las secuencias en base a tiempo requieren el uso de funciones de temporización.

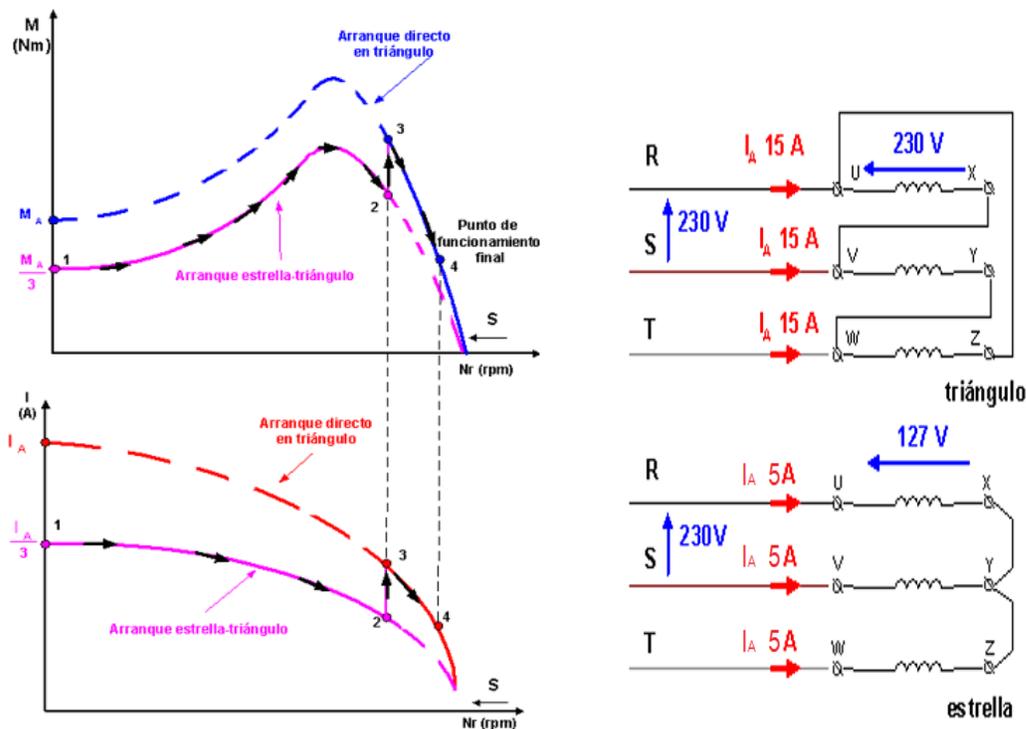


Figura 1: Características de conexión en en triángulo y estrella. Fuente: <https://sensoricx.com/maquinas-electricas/>

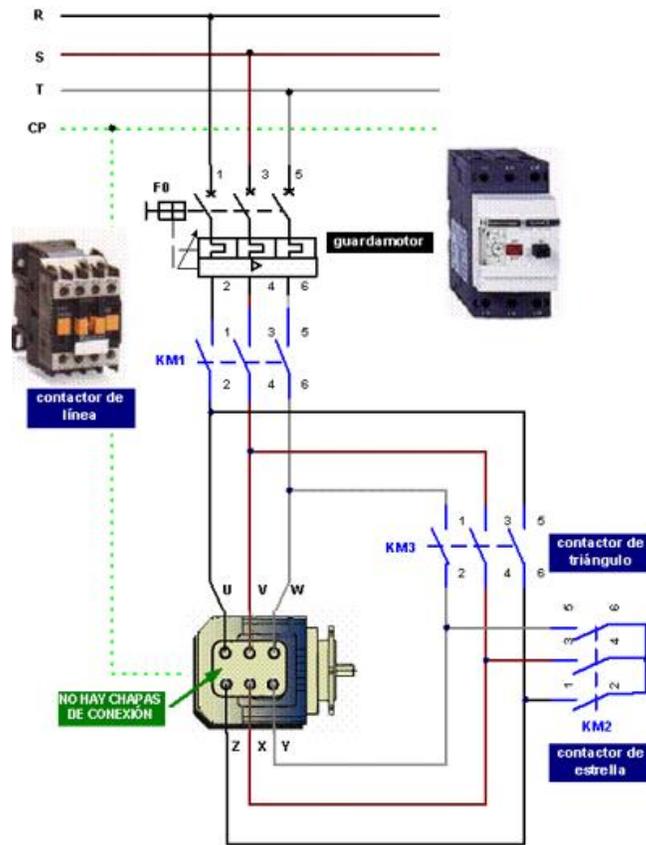
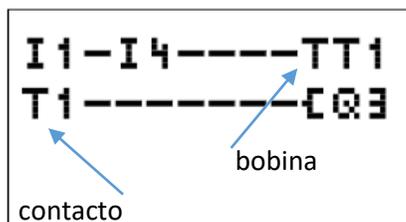


Figura 2: Representación esquemática del circuito de potencia de un arranque estrella triángulo

Para la programación del arranque estrella triángulo se hará uso de las funciones de temporización disponibles en todos los autómatas programables.

Existen varios tipos de funciones de temporización, pero lo recomendable para conocer el manejo de éstos, es recurrir a los manuales que presentan los fabricantes aunque todos ellos traten del mismo tipo de funcionamiento, es posible encontrar ligeras diferencias entre uno y otro, por lo que es importante hacer interpretación de las curvas temporales que presentan los mismos.

La marca EASY por ejemplo, ofrece 16 relés temporizadores T1 a T16 a escoger. Con relés temporizadores puede cambiarse el tiempo de conexión y la hora de conexión y desconexión de un contacto de maniobra.



Los tiempos de retardo regulables se encuentran entre 2 ms y 99 h 59 min. Como valores de consigna se pueden utilizar valores positivos, valores de entradas analógicas, valores reales de relés contadores y relés temporizadores.

Nótese la diferencia en representar bobina con contacto.

Fig. 3: Contacto T1 y bobina TT1 de un relé

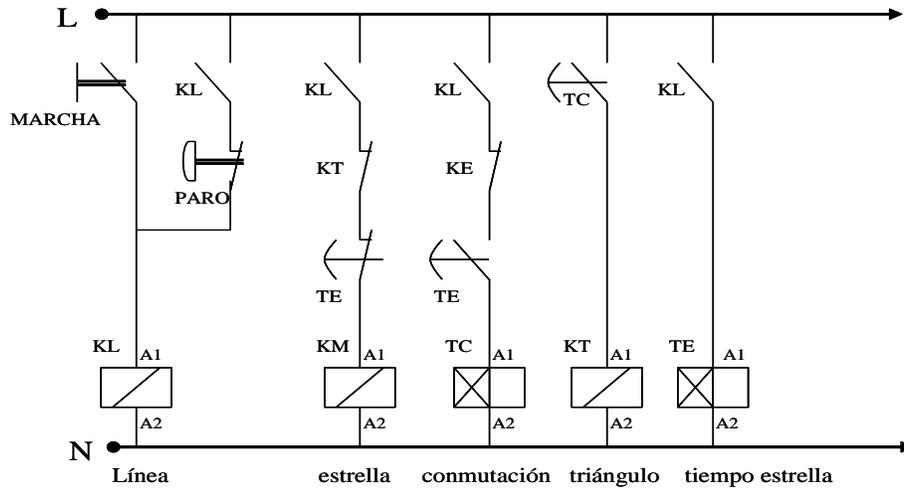


Figura 4: Representación esquemática del automatismo cableado para arranque estrella triángulo

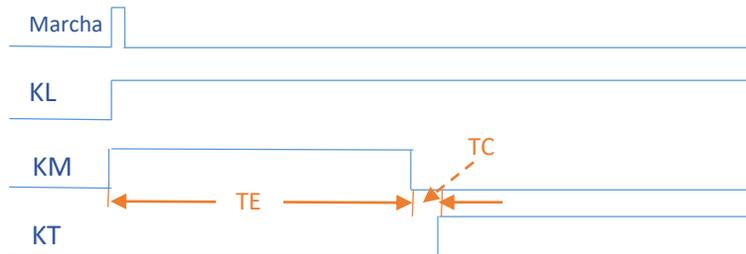


Figura 5: Operación básica de encendido estrella triángulo

Función de las bobinas

Representación del esquema de contactos	Indicador easy	Función de las bobinas	Ejemplo
	C	Función de contactor	C01, C02, C54, C:1, CM1
	D	Función de contactor con resultado denegado	D01, D02, D54
	L	Impulso de ciclo con flanco negativo	L03, L14, L08, L51
	F	Impulso de ciclo con flanco positivo	F04, F15, F01, F53
	I	Función de impulso de corriente	I03, I14, I08, I51
	S	Activar (enclavar)	S08, S12, S03, S54
	R	Desactivar (desenclavar)	R04, R15, R01, R53

Los relés auxiliares M y N se utilizan como "marcas". El relé S puede utilizarse como salida de una ampliación o como marca, en caso de que no exista ampliación. Sólo se diferencian del relé de salida Q por no tener ningún borne de salida.

Fig. 6: Funciones de bobina de autómatas EASY - Moller

Valoración de flancos negativos (impulso de ciclo) \downarrow
 Si la bobina sólo debe conectarse en un flanco negativo, se utilizará esta función. En caso de caída del estado de la bobina de "1" a "0", la bobina conecta durante un tiempo de ciclo sus contactos de cierre en el estado "1".

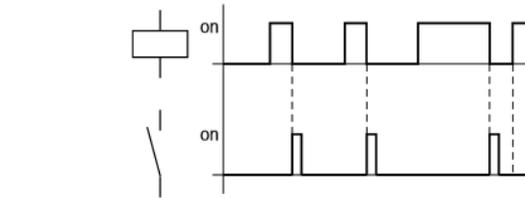


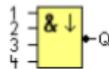
Figura 42: Diagrama de comportamiento Impulso de ciclo con flanco negativo

Representación en easy:

- Marcas M, N: $\downarrow M1$ a $\downarrow M1$ \downarrow , $\downarrow N1$ a $\downarrow N1$ \downarrow
- Saltos: \downarrow : 1 a \downarrow : 0

➔ Puesto que se generará un impulso de ciclo, no deberán utilizarse salidas físicas.

Fig. 7: Operaciones de flanco en autómatas programables diagrama escalera, Fuente EASY- Moeller



The output of a NAND with edge evaluation is only 1 **at least one** input is 0 and **all** inputs were 1 during the last cycle.

The output is set to 1 for the duration of one cycle and must be reset to 0 at least for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned: x = 1.

Timing diagram of a NAND with edge evaluation

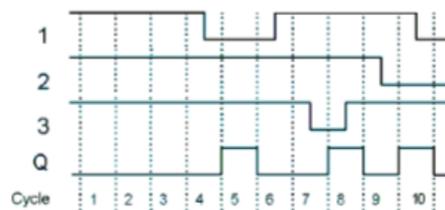


Fig. 8: Operaciones de flanco en autómatas programables con esquema de funciones, Fuente EASY x-Messenger

Puede utilizarse el método de grafos, como Grafcet, en caso de escogerse el primero, la aplicación de arranque estrella triángulo sería el siguiente:

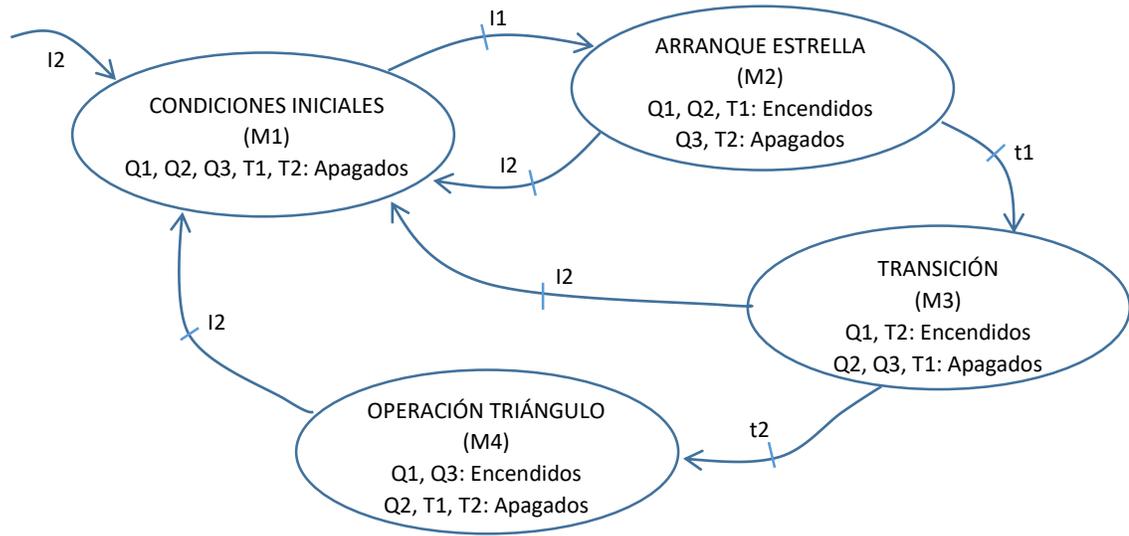


Fig. 9: Grafo para secuencia de arranque seguro estrella triángulo

Donde:

- Q1 es contactor principal de línea
- Q2 es contactor de operación estrella
- Q3 es contactor de operación triángulo
- T1 es la bobina del temporizador de retardo de activación, para cambio de estrella a triángulo, por ejemplo 5 s
- T2 es la bobina del temporizador de retardo de activación, por seguridad, para evitar que conexiones estrella y triángulo se activen simultáneamente, por ejemplo 50 ms
- I1 es pulsador de encendido
- I2 es pulsador de parada e inicialización
- t1 es contacto del temporizador T1
- t2 es contacto del temporizador T2

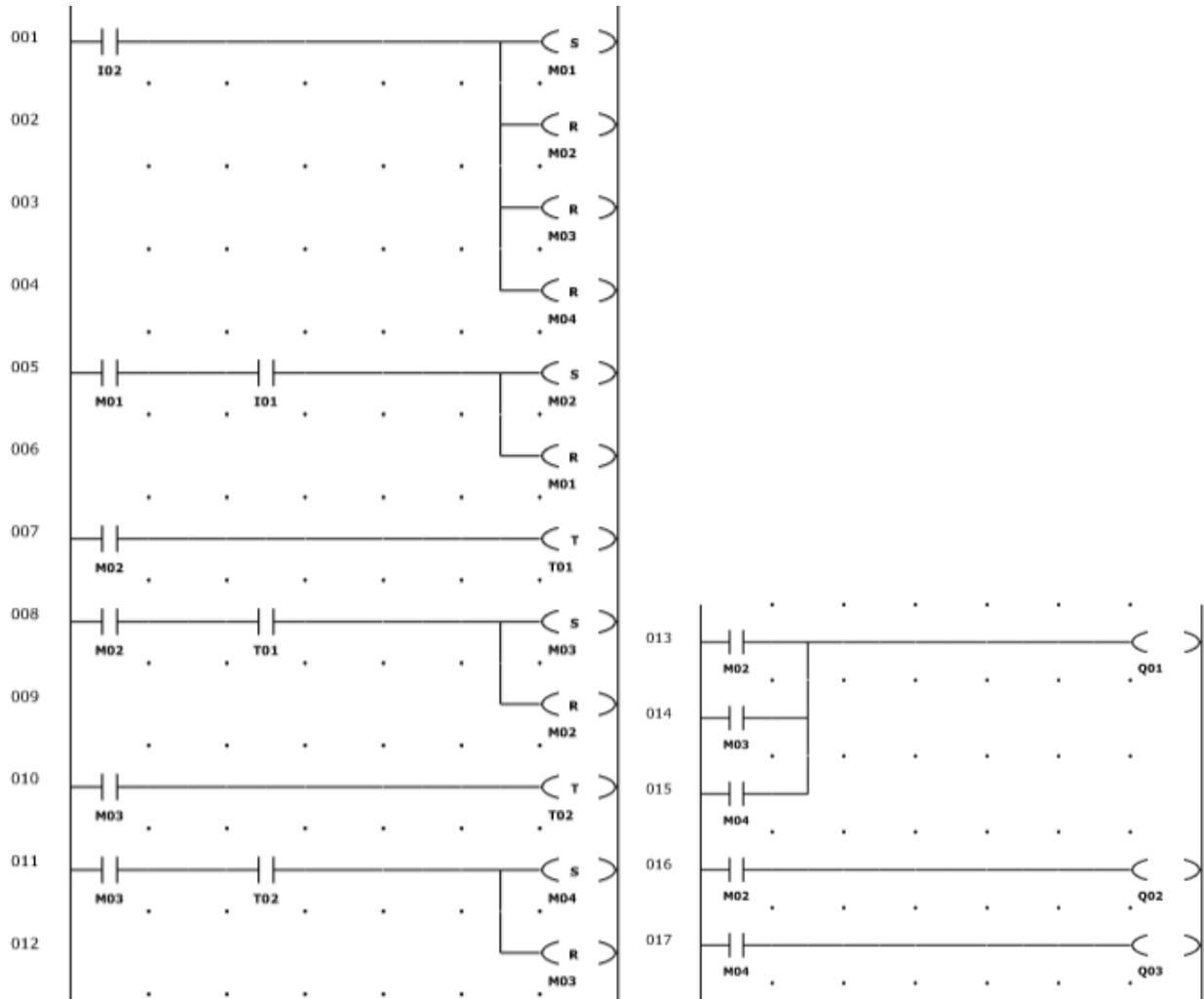


Fig. 10: Diagrama escalera para arranque estrella triángulo

En la figura 11, se muestra el circuito de potencia donde hay que aplicar las conexiones:

$$KM1 \rightarrow Q1 \quad KM2 \rightarrow Q3 \quad KM3 \rightarrow Q2$$

Debe tenerse mucho cuidado al hacer el circuito de potencia, por ejemplo de L1 de la línea principal se debe conectar a U1 y W2 del motor como se muestra en figura 11.

El método propuesto no es único, por ejemplo en la figura 12 se muestra el control con un automatismo o circuito de control cableado y, en la figura 13 una combinación de programa y cableado.

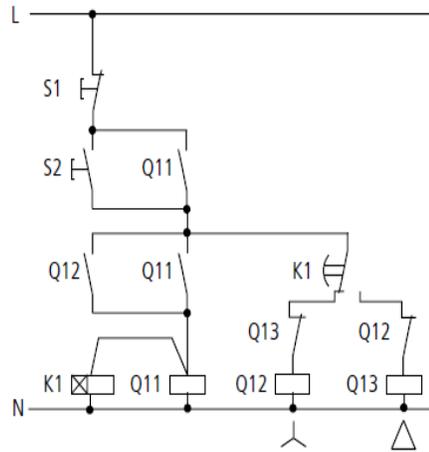
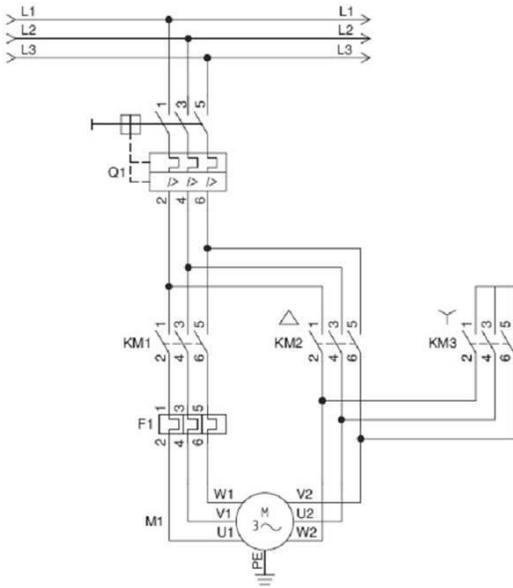


Fig. 11: Circuito potencia arranque estrella-triángulo

Fig. 12: Circuito de control con automatismo

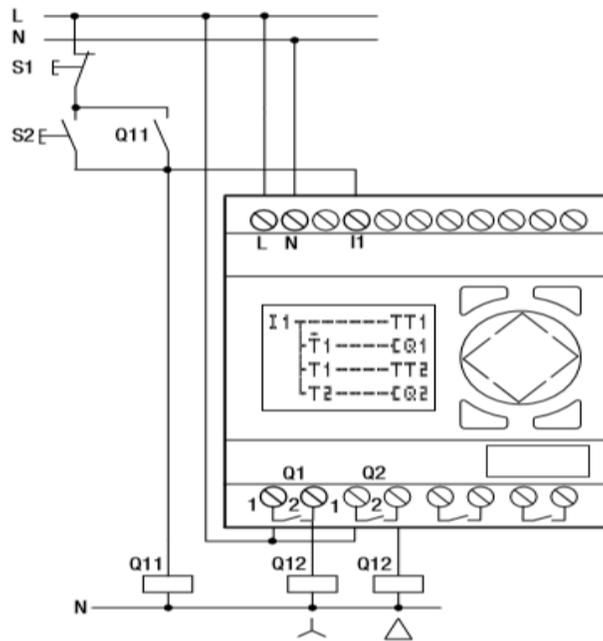


Fig. 13: Control con autómata EASY-Moller para arranque estrella triángulo (ejemplo obtenido de Manual de usuario 01/05 AWB2528-1508E)

Especificaciones de temporizadores proporcionados por fabricantes de micro autómatas programables disponibles en laboratorio.

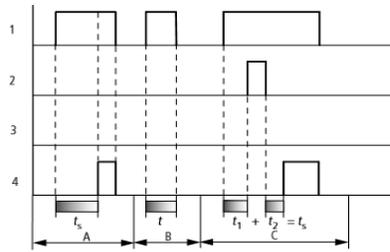


Diagrama de comportamiento Relé temporizador con temporización de trabajo (con/sin conexiones aleatorias)

- 1: bobina de disparo TTx
- 2: bobina de detención HTx
- 3: bobina reset RTx
- 4: contacto de maniobra (contacto de cierre) Tx

- t_s : tiempo nominal
- Margen A: el tiempo nominal ajustado se para de forma normal.
- Margen B: el tiempo nominal ajustado no se para porque la bobina de disparo se desexcita demasiado pronto.
- Margen C: la bobina de detención detiene el transcurso del tiempo.

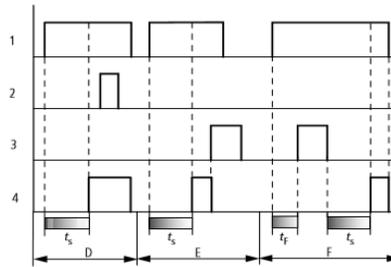


Diagrama de comportamiento Relé temporizador con temporización de trabajo (con/sin conexiones aleatorias)

- Margen D: la bobina de detención deja de tener efecto una vez transcurrido el tiempo
- Margen E: la bobina reset desactiva el relé y el contacto
- Margen F: la bobina reset desactiva el tiempo durante el transcurso. Una vez se ha desexcitado la bobina reset, el tiempo se para de forma normal.

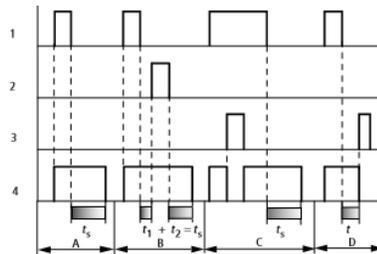


Diagrama de comportamiento Relé temporizador con temporización de reposo (con/sin conexiones aleatorias)

- 1: bobina de disparo TTx
 - 2: bobina de detención HTx
 - 3: bobina reset RTx
 - 4: contacto de maniobra (contacto de cierre) Tx
- t_s : tiempo nominal

- Margen A: tras la desconexión de la bobina de disparo el tiempo se para.
- Margen B: la bobina de detención detiene el transcurso del tiempo.
- Margen C: la bobina reset desactiva el relé y el contacto. Una vez se desexcita la bobina reset, el relé sigue trabajando de forma normal.
- Margen D: la bobina reset desactiva el relé y el contacto durante el transcurso del tiempo.

Fig. 14: Diagramas de operación de temporizador; Fuente EASY-Moller

View in xLogic	Name of the special function	Rem
Timer		
	On-delay	REM
	Off-delay	REM
	On-/Off-delay	REM
	Retentive on-delay	REM

Fig. 15: Operadores de temporización en diagrama de esquema de funciones o de bloques; Fuente LOGO

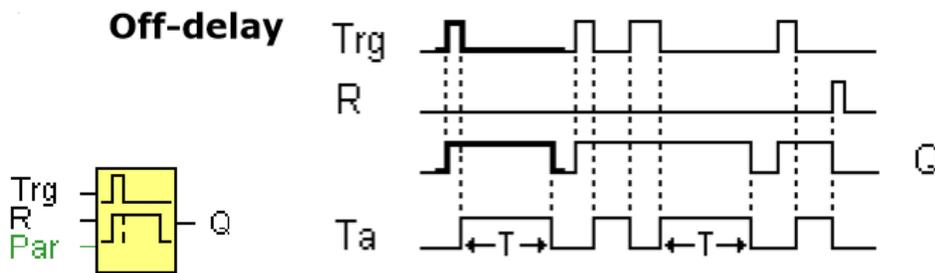


Fig. 16: Diagramas de operación de temporizador; Fuente LOGO

ACTIVIDADES A REALIZAR

1. Simule en CADeSIMU el automatismo cableado mostrado en la figura 4.
2. Simule en EASY-MOLLER el automatismo programado con diagrama de contactos según el programa mostrado en la figura 10 y posteriormente el mostrado en la figura 13.
3. Simule en CADeSIMU el automatismo programado con LOGO con esquema de funciones cuyas instrucciones se muestran en figuras 15 y 16.
4. Dibuje el GRAFCET equivalente al grafo de la figura 9.
5. Dibuje los esquemas eléctricos de los circuitos como realmente se verían correspondiente a alguna de las preguntas anteriores (usted escoja).

PRÁCTICA 6

Simulación control motor paso a paso con autómeta programable

OBJETIVO: Simulación en control básico de motores paso a paso.

COMPETENCIAS: Controla posición de un motor paso a paso monopolar, conectando y programando el mismo a través de un micro PLC.

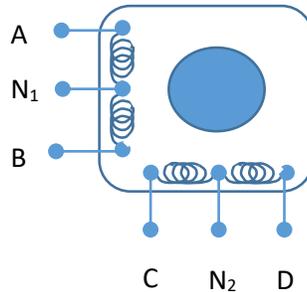


Figura 1: Esquema de un motor paso a paso monopolar con seis hilos

SOFTWARE DE SIMULACIÓN RECOMENDADO: eSMS Config (xLogicsoft), EASY-SOFT.

FUNDAMENTO TEÓRICO:

Secuencias para manejar motores paso a paso (unipolar)

Paso simple: Esta secuencia de pasos es la más simple de todas y consiste en activar cada bobina una a una y por separado, con esta secuencia de encendido de bobinas no se obtiene mucha fuerza ya que solo es una bobina cada vez la que arrastra y sujeta el rotor del eje del motor, véase figura 2.

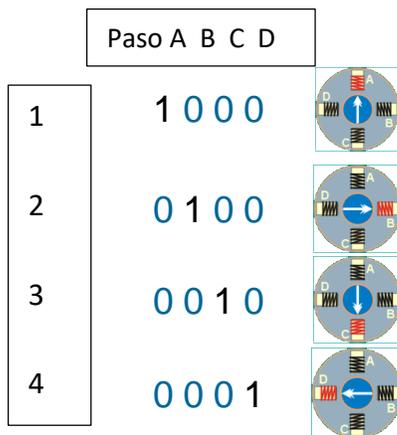


Fig. 2: Paso simple

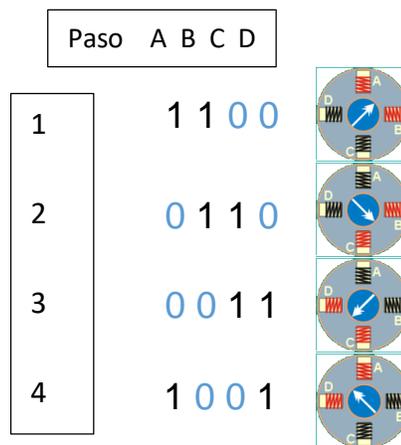


Fig. 3: Paso doble

Paso doble: Con el paso doble activamos las bobinas de dos en dos con lo que hacemos un campo magnético más potente que atraerá con más fuerza y retendrá el rotor del motor en el sitio con mayor intensidad de cuando se empleaba la configuración anterior de paso simple, véase figura 3

Medio Paso:

Combinando los dos tipos de secuencias anteriores podemos hacer moverse al motor en pasos más pequeños y precisos y así pues tenemos el doble de pasos de movimiento para el recorrido total de 360° del motor, véase figura 4.

Paso	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

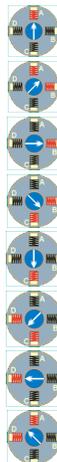


Fig.4: Medio paso

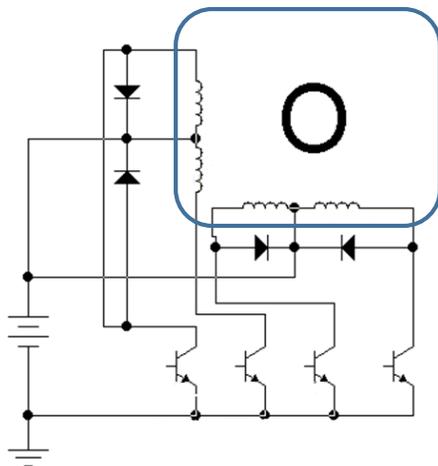


Figura 5: Conexión de motor monopolar

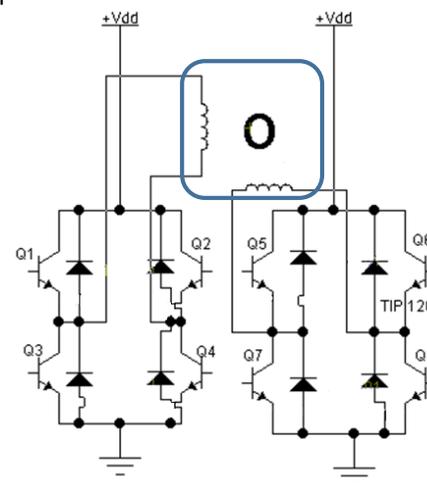


Figura 6: Conexión de motor bipolar

Aplicación de función de contador para limitar el número de pulsos en un motor paso a paso mediante un micro autómatas programable haciendo uso de salidas tipo transistor y tipo relé.

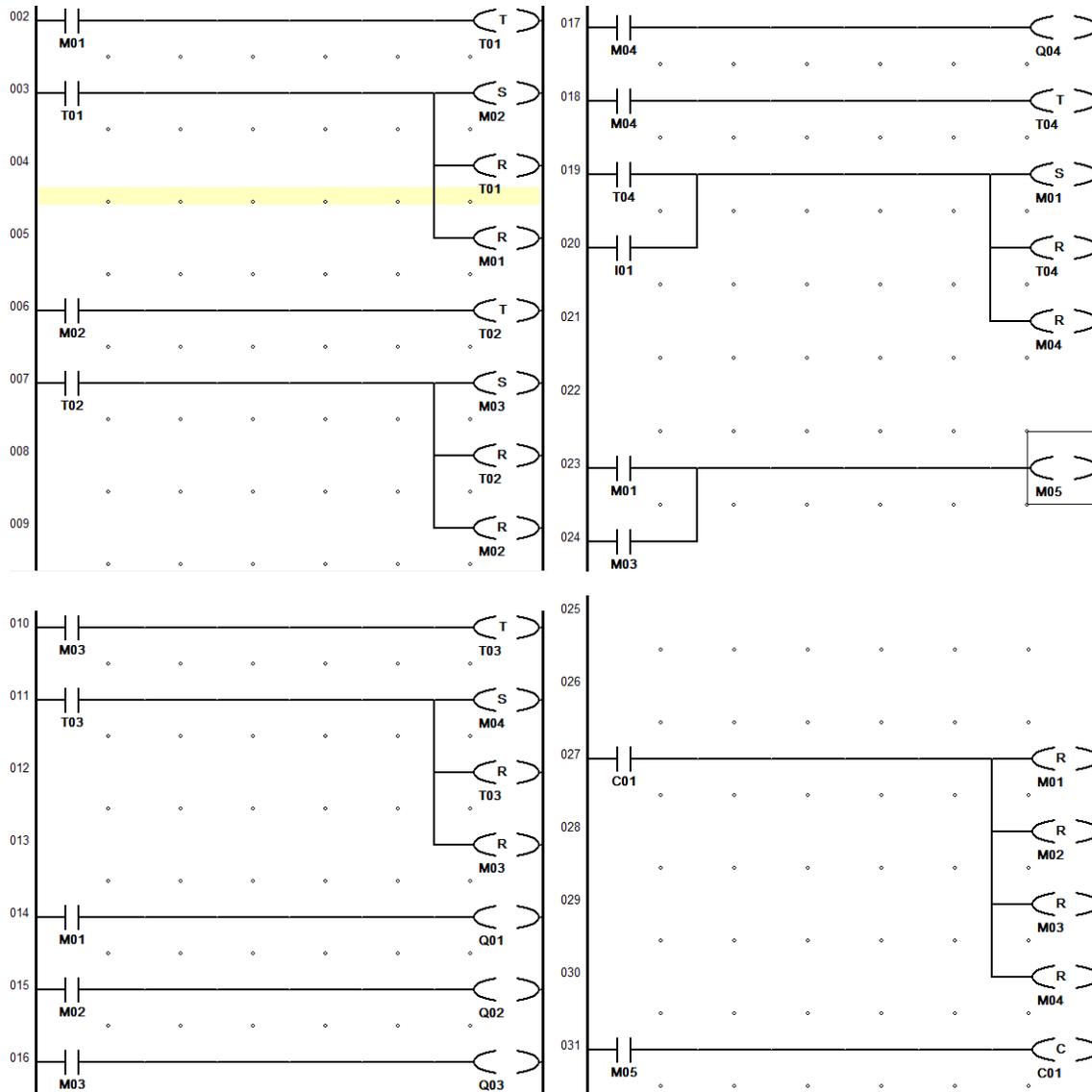


Figura 10: Programa en lenguaje escalera para controlar un motor paso en paso simple A,B,C,D,A,B,...

El programa incluye un contador, mismo que en este caso cuenta las veces que se activan las marcas M01 y M03; una vez llegado al valor de consigna, el contacto auxiliar “resetea” a todas las marcas parando de este modo al motor.

El programa de arriba es el más simple para accionar un motor paso a paso simple, no incluye pulsador de parada o contador para delimitar el número de pulsos del motor. Es posible simplificar dicho programa si en vez de usar un retardo de activación con un RS, se usara un retardo de activación memorizado.

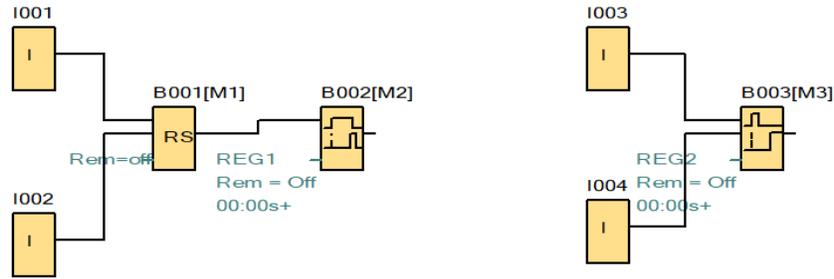


Figura 11: Equivalencia de programación RS + On Delay = Retentive On Delay

OBSERVACIONES A REALIZAR CON UN MOTOR MONOPOLAR:

Identifique las terminales del motor paso a paso:

Con un ohmímetro mida la resistencia entre las diferentes terminales del motor, y debe obtener:

- $R_{A-N1} = R_{B-N1} = R_{C-N2} = R_{D-N2} = R$
- $R_{A-B} = R_{C-D} = 2 \cdot R$
- $R_{(A, N1, B) - (C, N2, D)} = \infty$

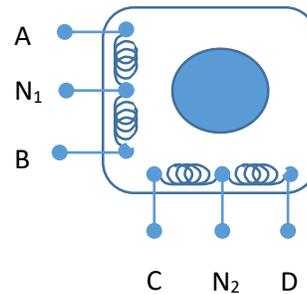


Fig. 12: Esquema motor paso a paso

Una los cables N₁ con N₂ y conéctelos a la fuente positiva de la fuente con el voltaje recomendado por el fabricante del motor, luego conecte el borne negativo o de tierra de la fuente con las terminales A, B, C y D del motor (uno por vez), si sigue correctamente la secuencia, el eje del motor girará paso a paso, la cantidad de grados por paso especificado por el fabricante. Si no estuviera bien la secuencia, dígase A, B, D, C, podrá apreciar que el eje del motor no gira de modo continuo.

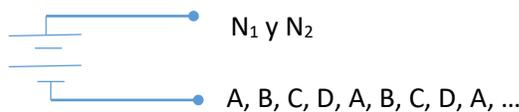


Fig. 13: Alimentación de un motor paso a paso

Verifique el torque que genera el eje del motor agarrándolo suavemente con los dedos. Si incrementa el valor de voltaje de la fuente verificará que el torque se hace mayor, por otra parte si cambia la secuencia a A, D, C, B, A, D, C, B, A, ..., el sentido de rotación se invierte.

Mida la cantidad de grados que gira el motor por cada pulso o paso.

Realice uno de los programas mostrados en figuras 9 y 10 según qué tipo de programa dispone el autómatas a utilizar y simúlelo.

La figura 14 muestra la conexión al autómatas de salida de transistores. Los diodos se instalan para disipar corrientes inducidas en las desconexiones.

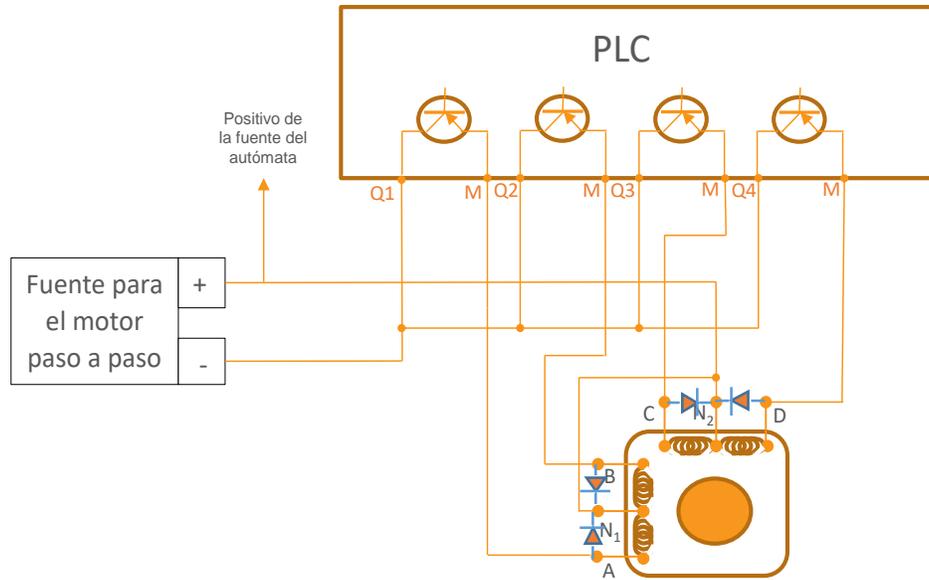


Figura 14: Conexión del motor paso a paso monopolar a un autómata con salida de transistores

ACTIVIDADES A REALIZAR:

Dado un motor paso a paso monopolar de 2º por pulso y conocida la secuencia A, B, C y D de conmutación y que no cuenta con “driver” de motor paso a paso:

1. Realizar un programa para PLC (diagrama escalera y esquema de funciones) para generar señales paso doble a 3 rpm
2. Realizar un programa para PLC (diagrama escalera y esquema de funciones) para generar señales “half step” o medio paso a 1 rpm
3. Al primer programa aplique un contador para que el motor pare al haber girado 30º.
4. ¿Cómo programaría un proceso de incremento de velocidad gradual?
5. Investigue sobre “drivers” de motores paso a paso y muestre la conexión eléctrica de uno de ellos.

PRÁCTICA 7

Aplicaciones de simulación con señales analógicas

OBJETIVO: Comprensión en el manejo de sensores analógicos para operaciones de comparación.

COMPETENCIA: Comprende a través de la simulación, la utilidad de emplear señales analógicas en operaciones de comparación para control.

SOFTWARE DE SIMULACIÓN RECOMENDADO: eSMS Config (xLogicsoft), EASY-SOFT.

FUNDAMENTO TEÓRICO:

Debido a la necesidad de incluir en los sistemas de control señales provenientes de sensores analógicos, los autómatas programables suelen incluir también módulos de entrada para señales analógicas, tanto en diferencia de voltaje como en intensidad de corriente, que están normados en los siguientes rangos:

- Diferencia de potencial o voltaje de 0 a 10 V
- Intensidad de corriente eléctrica de 4 a 20 mA

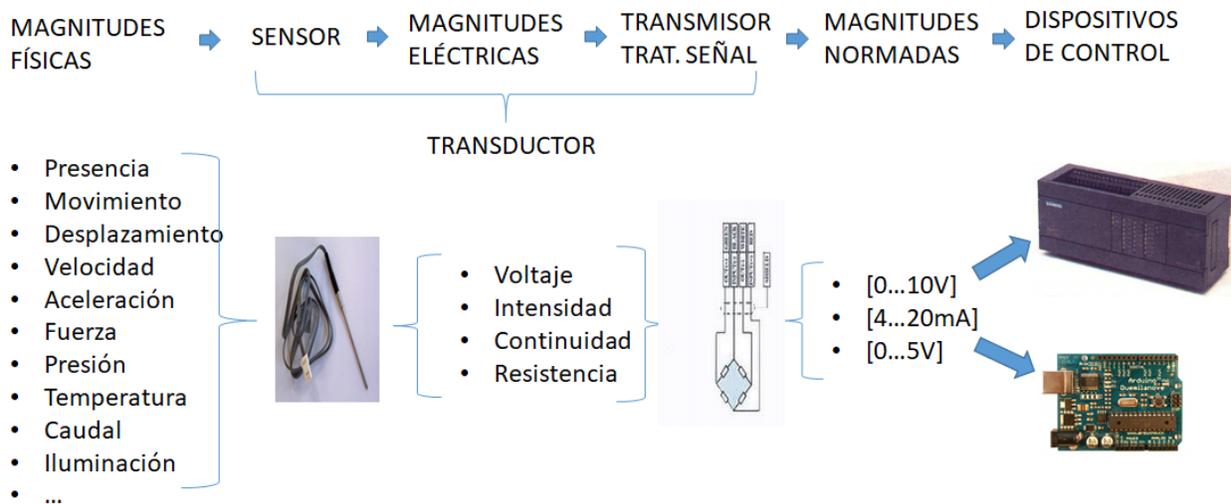


Figura 1: Interfaz entre magnitudes físicas y dispositivos de control

Dependiendo de la gama del autómata programable, éste puede realizar las siguientes operaciones con señales analógicas:

- Comparación
- Operaciones aritméticas
- Realimentación en control PID

En la presente práctica, se hará uso de las funciones de comparación de señales analógicas, de bastante aplicación para control de procesos y disponible en todos los autómatas que tienen módulos de entradas analógicas.

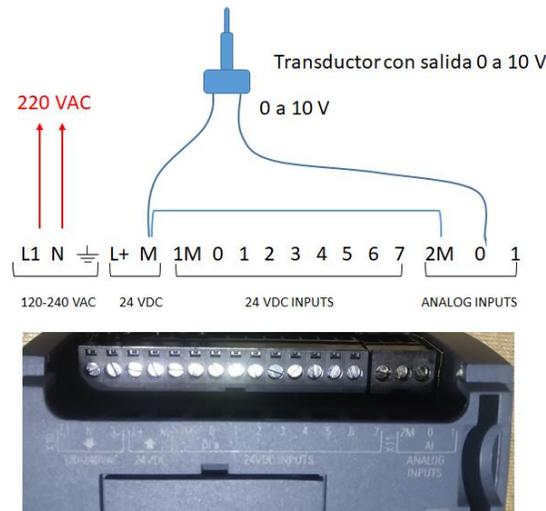


Figura 2: Vista de la conexión de un transductor a la entrada analógica de un autómata

Si se dispone de un transductor cuya salida entregue un rango de voltaje de 0 a 10 V, la conexión puede hacerse directamente como se muestra en la figura 2. Si en cambio, el sensor no genera una salida de 0 a 10 V, tendría que utilizarse un transmisor para obtener una señal de esa índole y rango, como se muestra en la figura 3.

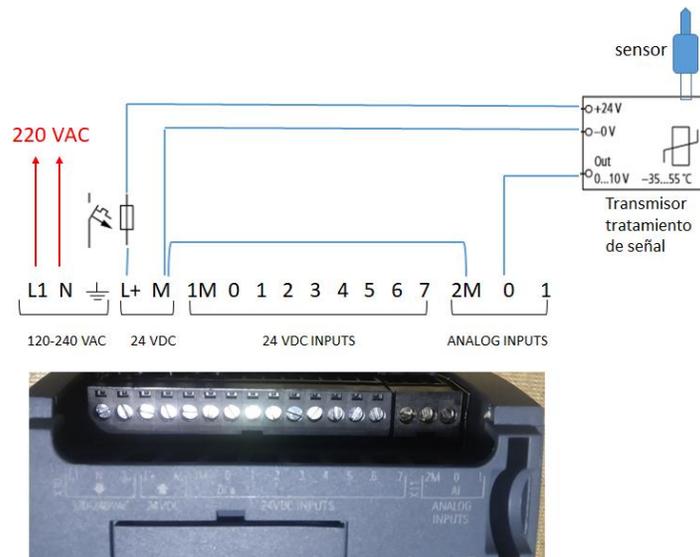


Figura 3: Vista de la conexión de un sensor con un transmisor de tratamiento de señal

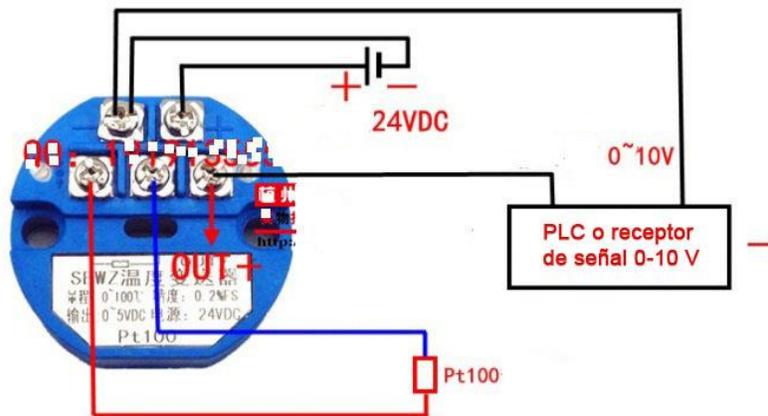


Figura 4: Transmisor de temperatura para sensor Pt100 utilizado en clases presenciales

A continuación se muestran especificaciones de operadores de comparación del micro autómatas EASY de Moller del inventario de microautómatas del laboratorio de la asignatura.

Valor en la entrada de valor del relé de función I1	Funciones de comparación		Valor en la entrada de valor del relé de función I2
		Selección del modo operativo en el relé de función	
Entrada analógica I7, I8, I11, I12			Entrada analógica I7, I8, I11, I12
Valor de consigna 0000 a 9999			Valor de consigna 0000 a 9999
Valor real del relé contador C1 a C16			Valor real del relé contador C1 a C16
Valor real del relé contador T1 a T16			Valor real del relé contador T1 a T16
	menor que	LT	
	menor o igual que	LE	
	igual	EQ	
	mayor o igual que	GE	
	mayor que	GT	

Figura 5: Funciones de comparación de EASY Moller

Relé de función A1 entrada de valor I1		Relé de función A1 entrada de valor I2
17	GE (mayor o igual)	18
17	LE (menor o igual)	18
17	GE (mayor o igual)	Valor de consigna
17	LE (menor o igual)	Valor de consigna
18	GE (mayor o igual)	Valor de consigna
18	LE (menor o igual)	Valor de consigna

Fig. 6: Ejemplos de comparación, Fuente EASY-Moller

Resolución de entradas analógicas I7, I8, I11 e I12 en EASY Moller 700; poseen la siguiente resolución: La señal analógica de 0 a 10 Vdc se transforma en un valor digital de 10 bits de 0 a 1023. Un valor de 100 digital significa un valor de 1,0 V analógico (exactamente 0,98 V).

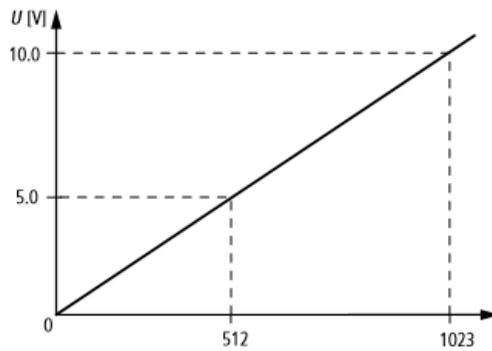


Fig. 7: Resolución de entradas analógicas EASY Moller 700

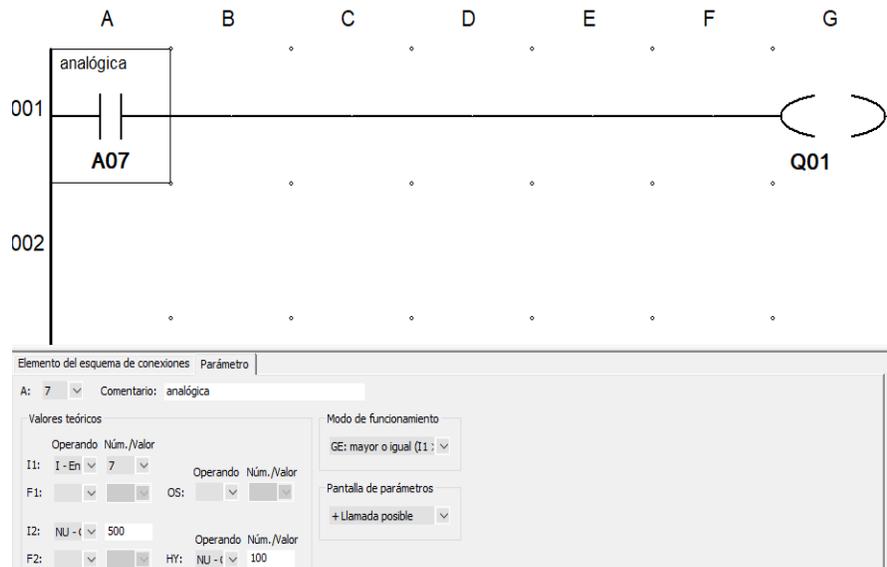


Fig. 8: Ejemplo de programación en EASY-Moller

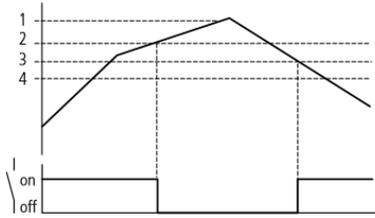


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "menor que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 9: Modo "menor que"

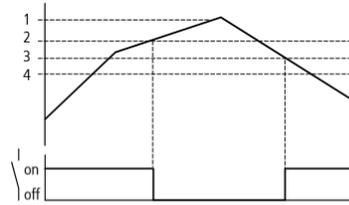


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "menor o igual que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 10: Modo "menor o igual que"

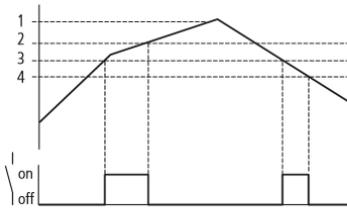


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "igual que"

- 1: valor real en I8, multiplicado con el factor de multiplicación F2
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 11: Modo "igual que"

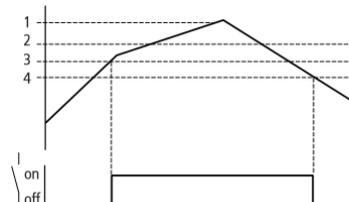


Diagrama de comportamiento Comparador de valores analógicos en el modo operativo "mayor o igual que"

- 1: valor real en I7
- 2: valor de consigna más valor de la histéresis
- 3: valor de consigna
- 4: valor de consigna menos histéresis

Fig. 12: Modo "mayor o igual que"

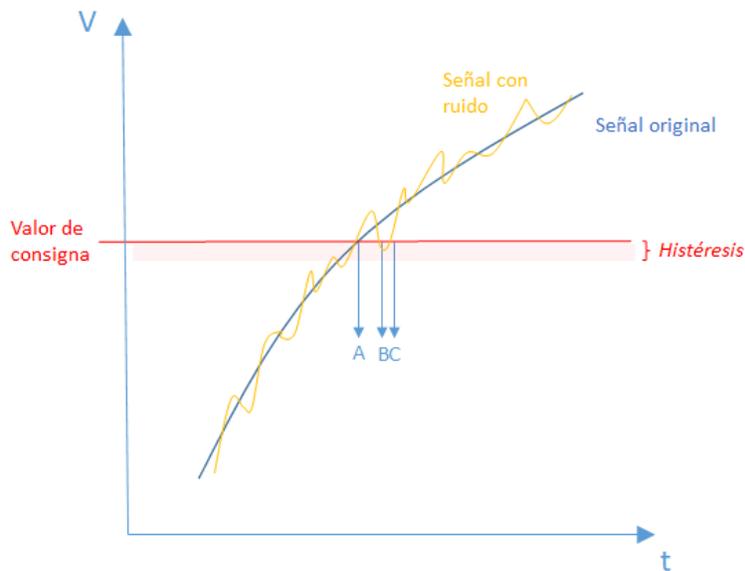


Figura 13: El rol de la histéresis en la eliminación de ruido

En la figura 13 se muestra la importancia de la histéresis para evitar acciones insulsas de conexión y desconexión ante la presencia de ruido en la señal.

En la figura 14, se aprecia el proceso de simulación con entradas analógicas con diagramas de bloque o esquemas de funciones correspondiente al modelo de microautómata “Messenger” de eSms.

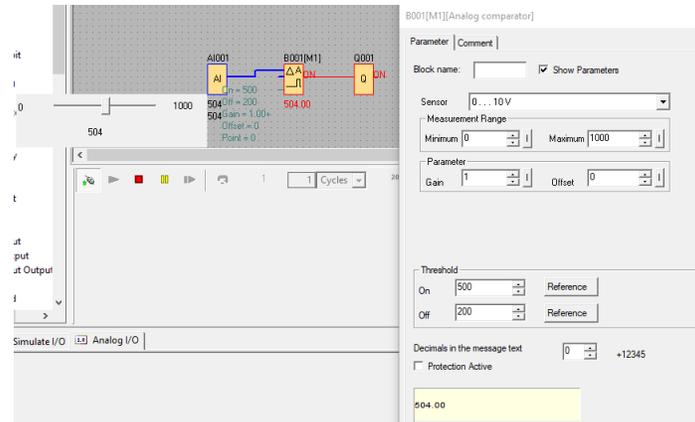


Fig. 14: Simulación de entradas analógicas con diagramas de bloque

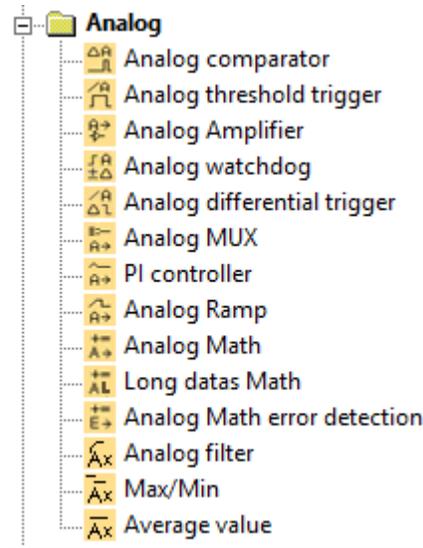


Figura 15; Menú de funciones analógicas en Messenger

La figura 15 muestra funciones analógicas disponibles en Messenger, de las cuales, se emplearán en la práctica de hoy: “Analog comparator”, “Analog threshold trigger”, “Analog differential trigger” y “Analog Math”.

La figura 16 muestra que el rango de medición en este modelo es de 0 a 1000.

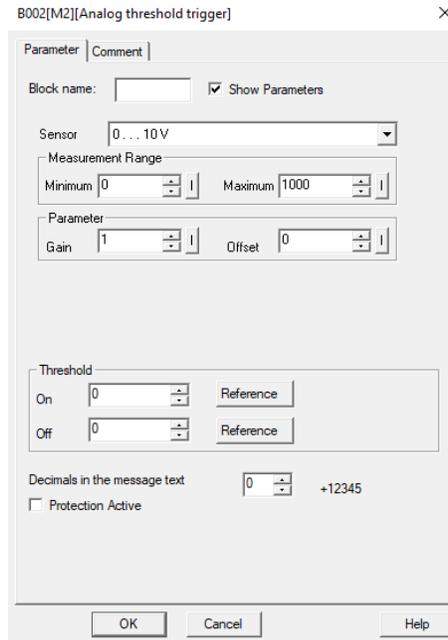


Figura 16: Parámetros de las funciones de comparación

CELDA	TRANSMISOR DE CELDA		Voltaje externo
color cable	Input	Output	
Negro	GND	GND	Fuente -
Blanco	S- (S+ tracción)	V0	Salida V
Verde	S+ (S- tracción)	I0	
Rojo	E+	24V	Fuente +

Tabla 1: Conexiones en la celda de carga DAYANG



(a) con tapa



(b) sin tapa

Figura 17: Transductor para celda de carga

Fuente: <https://toolsdirectsales.com/products/0-5v-0-10v-4-20ma-load-cell-sensor-amplifier-transmitter-strain-gauge-transducer?variant=50526504917>

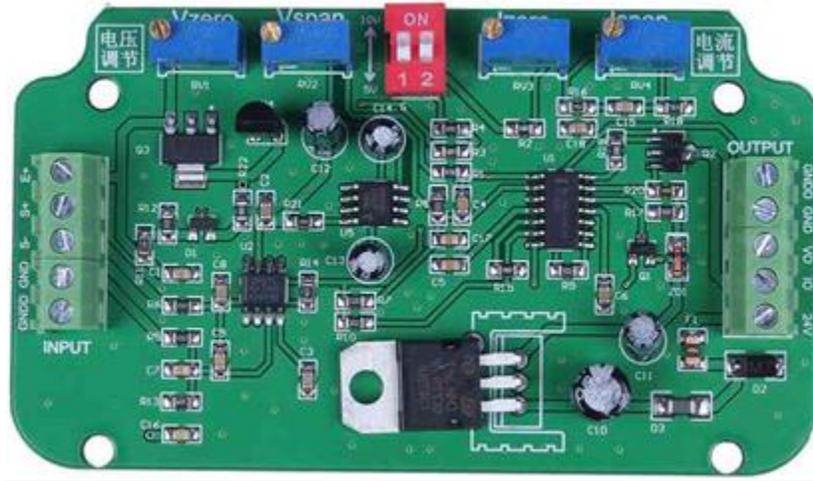
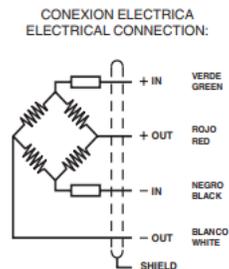


Figura 18: Vista ampliada de la tarjeta del transductor para celda de carga

Fuente: <https://toolsdirectsales.com/products/0-5v-0-10v-4-20ma-load-cell-sensor-amplifier-transmitter-strain-gauge-transducer?variant=50526504917>

ESPECIFICACIONES			SPECIFICATIONS		
Cargas nominales (Ln)	250-500-1000- 2000-5000- 7500	kg	Nominal capacities (Ln)		
Clase de precisión	3000	n. OIML (2)	Accuracy class		
Carga mínima	0	%Ln	Minimum dead load		
Carga de servicio	150	%Ln	Service load		
Cargas límite	200	%Ln	Safe load limit		
Error combinado	< ±0.017	%Sn (1) (2)	Total error		
Error repetibilidad	< ±0.015	%Sn	Repeatability error		
Efecto de la temperatura: en el cero	< ±0.01	%Sn/5°K	Temperature effect: on zero		
en la sensibilidad	< ±0.006	%Sn/5°K	on sensitivity		
Error de fluencia (30 minutos)	< ±0.016	%Sn (2)	Creep error (30 minutes)		
Compensación de temperatura	-10...+40	°C	Temperature compensation		
Límites de temperatura	-20...+70	°C	Temperature limits		
Sensibilidad nominal (Sn)	2 ±10%	mV/V	Nominal sensitivity (Sn)		
Tensión de alimentación nominal	10	V	Nominal input voltage		
Tensión de alimentación máxima	15	V	Maximum input voltage		
Resistencia de entrada	400 ±20	Ω	Input impedance		
Resistencia de salida	350 ±3	Ω	Output impedance		
Desequilibrio inicial	< ±2	%Sn	No load output		
Resistencia de aislamiento	> 5000	MΩ	Insulation resistance		
Deformación máxima (a Ln)	< 0.5	mm	Maximum deflection (at Ln)		

(1) Error combinado: No Linealidad e Histéresis / Total error: Non Linearity and Hysteresis
 (2) Ln = 250 kg; 2000 n. OIML



«CONVENIO SIGNOS SALIDA PARA TRACCIÓN»
 «OUTPUT SIGNS FOR TENSION APPLICATION»

Figura 19: Especificaciones de celdas de carga

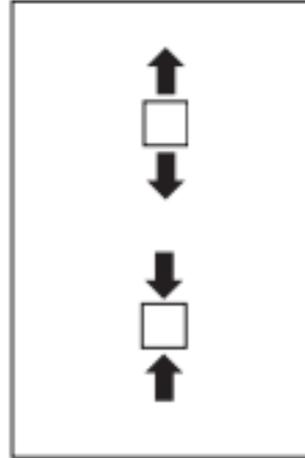
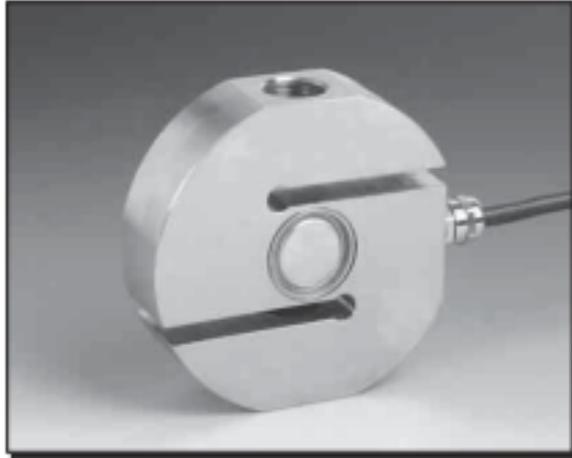
Fuente: https://www.disai.net/wp-content/uploads/catalogos_pdf/CELULAS_DE_CARGA.pdf

Ac



MODELO 650

250kg...7500kg



- Célula de carga de tracción/compresión
- Construcción en acero inoxidable
- Herméticamente soldada, protección IP 68 (EN 60529)
- 3000 divisiones O.I.M.L. R60 clase C
- Opción **ATEX** en proceso
Zona 0-1-2 (gas) y 20-21-22 (polvo)
- Tension/compression load cell
- Stainless steel construction
- Hermetically welded, protected IP 68 (EN 60529)
- 3000 divisions O.I.M.L. R60 class C
- **ATEX** option in process
Zone 0-1-2 (gas) and 20-21-22 (dust)

Modelo Model	Carga nominal Nominal capacity Ln	Clase de precisión Accuracy class n. OIML	División mínima Minimum division vmin	Carga de servicio Service load 150 % Ln	Carga límite Safe load 200 % Ln
650 250 kg	250 kg	2000	34 g	375 kg	500 kg
650 500 kg	500 kg	3000	50 g	750 kg	1000 kg
650 1000 kg	1000 kg	3000	100 g	1500 kg	2000 kg
650 2000 kg	2000 kg	3000	200 g	3000 kg	4000 kg
650 5000 kg	5000 kg	3000	500 g	7500 kg	10000 kg
650 7500 kg	7500 kg	3000	750 g	11250 kg	15000 kg

Figura 20: Vista de una celda de carga tipo S y especificaciones

Fuente: https://www.disai.net/wp-content/uploads/catalogos_pdf/CELULAS_DE_CARGA.pdf

Salidas analógicas de un autómata programable

Así como en las entradas de los autómatas, las salidas normalmente están disponibles en voltaje de 0 a 10 V y en intensidad de 4 a 20 mA. El modelo XLogic del laboratorio está provisto del módulo de expansión ELC-E-AQ (V) que permite tener dos salidas analógicas de 0 a 10 VDC.

El módulo de expansión debe conectarse al microautómata mediante el Puerto de expansión RS485 y alimentarse independientemente con 24 VDC y seleccionar en su programación (xLogic Soft) la siguiente instrucción mostrada en la figura 21.

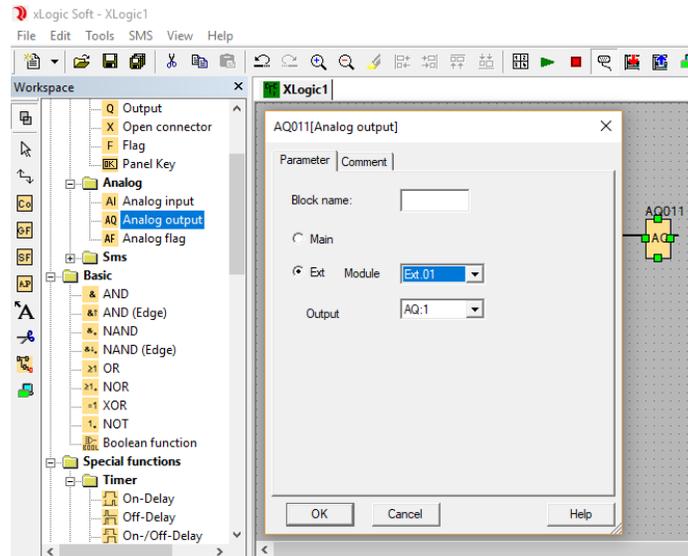


Fig. 21: Selección de módulo externo en xLogic Soft

Las salidas analógicas permiten modular el accionar de los actuadores, por ejemplo conjuntamente a las entradas analógicas, se pueden realizar controles en lazo cerrado PID.

Como ejemplo de actuador disponible en el laboratorio de la asignatura, controlado por una salida analógica, se tiene una válvula de bola motorizada (figura 22), es decir que la apertura de la misma está en función del voltaje aplicado.



Fig. 22: Vista de la válvula motorizada para modulación de flujo controlado por voltaje o corriente, fuente Winner Control Ball Valves Co., Ltd.

La válvula debe alimentarse con 24VDC (café + negra -) y conectar su entrada (amarilla) de 0~10 V a la salida analógica del módulo de expansión del microautómata, con tierra común (terminal negativa de alimentación "24 V con alimentación negativa de la válvula "negra" y del módulo de expansión analógica "M" del microautómata).

ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

1. Programar y simular en Easy de Moller (escalera) y en xLogic o Messenger (bloques o funciones) para que se active una salida cuando una de sus señales analógicas sea 4 V mayor a la otra.
2. Programar y simular en Easy de Moller (escalera) y en xLogic o Messenger (bloques o funciones) para que se active una salida cuando la entrada tenga un valor de consigna de 4V e histéresis de 1 V.
3. Programar en xLogic o Messenger (bloques o funciones) para modular la apertura de una válvula motorizada de modo que ésta esté en el doble de rango de lo que se comande con una entrada analógica. Dibuje cómo se vería físicamente su conexión.
4. Dibujar todas las conexiones a realizar para una celda de carga – transmisor – autómata programable según lo mostrado y especificado en tabla 1 y figuras 17, 18, 19 y 20.
5. Para la pregunta anterior, si la celda de carga tiene una capacidad de 250 Kg y el CAD (Convertidor Analógico Digital) del autómata programable es de 16 bits, suponiendo que se aprovecha todo el rango de peso en el rango 0 a 10 V y, que la relación es lineal; Indique:
 - La resolución en Kg que puede detectar el autómata.
 - 6 V en la señal de entrada, ¿cuánto representa en Kg?
 - 6 V en la señal de entrada, ¿a qué número digital del autómata equivale?

PRÁCTICA 8

Simulación de control combinacional con circuitos integrados

OBJETIVO: Simular automatismos realizados con circuitos integrados

COMPETENCIA: A través de la simulación, comprende la técnica de armar circuitos electrónicos para automatización básica.

SOFTWARE DE SIMULACIÓN RECOMENDADO: Proteus y Fritzing.

FUNDAMENTO TEÓRICO: Para poder implementar diseños lógicos digitales se requiere entender qué significan las familias lógicas y sus diferentes principios de operación. Los circuitos integrados hechos con la misma tecnología y características eléctricas comprenden una familia lógica. Entre las familias más comunes están transistor-transistor lógico (TTL), semiconductor complementario metal-óxido (CMOS) y el emisor acoplado lógico (ECL). Los parámetros generales son:

- 1 Nivel lógico, es decir, el rango de niveles de voltaje que puede corresponder a los estados binarios 1 y 0. Para la serie estándar 74XX TTL, el voltaje garantizado para registrarlo como binario 0 está entre 0 y 0,4 V y para el binario 1 es entre 2,4 V y 5,0 V. Para el CMOS los niveles dependen del voltaje alimentado VDD que se emplee, el cual puede ser desde +3 V hasta +15V y el voltaje máximo para el lógico 1 es de 0,3 VDD en tanto que el mínimo para el lógico 1 es 0,7 VDD.
- 2 Inmunidad contra el ruido o margen de ruido, es decir, la capacidad del circuito de tolerar ruido sin ocasionar cambios espurios en el voltaje de salida. Para la serie estándar 74XXTTL, el margen de ruido es 0,4 V. Por tanto 0,4 V es el margen que se puede aceptar en las entradas lógico 0 y lógico 1 y que lo registrarán como 0 y como 1. Para el CMOS el margen de ruido depende del voltaje alimentado y es de 0,3 VDD.
- 3 Apertura, es decir, la cantidad de entradas de compuerta que puede controlar una salida de compuerta estándar al tiempo que conserva los niveles LOW o HIGH deseados. Esto se determina por cuánta corriente puede abastecer una compuerta y cuánta se necesita para controlar una. Para una compuerta TTL estándar la apertura es de 10, para el CMOS es de 50 y de 25 para el ECL. Si hay más compuertas conectadas a la compuerta controladora, entonces no abastecerá la corriente suficiente para controlarlas.
- 4 Abastecimiento de corriente o acción de disminución de corriente, es decir, cómo fluye la corriente entre la salida de una compuerta lógica y la entrada de otra. Para que una compuerta conduzca a otra con abastecimiento de corriente, la compuerta controladora en alta suministra una corriente para la entrada de la siguiente compuerta. Pero con la disminución de corriente la compuerta controladora en baja recibe una corriente de vuelta desde la compuerta controladora. Las compuertas TTL operan con este tipo de corriente.

5 Tiempo de retardo de propagación, es decir, qué tan pronto responde un circuito digital ante un cambio a nivel de entrada. Por lo común las compuertas TTL se retardan de 2 a 40 ns, lo que hace por lo general alrededor de 5 a 10 veces más rápido que las compuertas CMOS, pero más lento que las compuertas ECL, las cuales suelen tener retardos de propagación de 2 ns.

6 Consumo de potencia, es decir, la cantidad de potencia que la compuerta lógica consumirá a partir del abastecimiento de potencia. La TTL consume unos 10 mW por compuerta en tanto que el CMOS no consume potencia a menos que esté en punto de interruptor. El ECL consume de 15 a 60 mW por compuerta.

Los principales criterios implicados en la determinación de qué familia lógica utilizar son el retardo de propagación y el consumo de potencia. La ventaja mayor del CMOS sobre el TTL es el menor consumo de potencia, lo que lo hace ideal para equipos operados por baterías. Es posible que circuitos integrados por familias lógicas diferentes se puedan conectar entre sí, pero se deben utilizar técnicas especiales de interfaz.

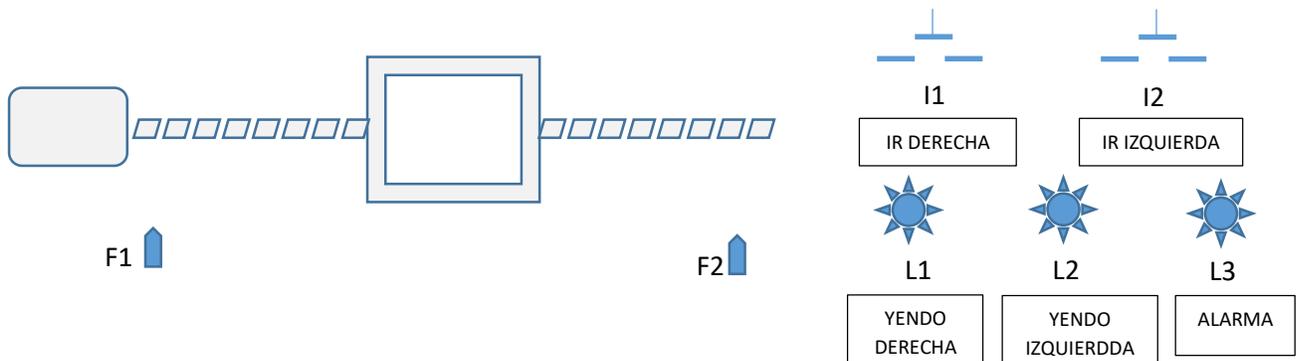
La familia TTL se utiliza ampliamente y se ha de identificar como la serie 74XX. Hay multiplicidad de formas; una de las típicas el TTL estándar es el 7400 cuya disipación de potencia es de 10 mW y un retardo de propagación de 10 ns. El TTL Schottky de baja potencia (LS) es el 74LS00 con una disipación de potencia de 2 mW y el mismo tiempo de retardo de propagación. El TTL Schottky de baja potencia avanzado (ALS) es el 74ALS00 y es más rápido y disipa inclusive potencias más bajas; el retardo de propagación llega a 4 ns y la disipación de potencia es de 1 mW. El rápido TTL(F) es el 74F00 y tiene un retardo de propagación de 3 ns y una disipación de potencia de 6 mW.

La familia del CMOS incluye la serie de lo 4000, la cual tenía la ventaja de la disipación de baja potencia sobre la serie TTS, pero por desgracia era muy lenta. La serie 40H00 era más rápida pero todavía más lenta que TTL(LS). La serie 74C00 se desarrolló para que fuera de clavija compatible con la familia TTL, utilizando el mismo sistema de numeración pero empezando con 74C. Aun cuando tenía una ventaja de potencia sobre la familia TTL, aún así era más lenta. Las 74HC00 y 74HCT00 son más rápidas y sus velocidades son comparables con las de la serie TTL(LS).

Extraído de Mecatrónica de Bolton

Planteamiento del experimento:

Dado un motor de inducción (jaula de ardilla) trifásico que se emplea para controlar la siguiente modalidad de operación:



- Al pulsar I1, la mesa recorre a la derecha mientras se pulsa I1, mientras la mesa avanza se enciende lámpara L1.
- Al pulsar I2, la mesa recorre a la izquierda mientras se pulsa I2, mientras la mesa avanza se enciende lámpara L2.
- F1 es un final de carrera que debe inhibir que la mesa siga avanzando a la izquierda aunque se pulse I2, si se da el caso que al estar activado F1 y se pulsa I2, debe encenderse la lámpara L3.
- F2 es un final de carrera que debe inhibir que la mesa siga avanzando a la derecha aunque se pulse I1, si se da el caso que al estar activado F2 y se pulsa I1, debe encenderse la lámpara L3.
- Si se pulsan simultáneamente I1 e I2 también se enciende la lámpara L3.

ACTIVIDADES A REALIZAR:

1. Diseñe los circuitos de control con compuertas lógicas para funcionamiento de cada una de las tres lámparas y simule su operación en Proteus y circuitos integrados de tecnología TTL.
2. Suponiendo a los finales de carrera F1 y F2 así como a I1 e I2 ideales, es decir que no producen rebote, dibuje con Proteus cómo serían los circuitos interfaz para las señales de entrada (Debe dar un alto o un bajo).
3. Dibuje con Proteus los circuitos interfaz de las señales de salida, (los circuitos integrados no pueden brindar más de 30 mA y 5V), empleando transistores y relés para accionar las lámparas y contactores del motor suponiendo que éstos son de 220 VCA.
4. Dibuje en Fritzing el circuito de control así como las interfaces de entrada y salida, tal como lo haría desarrollando un prototipo empleando un protoboard.
5. Investigue y dibuje con Proteus la forma de conectar los finales de carrera de dos y tres hilos, además de los pulsadores de manera de eliminar rebote.

PRÁCTICA 9

Simulación con circuitos integrados bi-estables

OBJETIVO: Simular secuencias básicas realizadas con circuitos integrados bi-estables

COMPETENCIA: Comprende la técnica de armar circuitos secuenciales básicos con circuitos integrados bi-estables y de corrimiento.

SOFTWARE DE SIMULACIÓN RECOMENDADO: Proteus.

FUNDAMENTO TEÓRICO

Biestables disparados por flancos o flip-flops.

En muchas situaciones es necesario sincronizar el funcionamiento de muchos circuitos diferentes y resulta de utilidad poder controlar el momento en el que un circuito cambiará de estado.

Algunos biestables están contruidos de manera que sólo cambian de estado ante la aplicación de una señal de disparo, en concreto ante el flanco de bajada o de subida de una señal de entrada llamada reloj (CLK). Estos biestables reciben el nombre de biestables disparados por flanco, o más comúnmente flip-flops.

Los flip-flops son dispositivos síncronos. El término síncrono significa que la salida cambia de estado únicamente en un instante específico de una entrada de disparo (reloj), es decir, los cambios en la salida se producen sincronizadamente con el reloj.

Podemos encontrar dos tipos de flip-flops:

1. Los que son disparados por el flanco de subida de la señal de reloj.
2. Los que son disparados por el flanco de bajada de la señal de reloj.

Flip-flops S-R disparado por flanco.

Se asemeja al circuito R-S excepto en que el circuito sólo responde a sus entradas en el flanco ascendente o descendente de la señal de reloj. Los símbolos gráficos (figura1) se asemejan a los de los circuitos R-S con entrada de habilitación, excepto en que esta última entrada se reemplaza por una entrada de reloj.

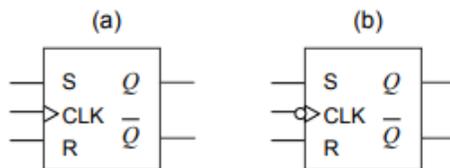


Figura 1: Flip-flop R-S (a) disparado por flanco de subida. (b) disparado por flanco de bajada.

En ausencia de la transición de reloj el flip-flop permanece en su modo de memoria, como se aprecia en el diagrama de la Figura 2, correspondiente a un flip-flop disparado con flanco de subida.

El funcionamiento de un flip-flop R-S activado por flanco descendente es, por supuesto, idéntico, excepto que el disparo tiene lugar en el flanco de bajada de la señal de reloj (cuando cambia de '1' a '0').

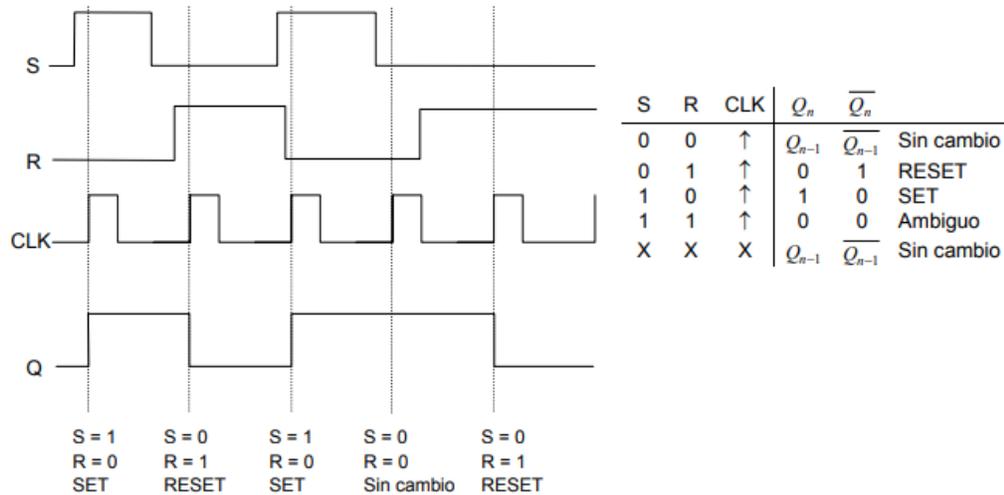


Figura 2: Ejemplo de formas de onda en las entradas y en la salida de un flip-flop disparado por flanco ascendente.

Flip-flop D disparado por flanco.

Su comportamiento es similar al del latch D descrito con anterioridad, la salida del flipflop tipo D se igualará a la entrada en el instante en el que se produzca el flanco ascendente o descendente (según el tipo de flip-flop) de la señal de reloj (CLK). En la Figura 3 se observa el símbolo lógico y la tabla de verdad de un flip-flop tipo D disparado por flanco ascendente.

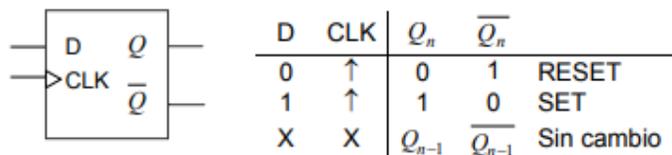


Figura 3: Flip-flop D disparado por flanco ascendente.

El funcionamiento de un flip-flop D disparado por flanco ascendente se resume en la Figura 4.

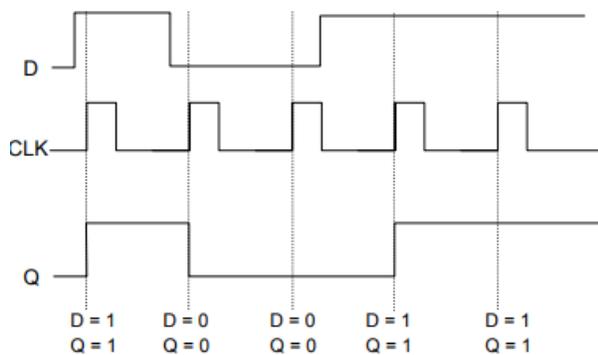


Figura 4: Ejemplo de formas de onda en las entradas y en la salida de un flip-flop D disparado por flanco ascendente.

https://www.infor.uva.es/~jialvarez/asignaturas/fundamentos/lectures/digital/Tema3_secuenciales.pdf

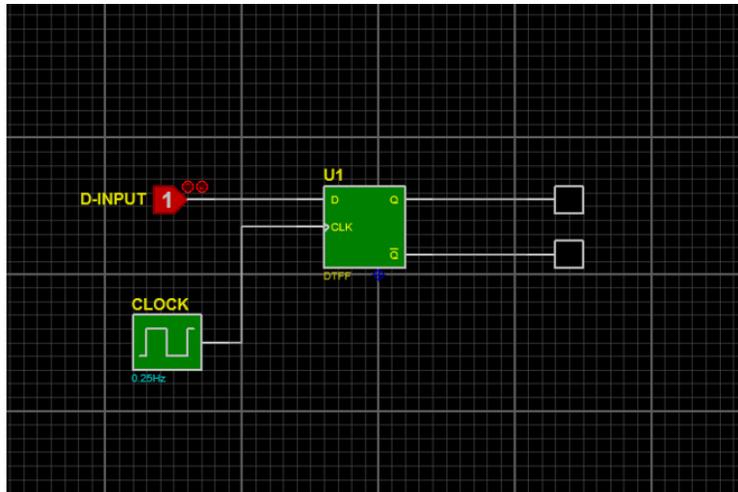


Figura 5

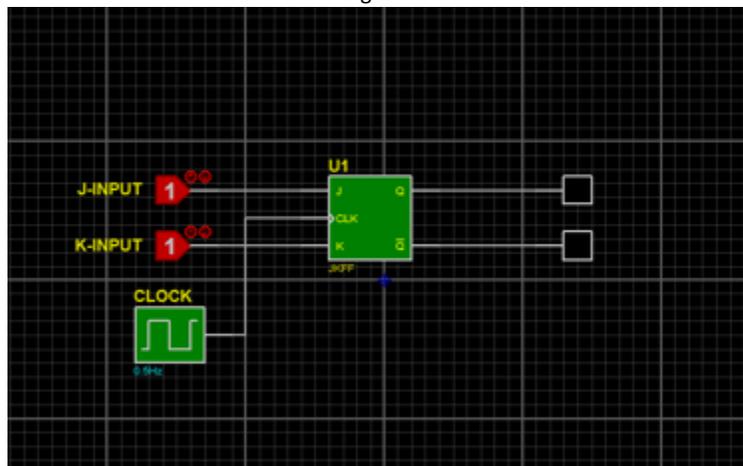


Figura 6

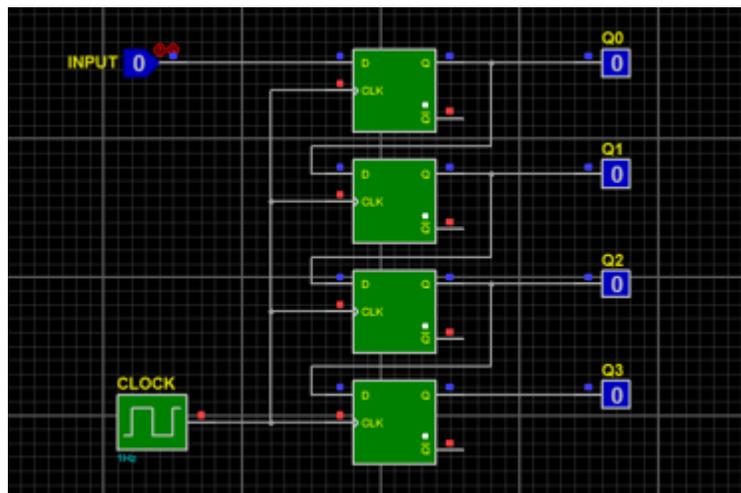


Figura 7

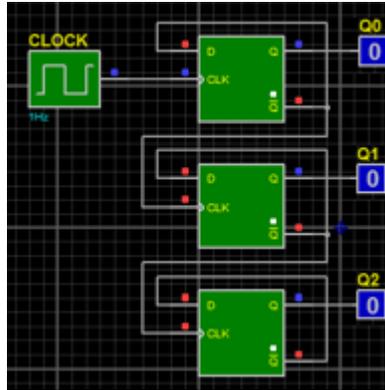


Figura 8

ACTIVIDADES

1. Simular el circuito de la figura 5 y describir su funcionamiento, explicar el efecto cambiando la frecuencia de clock.
2. Simular el circuito de la figura 6 y describir su funcionamiento, indicar la diferencia entre este circuito flip flop J - K y el flip flop S – R.
3. Simular el circuito de la figura 7 que funciona como un “shift register”, investigue sobre shift registers, mencione algunos de ellos e indique en qué los aplicaría.
4. Simular el circuito de la figura 8, realice un diagrama de las señales del reloj, Q1, Q2 y Q3 e indicar la utilidad que podría darle.
5. Utilizando flip flops, genere secuencia para operación continua de un motor paso a paso monopolar.
 - a) Paso simple
 - b) Paso doble

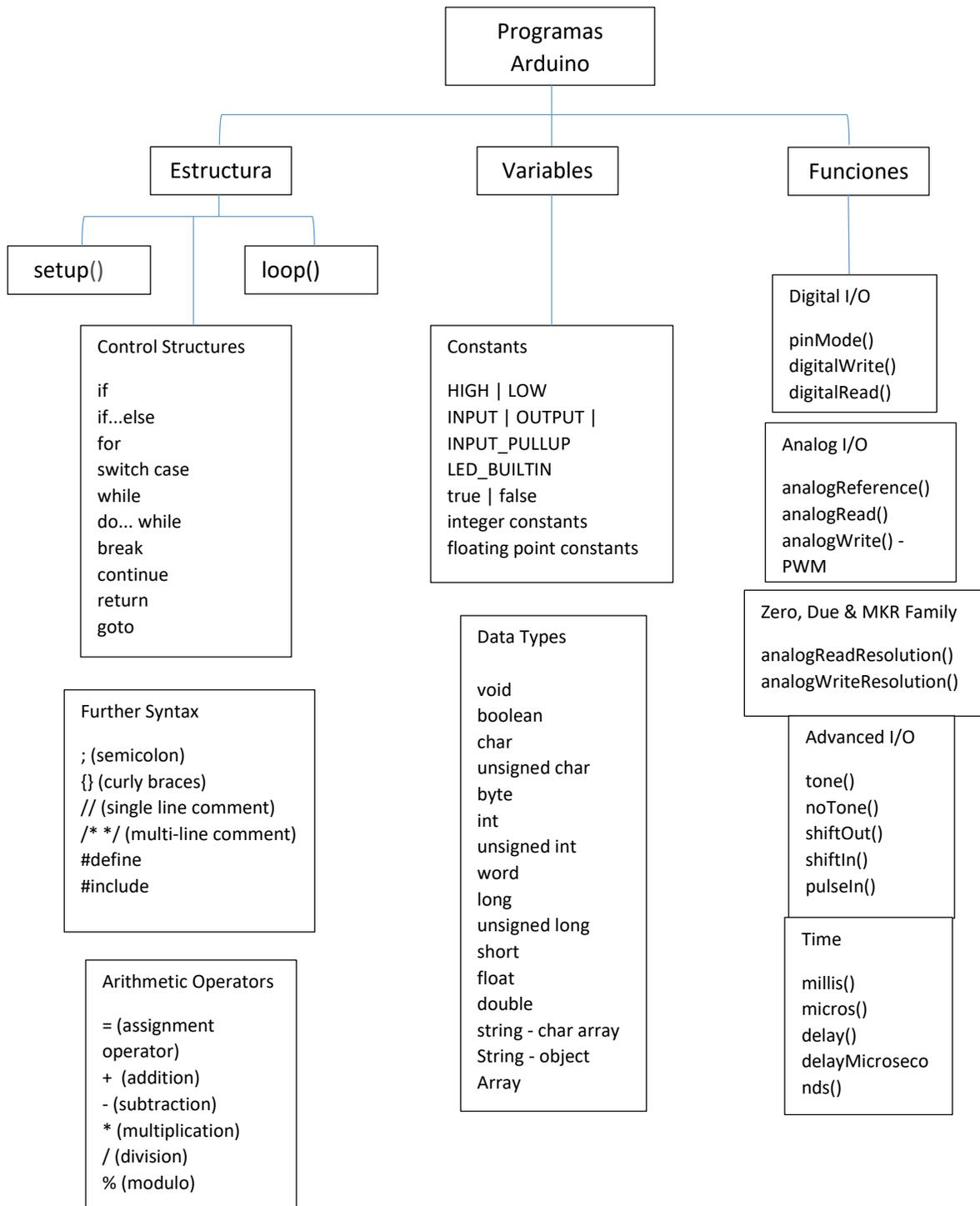
PRÁCTICA 10

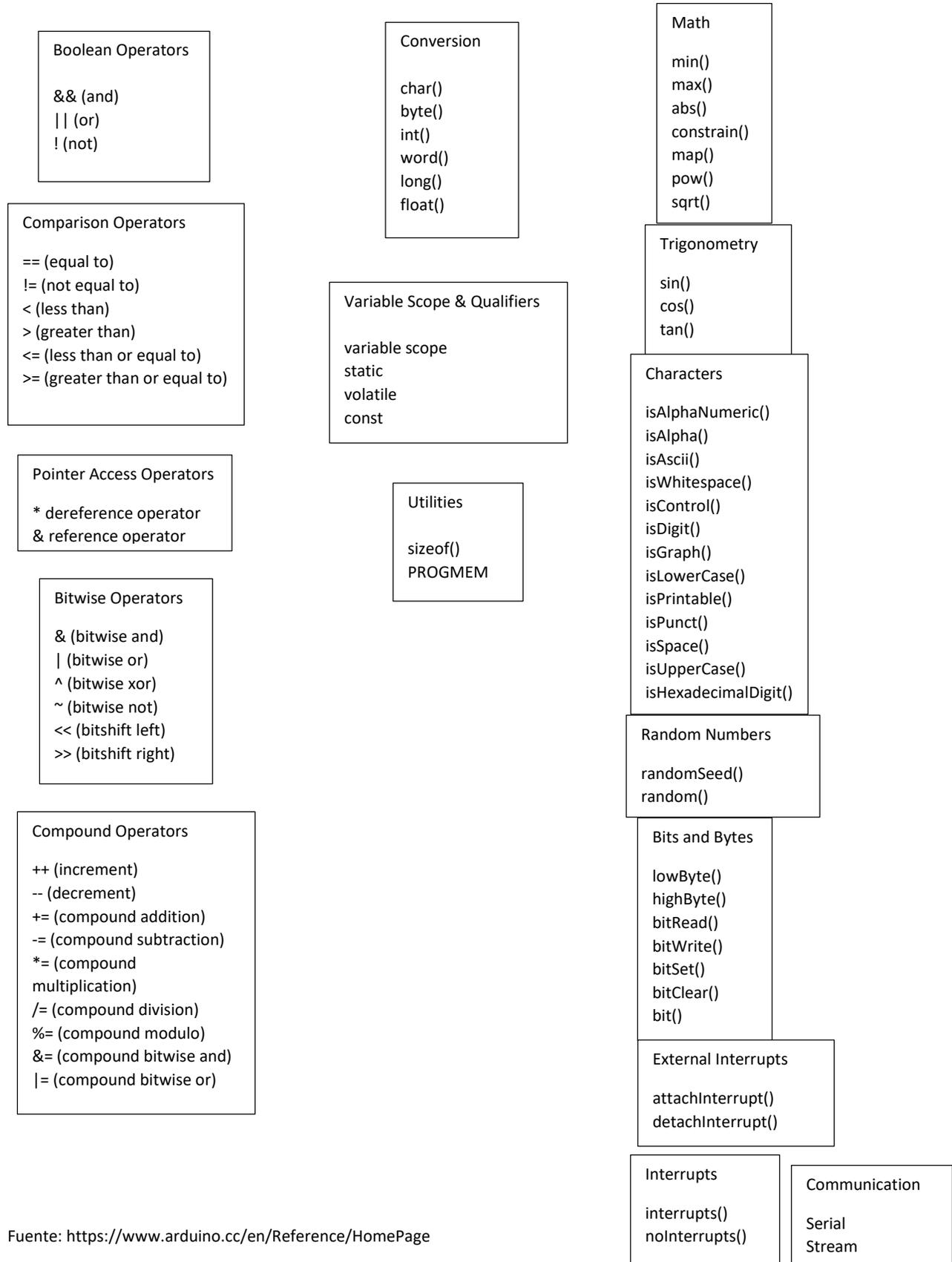
Programas en Arduino

OBJETIVO: Introducción a la programación de microcontroladores Arduino.

COMPETENCIA: Conecta, programa y ejecuta aplicaciones básicas con Arduino.

FUNDAMENTO TEÓRICO:





Fuente: <https://www.arduino.cc/en/Reference/HomePage>

Estructuras de programación

Estructura condicional simple: IF

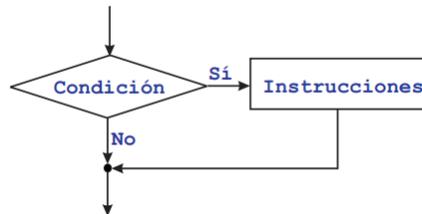
Este es el tipo más sencillo de estructura condicional. Sirve para implementar acciones condicionales del tipo siguiente:

Si se verifica una determinada condición, ejecutar una serie de instrucciones y luego seguir adelante.

Si la condición NO se cumple, NO se ejecutan dichas instrucciones y se sigue adelante.

```

if (Condición)
{
    // Instrucciones
}
    
```



Estructura condicional doble: IF - ELSE

Este tipo de estructura permite implementar condicionales en los que hay dos acciones alternativas:

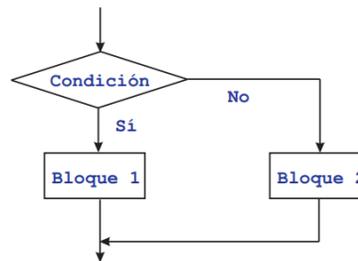
Si se verifica una determinada condición, ejecutar una serie de instrucciones (bloque 1).

Si no, esto es, si la condición NO se verifica, ejecutar otra serie de instrucciones (bloque 2).

En otras palabras, en este tipo de estructuras hay una alternativa: se hace una cosa o se hace la otra. En ambos casos, se sigue por la instrucción siguiente a la estructura IF - ELSE.

```

if (Condición)
{
    // Bloque 1
}
else
{
    // Bloque 2
}
    
```



Estructura condicional múltiple: IF - ELSEIF - ELSE

En su forma más general, la estructura IF - ELSEIF - ELSE permite implementar condicionales más complicados, en los que se “encadenan” condiciones en la forma siguiente:

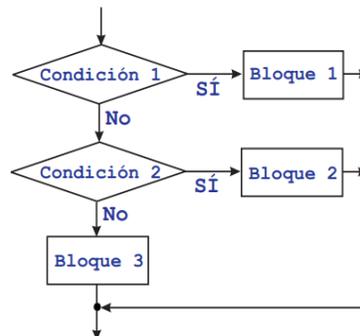
Si se verifica la condición 1, ejecutar las instrucciones del bloque 1.

Si no se verifica la condición 1, pero Sí se verifica la condición 2, ejecutar las instrucciones del bloque 2.

Si no, esto es, si no se ha verificado ninguna de las condiciones anteriores, ejecutar las instrucciones del bloque 3.

```

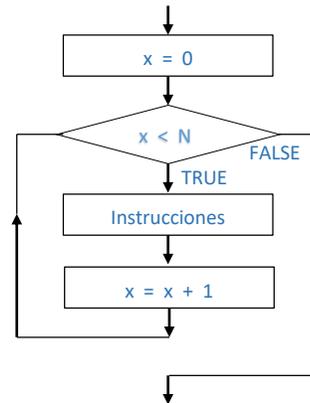
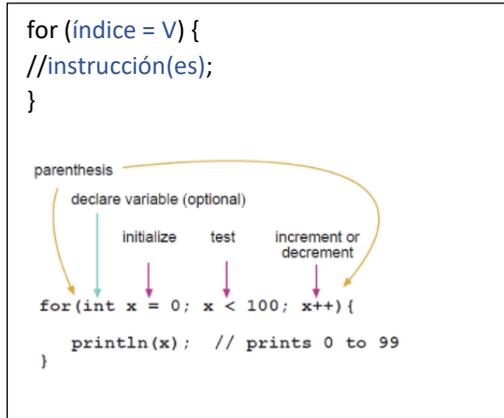
if (Condición 1)
{
    // Bloque 1
}
else if (Condición 2)
{
    // Bloque 2
}
else
{
    // Bloque 3
}
    
```



Estructura de repetición indexada: FOR

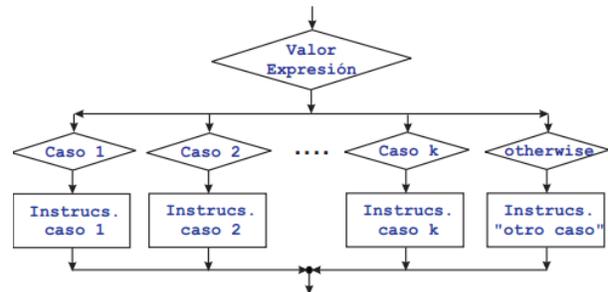
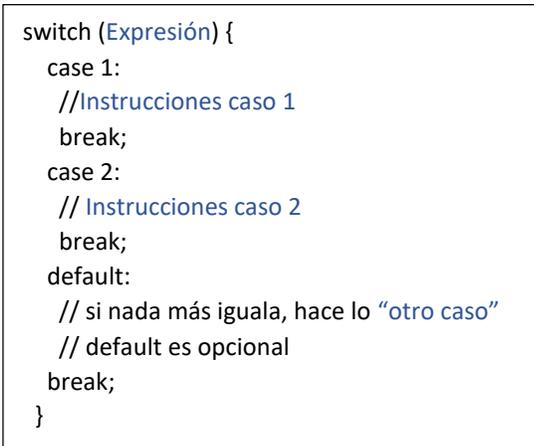
Este tipo de estructura permite implementar la repetición de un cierto conjunto de instrucciones un número pre-determinado de veces.

Para ello se utiliza una variable de control del bucle, llamada también índice, que va recorriendo un conjunto pre-fijado de valores en un orden determinado. Para cada valor del índice en dicho conjunto, se ejecuta una vez el mismo conjunto de instrucciones.



Estructura de elección entre varios casos: SWITCH

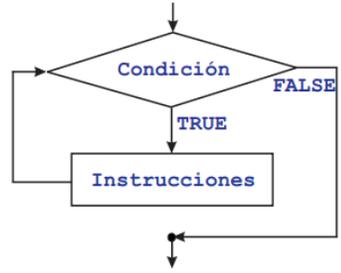
Este tipo de estructura permite decidir entre varios caminos posibles, en función del valor que tome una determinada instrucción.



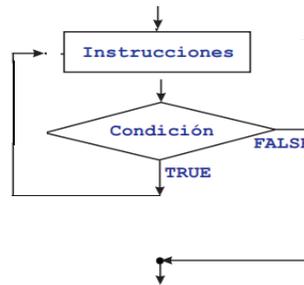
Estructura repetitiva condicional: WHILE

Permite implementar la repetición de un mismo conjunto de instrucciones mientras que se verifique una determinada condición: el número de veces que se repetirá el ciclo no está definido a priori.

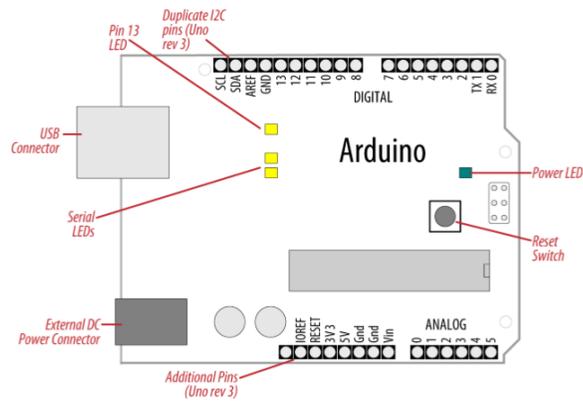
```
while(Condición){
  // Instrucción(es)
}
```



```
do
{
  // Instrucciones
} while (condición);
```



ARDUINO UNO



Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

Fig. 1: Características de Arduino UNO

Análisis de entradas y salidas de Arduino UNO

Cada uno de los 20 pines digitales de Arduino puede ser usado como entrada o salida, usando las funciones adecuadas. Todos los pines funcionan a 5 V, por lo que si se conecta a una tensión mayor puede que dejen de funcionar permanentemente. Cada pin tiene una salida de intensidad de 40 mA máximo y una resistencia de pull-up integrada de unos 20-50 kOhms.

Además algunos pines tienen alguna función adicional:

Serie: 0 (RX) y 1 (TX). Se usan para recibir y transmitir datos serializados con niveles TTL. Estos pines están conectados al conversor USB-Serie para comunicarse con el ordenador. Si en algún proyecto se utiliza esta comunicación, sus pines digitales no estarán disponibles.

Interrupciones externas: 2 y 3, pueden ser configurados para interrumpir la ejecución de código y realizar otra tarea. Pueden ser configurados para interrumpir por nivel, por flanco o por cambio de valor.

PWM: Seis pines marcados en la placa con el símbolo ~, pueden proveer una salida de PWM con una resolución de 8 bits (0 a 255) y con una frecuencia en el orden de kHz.

Entradas Analógicas: Pines A0 a A5, con resolución de 2^{10} (0 a 1023).

SPI: Protocolo de comunicación serie mediante 4 hilos muy utilizado para comunicación de corto alcance entre microcontroladores. Suele ser utilizado también para programar el microcontrolador Atmel mediante un programador compatible.

I2C: Protocolo de comunicación serie mediante 2 hilos muy utilizado para la comunicación de corto alcance con sensores inteligentes.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples>

MATERIAL:

- Computadora personal con Software Arduino (IDE), mismo que puede bajarse libremente de <https://www.arduino.cc/en/Main/Software>
- Arduino UNO con conector de comunicación UART.
- PROTEUS con módulo Arduino

ACTIVIDADES:

Compruebe funcionamiento haciendo uso de simulador proteus o empleando una tarjeta de Arduino (UNO).

Para todos los casos, usted escoja si va a emplear conexión de entrada pull-down o pull-up.

1. Hacer un programa para que el led 13 se encienda mientras se pulse el pin 2.
2. Hacer un programa para que led 13 parpadee 6 veces a una frecuencia de 1Hz y luego el mismo led parpadee 3 veces a una frecuencia de 0,5 Hz y así sucesivamente hasta apagar Arduino.
3. Hacer un programa para que led 13 parpadee sólo tres veces si damos un pulso al pin 3, sólo cuatro veces si damos un pulso al pin 4.

4. Hacer un programa para que el led 13 se encienda con solo pulsar como entrada el pin 5 y se mantenga encendido aunque dejemos de pulsar el pin 5, pero se apague si se pulsa el pin 6.
5. Dibujen con proteus las conexiones físicas para los ejercicios anteriores.
 - a) Sin eliminar rebote “debounce”
 - b) Eliminando rebote “debounce” por hardware.

PRÁCTICA 11

Introducción Arduino (UNO) entradas-salidas digitales y conexión interfaz

OBJETIVO: Introducción a la conexión y programación de entradas/salidas digitales del microcontrolador Arduino.

COMPETENCIA: Conecta, programa y ejecuta aplicaciones básicas con entradas y salidas de microcontrolador Arduino.

FUNDAMENTO TEÓRICO:

Señales de entrada discreta todo o nada:

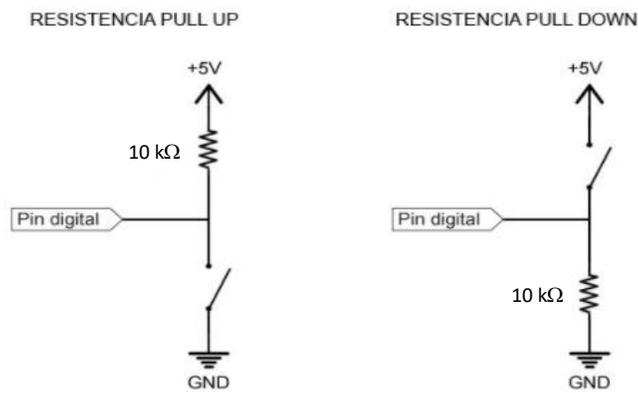


Fig.1: Modos de conexión entradas discretas a Arduino

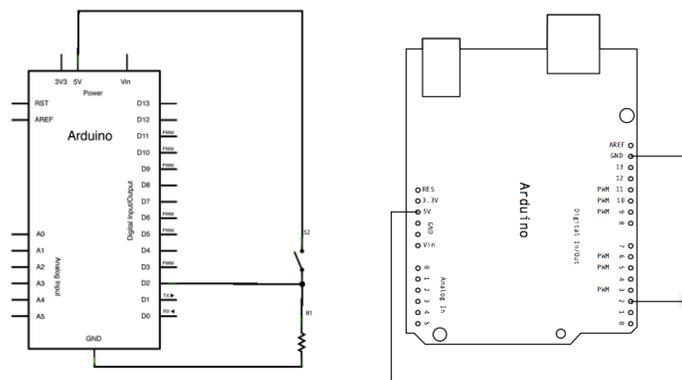


Fig. 2: Entradas "pull down" en Arduino

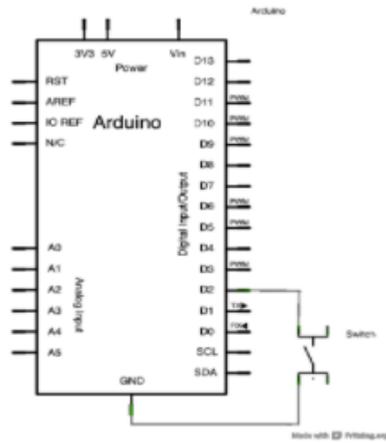


Fig. 3: Conexión Pull Up (resistencia interna Arduino)

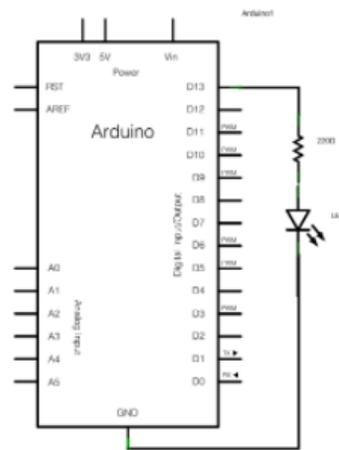
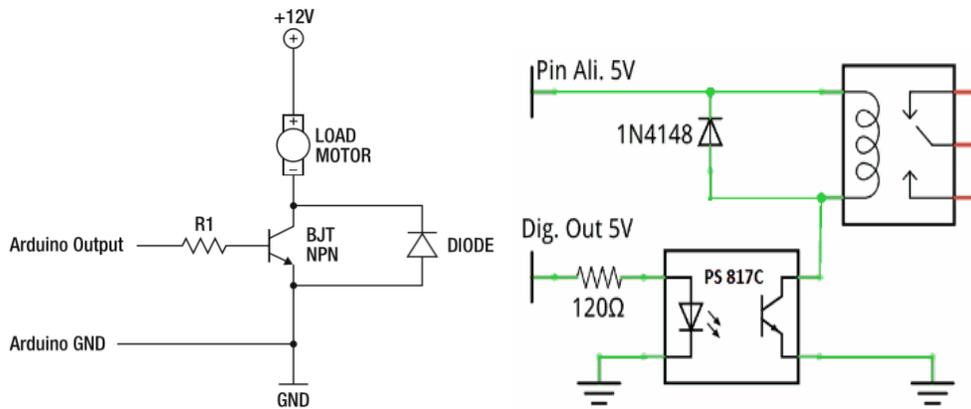


Fig. 4: Conexión de salidas discretas

Interfaz para circuitos de potencia:



(a) Con transistor operando en saturación

(b) Con optoacoplador y relé para potencia

Fig. 2: Circuitos interfaz para potencia

La corriente que puede generar el Arduino es de aproximadamente 20 a 30 mA, insuficientes para aplicaciones que requieran mayores niveles de corriente, sin embargo con un transistor bipolar NPN, como se muestra en la figura 2(a), funcionando en SATURACIÓN, éste se comporta como un conmutador que permite el funcionamiento de por ejemplo, un motor de CC 12V. En vez del motor se puede energizar la bobina de un relé como se ve figura 2 (b), donde además se emplea un optoacoplador para dar aislamiento galvánico.

Con un relé se puede formar parte de un circuito de 220 CA alimentando el contactor de un motor trifásico.

En la práctica debe medirse con un voltímetro la caída de voltaje entre colector y emisor del transistor u optoacoplador para cuando se alimenta su base, éste estará saturado cuando la lectura del voltímetro se hace próximo a cero.

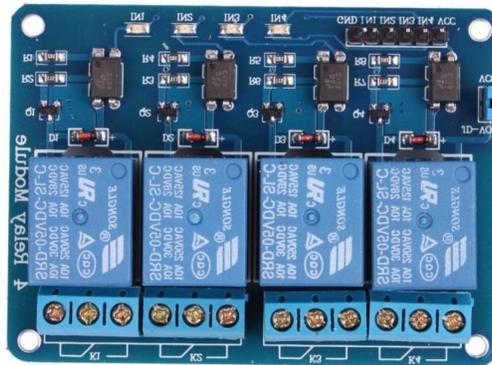


Figura 3: Módulo de relés para Arduino

También están disponibles en el mercado módulos para Arduino con 1, 2, 4 (como figura 3) y 8 relés para fácil conexión a Arduino.

OBSERVACIONES Y ACTIVIDADES A REALIZAR EN LA PRÁCTICA:

El docente le planteará un programa para generar algún tipo de relación digital entrada/salida con Arduino.

Aplique alguno de los circuitos interfaz, por ejemplo los de las figuras 2 para energizar con una salida de Arduino, algún elemento de potencia como un contactor.

Mida caída de voltaje entre colector y emisor del transistor para verificar si el mismo está saturado.

CUESTIONARIO:

1. Dibuje el circuito armado en laboratorio.
2. Investigue otros diagramas de circuitos interfaz y dibújelos.
3. Indique las diferencias de conexión “pull up” y “pull down” en Arduino.
4. Explique en pocas palabras qué entiende por saturación de un transistor.
5. Indique cómo escoge el valor de R1 en el circuito de la figura 2 (a).

PRÁCTICA 12

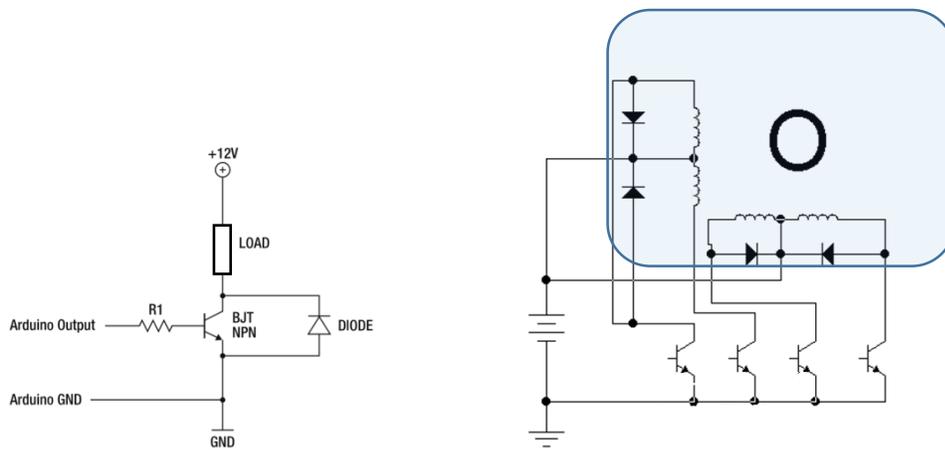
Control de motores paso a paso con Arduino

OBJETIVOS: Simulación de operación de motores paso a paso con Arduino

ACTIVIDADES

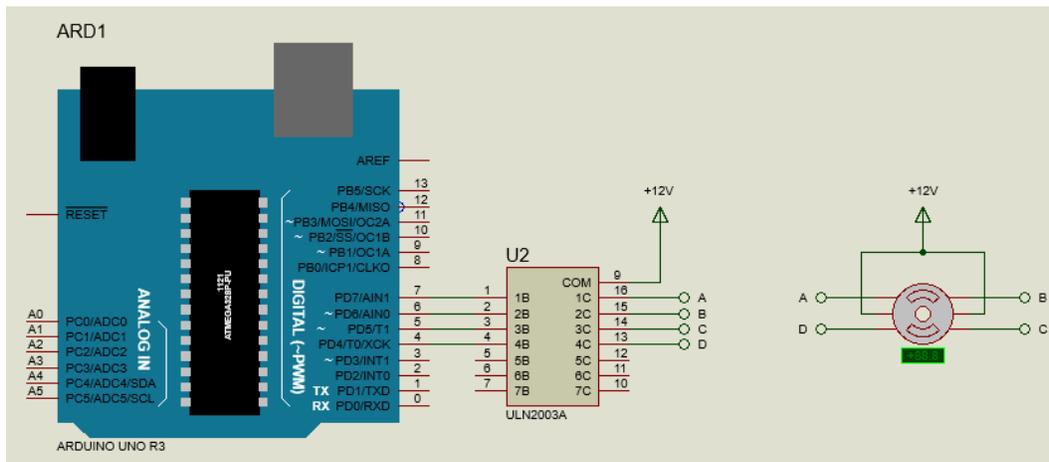
1.- Simular en proteus control con arduino de un motor paso a paso monopolar 12 V y 7,5 grados por pulso para operación continua con un pulso por segundo.

A través de transistores bipolares como interfaz entre Arduino y el motor.



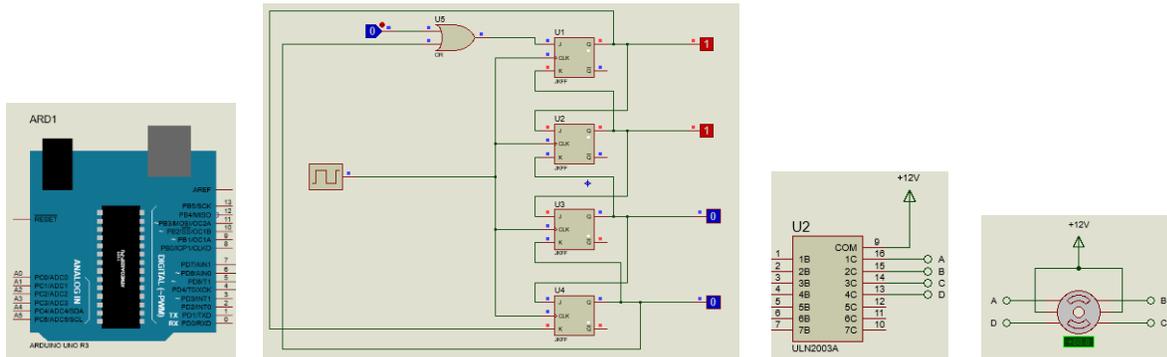
2.- Simular en proteus control con arduino de un motor paso a paso monopolar 12 V y 7,5 grados por pulso para operación continua con un pulso por segundo.

A través de un buffer ULN2003A como interfaz entre Arduino y el motor.



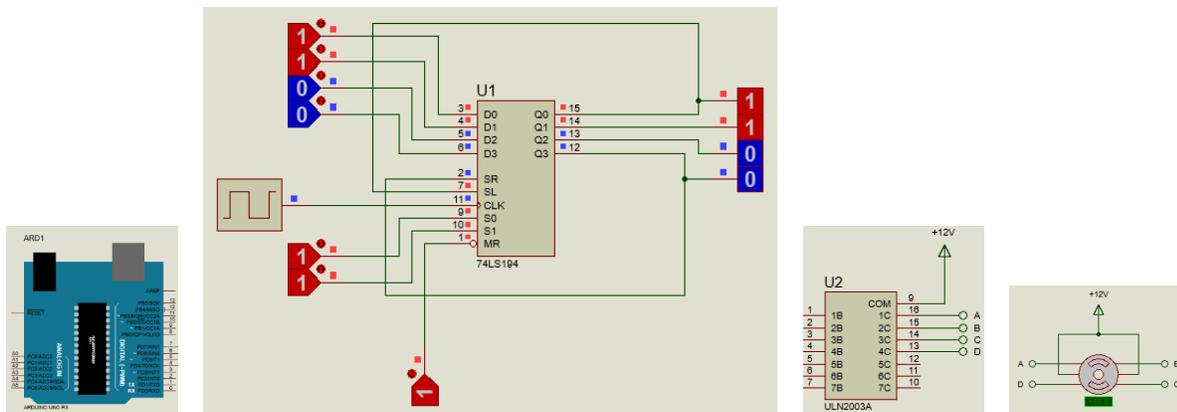
3.- Simular en proteus control con arduino de un motor paso a paso monopolar 12 V y 7,5 grados por pulso para operación continua con un pulso por segundo.

A través de un circuito de corrimiento hecho con flip flops J-K y un buffer ULN2003A como interfaz entre Arduino y el motor.



4.- Simular en proteus control con arduino de un motor paso a paso monopolar 12 V y 7,5 grados por pulso para operación continua con un pulso por segundo.

A través de un circuito de corrimiento 74LS 194 y un buffer ULN2003A como interfaz entre Arduino y el motor.



5.- Escoja cualquiera de las anteriores configuraciones para hacer que un motor paso a paso gire una vuelta completa en un sentido y luego media vuelta en el otro.

Para todas las preguntas:

- Presentar el programa Arduino. PDF
- Presentar el circuito en Proteus. PDF
- Indicar si están aplicando control de paso simple o doble. PDF

APOYO SUGERIDO:

Tutorial elaborado por César Mendoza (antiguo auxiliar de la asignatura)

<https://www.youtube.com/watch?v=55C0tLhX6nc&feature=youtu.be>

Tutorial para uso del circuito integrado “shift register” 74194 (registro de corrimiento o desplazamiento)

<https://www.youtube.com/watch?v=qrfR7KQAUQo>

PRÁCTICA 13

Proyectos con Arduino

OBJETIVO: Aprendizaje en la realización de proyectos mecatrónicos con Arduino, caso vehículo TRONIC

COMPETENCIA: Concibe y proyecta la implementación de un proyecto mecatrónico en base a experiencias realizadas en cursos anteriores.

FUNDAMENTO TEÓRICO: Remitirse a la Guía oficial de la asignatura. Específicamente a la Práctica Opcional – A.- Aplicación de Arduino y “shields” en proyecto TRONIC.

MATERIAL:

- Computadora personal con Software Arduino (IDE), mismo que puede bajarse libremente de <https://www.arduino.cc/en/Main/Software>
- PROTEUS con módulo Arduino.
- APPINVENTOR de MIT
- Arduino UNO con conector de comunicación UART. (opcional)

ACTIVIDADES:

Actividad 1: Hacer un resumen de 5 a 10 líneas por video, de los siguientes videos:

- PRINCIPIO DE FUNCIONAMIENTO MOTOR PROYECTO TRONIC
- Circuito interface motor TRONIC PARTE 1
- Circuito interface motor TRONIC PARTE 2
- Circuito interface motor TRONIC PARTE 3 FINAL
- Creación de aplicación Android para carro TRONIC (PARTE 1)
- Creación de aplicación Android para carro TRONIC (PARTE 2)
- Creación de aplicación Android para carro TRONIC (PARTE 3) FINAL

Presentación en PDF

Actividad 2: Hacer un diseño de aplicación para celular para controlar Arduino, mediante APPINVENTOR de MIT <http://ai2.appinventor.mit.edu>

- Presentación de captura de imagen del diseño de presentación y diagrama de bloques en PDF (Debe tener algún distintivo de la Carrera o su nombre)

Actividad 3: Desarrollar en Proteus el proyecto Tronic, presentar:

- Conexiones en Proteus PDF (Motor CC, Servomotor, Sensor ultrasonido, HC06 y Leds de señalización.

Actividad 4: Programar Arduino para que Tronic tenga operación autónoma en la evasión de obstáculos en base a su sensor de ultrasonido.

- Programa en IDE de Arduino PDF

Actividad 5: Programar Arduino para que Tronic opere mediante mandos de un celular vía bluetooth.

- Programa en IDE de Arduino PDF

MATERIAL DE TRABAJO

Proyecto TRONIC INTRO

<https://www.youtube.com/watch?v=1zeOwseyTOs&t=5s> (1:10)

PRINCIPIO DE FUNCIONAMIENTO MOTOR PROYECTO TRONIC

<https://www.youtube.com/watch?v=8Yu6Sgy0B98> (7:48)

Circuito interface motor TRONIC PARTE 1

<https://www.youtube.com/watch?v=zScUjSjFuyw> (5:23)

Circuito interface motor TRONIC PARTE 2

<https://www.youtube.com/watch?v=y2m8wv1EyuA> (7:09)

Circuito interface motor TRONIC PARTE 3 FINAL

<https://www.youtube.com/watch?v=18qiru8VgMw> (6:40)

Creación de aplicación Android para carro TRONIC (PARTE 1)

<https://www.youtube.com/watch?v=6pG6YOOMct4> (7:12)

Creación de aplicación Android para carro TRONIC (PARTE 2)

<https://www.youtube.com/watch?v=Zv8u-isgjkY> (7:12)

Creación de aplicación Android para carro TRONIC (PARTE 3) FINAL

<https://www.youtube.com/watch?v=WXMwvFjnaRM> (8:19)

Librerías de complemento y un programa para generar puertos com virtuales

<https://mega.nz/file/J4VDHALb#PUShzauXjApl2VeT5DB1Ru2CHb79eh5OZITkw0OLM6Y>

https://www.naylampmechatronics.com/blog/12_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html bluetooth

https://www.youtube.com/watch?v=ibI6bhT_Hck video habilitar bluetooth

https://www.youtube.com/channel/UChzziLREtNjpvGR2_m59Gtw Página Mecatrónica

PRÁCTICA 14

Programación básica con LabVIEW

OBJETIVO: Práctica de programación gráfica mediante instrumentos virtuales de LabView

COMPETENCIA: Programa y hace aplicaciones de secuencias en función a tiempo y eventos con myRIO y LabVIEW.

FUNDAMENTO TEÓRICO: Remitirse a la Guía oficial de la asignatura (PRÁCTICA 14).

ACTIVIDADES:

Actividad 1:

Simule la segunda práctica “inversión de giro” realizada en la asignatura haciendo uso de VI Labview, con dos pulsadores virtuales que operen girar en sentido horario y anti horario respectivamente y dos leds virtuales que emulen salida a los contactores correspondientes. Emplee acción mecánica de interruptor para apreciar qué sucede si pulsa simultáneamente ambos botones.

Actividad 2:

Programa la simulación de secuencia de un motor paso a paso monopolar en modalidad de medio paso (half step). Use leds virtuales para emular a las fases del motor.

Actividad 3:

Programa la siguiente secuencia de parpadeo de un led virtual, se inicia con cinco parpadeos de 1Hz, luego cinco parpadeos de 2 Hz, se repite el proceso cinco veces y se apaga.

Actividad 4:

Programa la secuencia de arranque de dos motores simulándolos con leds virtuales, un botón virtual para que arranque el primer motor, un segundo botón virtual para arrancar al otro motor sin parar al primero y un tercer botón virtual para parar en cualquier momento.

Actividad 5:

Simule arranque estrella triángulo con un botón virtual arranca encendiendo los leds virtuales que representan a los contactores de este tipo de arranque y una vez funcionando en triángulo pueda pararse con otro botón virtual.

NOTA: Para todos los casos, presentar PDF del panel frontal y diagrama de bloques.

PRÁCTICA 15

Integración Arduino (UNO) y LabVIEW, almacenamiento de datos

OBJETIVO: Introducción al manejo de entrada analógica y salida pseudo analógica combinando tarjeta Arduino con programas LabVIEW.

COMPETENCIA: Programa y hace aplicaciones con entrada analógica y salida pseudo analógica de una tarjeta Arduino UNO e instrumentos virtuales de LabVIEW.

FUNDAMENTO TEÓRICO: Remitirse a la Guía oficial de la asignatura (PRÁCTICA 15).

ACTIVIDADES:

Actividad 1

Programa una secuencia de parpadeo de un led virtual en paralelo a un led real conectado a una salida de Arduino, de manera que se inicie con cinco parpadeos de 1Hz, luego cinco parpadeos de 2 Hz, se repita este proceso cinco veces y luego se apague.

Actividad 2:

Programa en LabView la secuencia de pulsos (uno por segundo) mediante pines de salidas de Arduino para accionar un motor paso a paso monopolar en modalidad de medio paso (half step). Use leds virtuales para visualizar los pulsos en el panel frontal de LabView.

Actividad 3:

Programa la aplicación “inversión de giro” comandado en paralelo por dos pulsadores conectados a entradas de Arduino con dos botones virtuales del panel frontal de LabView, en el que también debe desplegarse un mensaje indicando si está accionada la salida que se conecta al relé de sentido horario de Arduino o la salida del relé de sentido antihorario de Arduino.

Actividad 4:

Programa la secuencia de arranque de dos motores a través de dos salidas de Arduino desplegándolos con leds virtuales del panel frontal de LabView. En el mismo panel frontal un botón virtual para que arranque el primer motor, un segundo botón virtual para arrancar al otro motor sin parar al primero y un tercer botón virtual para parar en cualquier momento.

Actividad 5:

Programa arranque estrella triángulo, con un pulsador real arranca encendiendo las correspondientes salidas de Arduino que activan los relés de la bobina principal, la bobina de estrella y bobina de triángulo, mediante leds virtuales visualiza el estado de dichas salidas. Debe haber un pulsador real para parar todo estando en el modo triángulo.

NOTA: Para todos los casos, presentar PDF del panel frontal y diagrama de bloques.

