TEC | Tecnológico de Costa Rica

Escuela de Ingeniería Electromecánica

Ingeniería en Mantenimiento Industrial

Instructivo de Laboratorio

Laboratorio de Control Eléctrico

20 de febrero de 2026
Versión: 0.2

Dr.-Ing. Luis Diego Murillo
Dr.-Ing. Juan J. Rojas

Índice general

Laboratorio 1

Introducción a Arduino

1.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Desarrollar un programa sencillo en la plataforma de Arduino.
- Implementar en C las funciones lógicas AND, OR, XOR, NAND y NOR

1.2. Materiales y equipo

- 1 Arduino UNO MINIMA R4.
- 1 Multímetro.
- 5 Resistencias de 270 o 330 Ω.
- 2 Resistencias de 1 kΩ
- 2 interruptores pulsadores.
- 1 Protoboard.
- 5 Diodos emisor de luz (LEDs).
- 2 Generadores de funciones.
- 2 Osciloscopio.
- 1 Computadora portátil.

1.3. Procedimiento

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

1.4. Práctica en Clase

1.4.1. Actividad 1

El código del apéndice B.1 apaga y enciende un LED según el estado de un botón. La figura 1.1 muestra el esquema básico de conexión.
Elimine el botón y alimente el Arduino con un generador de funciones que brinde una señal cuadrada de 0 a 5 voltios. Conecte los dos canales del osciloscopio en la entra y salida. Recuerde que todas las tierras deben estar conectadas entre si.

Conteste las preguntas:

¿Cuanto es el retardo de la señal? ¿Cual es la frecuencia de la señal de entrada en que la señal de salida se distorsiona? Es decir, la señal de salida no es inversa a la señal de entrada. Recuerde capturar las pantallas del osciloscopio en ambos casos.

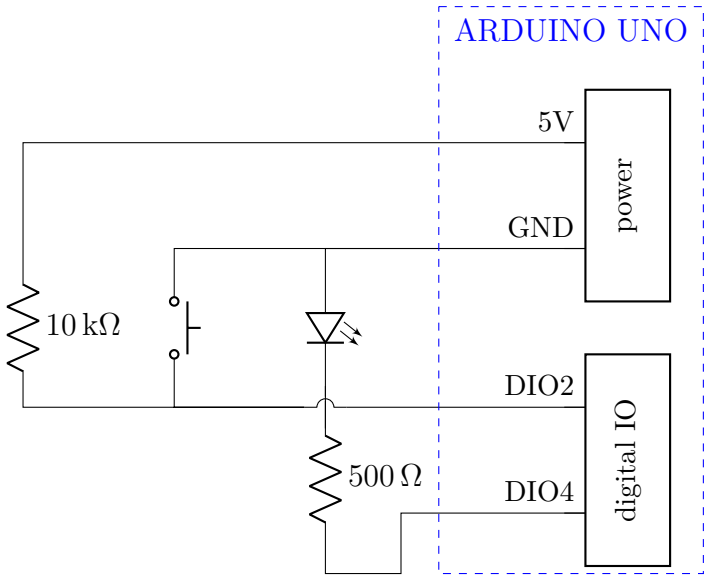


Figura 1.1: Conexión de circuito para Actividad 1

1.4.2. Actividad 2

Esta actividad permite analizar el comportamiento de las conectivas lógicas AND, OR, XOR, NAND, y NOT, y exportar los resultados al puerto serial, así como mostrar los resultados mediante los LEDs conectados a los pines $\{4, 5, 6, 7, 8\}$; el código de ejemplo se muestra en el apéndice B.2. Las entradas declaradas mediante los pines $\{2, 3\}$ reciben las señales de dos generadores de funciones. Los generadores brindan una señal triangular que oscila ente $[0 - 5]$ voltios, debe ajustar cuidadosamente con el osciloscopio cada señal. La señal del pin 2 tendrá una frecuencia del doble del pin 3. Adicionalmente las señales triangulares alimentaran el convertidor Analógico-Digital (ADC) mediante los pines A0 y A1. Conecte su Arduino según el esquema de la Figura 1.2.

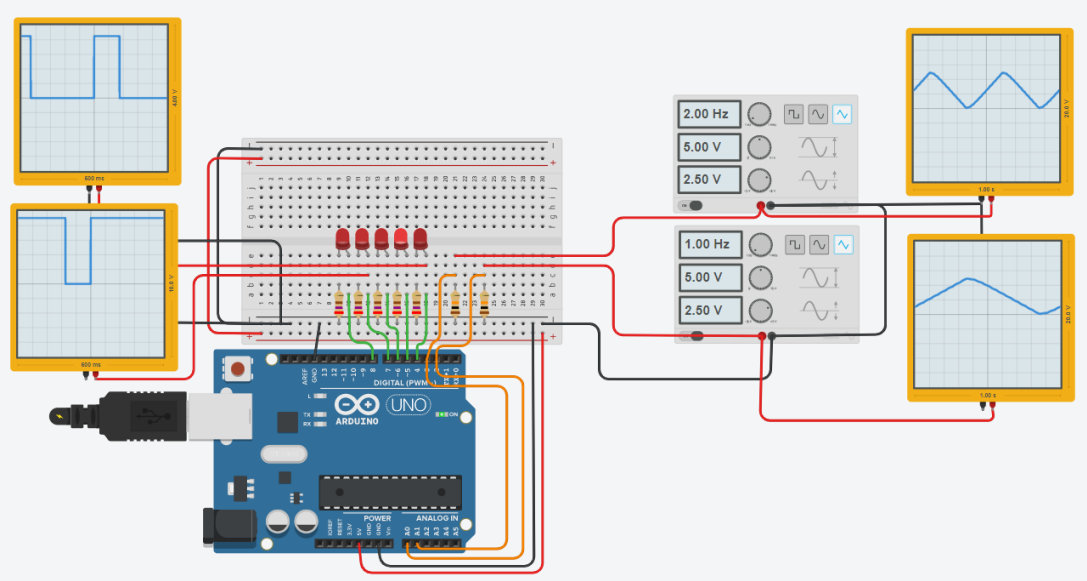


Figura 1.2: Esquema de conexión del Arduino, Generador de funciones y Osciloscopio

El código del Anexo B implementa las funciones lógicas AND, OR. Un repaso de como implementar funciones en C se muestra en este enlace.

Conteste las preguntas:

Guarde los datos del puerto serial en un archivo .TXT e importelos en MS EXCEL para graficar los datos. Se recomienda utilizar el monitor serial de MS CODE STUDIO dado que este permite salvar los datos. ¿Se logra apreciar las señales triangulares y digitales? ¿Las gráficas de las señales digitales de entrada y salida cumplen las tablas de verdad de las conectivas lógicas? ¿Cual es el voltaje de entrada en bajo máximo $V_{IL}max$?, ¿Cual es el voltaje de entrada en alto mínimo $V_{IH}min$?, ¿Cual es el error que presenta las mediciones de los voltajes ?, ¿Si desea un error de ± 1 mV, de cuantos bits debe ser el ADC?¿Explique?

Laboratorio 2

Programación de funciones combinacionales.

2.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Programar funciones booleanas combinacionales en Arduino.
- Verificar la compuertas NAND y NOR son en si mismas un conjunto Universal, equivalentes a el conjunto de conectivas lógicas OR, AND y NOT.
- Verificar que a partir de los mintérminos de una expresión lógica se obtienen los circuitos con compuertas NAND y de los maxtérminos, los circuitos con compuertas NOR.
- Verificar que para obtener la solución mínima de un circuito lógico, es necesario sintetizar los circuitos tanto por mintérminos como por maxtérminos.
- Programar multiplexores 4×1 y 8×1.

2.2. Materiales y equipo

- 1 Arduino UNO o equivalente: MEGA o ESP32.
- 1 Multímetro.
- 10 Resistencias de 270 o 330 Ω.
- 5 Resistencias de 1kΩ.
- 10 interruptores pulsadores.
- 1 Protoboard.
- 10 Diodes emisor de luz (LEDs).
- 1 Computadora portátil.

2.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

2.4. Práctica en Clase

2.4.1. Actividad 1

Programe en Arruino una lógica combinacional que resuelva el siguiente problema:

- En una planta industrial se desea automatizar un motor y una alarma de acuerdo a 4 sensores llamados a, b, c y d
- La señal del motor se representa por la función lógica $f1$, y se activará cuando los sensores a y b estén encendidos, o cuando los sensores b y d estén encendidos, o cuando los sensores a, c y d están encendidos.
$$f1 = ab + bd + acd$$
- La señal de alarma se representa con la función lógica $f2$, y se activará siempre que estén todos los sensores apagados excepto a o todos apagados excepto c o estén encendidos c y d y el resto apagados. También la alarma se enciende cuando todos los sensores están encendidos o cuando están todos encendidos excepto d o todos encendidos excepto c.
$$f2 = a\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + abcd + abc\bar{d} + ab\bar{c}d$$
$$f2 = a\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c + abc + abd$$
- Los pines $\{2, 3, 4, 5\}$ serán las entradas digitales $\{a, b, c, d\}$.
- Los pines $\{6, 7, 8\}$ se reservan para $f1$. Se deben implementar tres funciones: la suma de productos (SDP), el producto de sumas (PDS) y la función resuelta por conectivas $NAND/NAND$.
$$f1_{SDP} = ab + bd + acd$$
$$f1_{PDS} = (a + b) \cdot (a + d) \cdot (b + d) \cdot (b + c)$$
$$f1_{NAND/NAND} = \overline{\overline{ab} \cdot \overline{bd} \cdot \overline{acd}}$$
- Los pines $\{9, 10, 11\}$ se reservan para $f2$. Se deben implementar tres funciones: la suma de productos (SDP), el producto de sumas (PDS) y la función resuelta por conectivas NOR/NOR.
$$f2_{SDP} = a\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c + abc + abd$$
$$f2_{PDS} = (a + c) \cdot (a + \bar{b}) \cdot (b + c + \bar{d}) \cdot (\bar{b} + c + d) \cdot (\bar{a} + b + \bar{c})$$
$$f2_{NOR/NOR} = \overline{\overline{(a + c)} \cdot \overline{(a + \bar{b})} \cdot \overline{(b + c + \bar{d})} \cdot \overline{(\bar{b} + c + d)} \cdot \overline{(\bar{a} + b + \bar{c})}}$$

Conteste las preguntas:

¿Cómo sería el circuito digital de cada una de las funciones implementadas? ¿Como es la tabla de verdad experimental de cada salida versus la tabla teórica?

2.4.2. Actividad 2

Programe las funciones de un Multiplexor 2x1, 4x1 y 8x1. Por otra parte, existe un circuito combinacional que se comporta como la tabla de verdad mostrada en la Tabla 2.1.

Tabla 2.1: Comportamiento esperado de la estructura lógica.

a	b	$F(a,b)$
0	0	$c+d$
0	1	$\overline{c+d}$
1	0	$\overline{c \cdot d}$
1	1	$c \oplus d$

Conteste las preguntas:

¿El comportamiento de un multiplicador 8×1 coincide con la tabla teórica? En la funcion Loop() , llame la función del MUX 8×1 e ingrese parámetros $\{a, b, c, d\}$ de tal forma que se comporte igual que la tabla de verdad presentada en el Cuadro 2.1.¿Tiene el mismo comportamiento? ¿Es posible implementar la T.V. con dos MUX 4 y un MUX 2?¿Como se implementa?

Laboratorio 3

Laboratorio: Arranque-pare sencillo y reversible.

3.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Estudiar las representaciones simbólicas normalizadas de los elementos de control eléctrico, según norma NEMA ICS 19-2002 y IEC 60617.
- Seleccionar contactores y relevadores térmicos.
- Estudiar la representación gráfica del estandar NEMA ICS 19-2002 o IEC 60617.
- Diseñar e implementar circuitos de control para arranque sencillo de un motor trifásico.
- Diseñar e implementar circuitos de control para arranque-pare reversible de un motor trifásico.

3.2. Equipos y materiales

Para este laboratorio de necesitaran:

- 1 motor trifásico 1/2 hp de 3 puntas
- 1 relevador térmico
- 1 botón pulsar cerrado
- 2 botones pulsadores abiertos
- 1 multimetro.

3.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

3.4. Práctica en Clase

3.4.1. Actividad 1

Se desea seleccionar un arrancador térmico para dos motores que mueve una bomba de agua. El primer motor es de 100 hp, trifásico (3∅), voltaje de línea 480 voltios, voltaje de control de 120 voltios. No se requiere gabinete. El segundo motor es de 50 hp, 3∅, 240V, 60Hz, voltaje control 120 V, ambos a plena carga con factor de potencia 0.95 en atraso. Asuma eficiencias del motor de 86 %. Por ejemplo las bombas se usan en promedio 3 veces al día todos los días y se requiere que los contactores duren al menos 20 años.

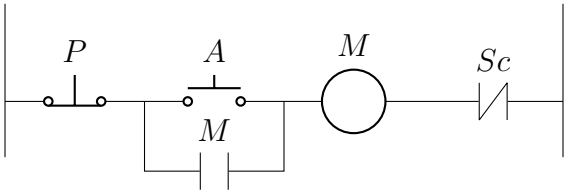
Conteste las preguntas

¿Cual sería para ambos aplicaciones el modelo de arrancador con reveladores electrónicos con falla a tierra para la marca EATON según el catalogo *NEMA Contactors and Starters*, serie Freedom?. ¿Puede explicar porque se usa el mismo contactor para ambos casos?

3.4.2. Actividad 2

Los ingenieros Electricistas, electromecánicos y de Mantenimiento pueden diseñar sus circuitos de control eléctricos, sin embargo cada fabricante posee libros de diagramas estandarizados como el que brinda Schneider con su documento *Wiring Diagram Book*[1].
Estudie toda la simbología NEMA, Tabla 1 y Tabla 2 de la referencia [1] y asocie los con el elementos físicos y su funcionamiento. Deduzca la ecuación lógica del circuito eléctrico de control, para esto use un mapa de Karnaugh o la ecuacion caracteristica del cerrojo S-R prioridad al reset. Realice el alambrado del circuito de control y luego del circuito de potencia del arranque sencillo de un motor trifásico de 1/2 hp.

$$M^+ = \overline{P} \cdot \overline{Sc} \cdot (A + M)$$



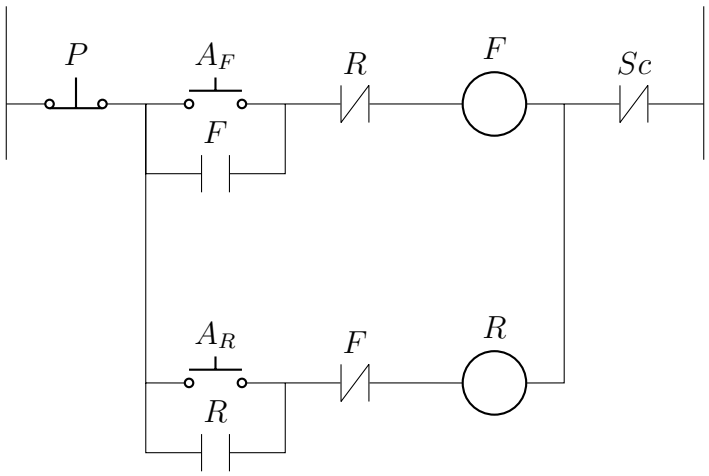
Conteste las preguntas:

¿Es el circuito alambrado control a dos o tres hilos? Explique. ¿El circuito de control funcionó según la tabla de verdad?, Que pasa si se pierde una fase, como se comporta el circuito?. Que diferencia existe entre un arrancador automático y uno manual en el caso que exista un corte de flujo eléctrico? Explique. Hasta ahora los circuitos fueron representados según la norma NEMA, muestre el circuito de control y potencia según la simbología de la norma IEC.

3.4.3. Actividad 3

Deduzca el circuito de control de un arranque-pare-reversible. Realice el alambrado del circuito de control y luego del circuito de potencia del circuito para un trifásico de 1/2 hp.

$$F^+ = \overline{P} \cdot \overline{R} \cdot \overline{Sc} \cdot (A_F + F)$$
$$R^+ = \overline{P} \cdot \overline{F} \cdot \overline{Sc} \cdot (A_R + R)$$



Conteste las preguntas:

¿El circuito de control funcionó según la tabla de verdad?, ¿Que pasa si se presionan los botones pulsadores de arranque hacia la derecha y arranque hacia la izquierda al mismo tiempo? Explique usando el concepto de enclavamiento. Muestre el circuito de control y potencia según la simbología de la norma IEC.

Laboratorio 4

Laboratorio: Arranques a tensión reducida y circuitos temporizados.

4.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Diseñar un circuito secuencial temporizado que permita arrancar un motor en tensión reducida utilizando el método de estrella delta.
- Diseñar un circuito secuencial temporizado que permita arrancar un motor mediante auto-transformador.
- Diseñar un circuito secuencial temporizado que permita arrancar en cascada dos motores y apagarlos en cascada.
- Alambrar el circuito de control y potencia.
- Detectar fallas en circuitos alambrados con lógica cableada.

4.2. Equipos y materiales

Para este laboratorio de necesitaran:

- 2 motor trifásico 1/2 hp de 3 puntas.
- 1 motor trifásico 1/2 hp de 6 puntas.
- 2 relevadores térmico.
- 4 contactores.
- 1 botón pulsar cerrado.
- 2 botones pulsadores abiertos.
- 1 temporizador On-Delay.
- 1 temporizador Off-Delay.
- 1 amperimetro de gancho con medición INRUSH.
- 1 Analizador de señales Fluke 143B

4.2.1. Arranque a tensión reducida

Los arranques a tensión reducida buscan reducir el voltaje que alimenta los devanados estatóricos para que en el momento del arranque la corriente de línea se reduzca.
El arranque por autotranformador, utilizado en motores trifásicos de inducción jaula de árdilla de tres terminales, permite reducir tanto el voltaje de línea como la corriente de linea que consume el motor. Si la relacion de tranformacion es $a = V_{out}/V_{in}$, donde V_{out} es el voltaje hacia el motor y V_{in} es el voltage de la red. La corriente de línea antes del autotransformador es $I_{linea} = a^2 \cdot I_{arr}$, donde I_{arr} es la corriente de arranque a tensión plena. La Figura 4.3 muestra el circuito de potencia y control para el arranque por auto-transformador.
Los motores trifásicos de indución jaula de ardilla de seis puntas se pueden arrancar realizando una reconfiguración de su devanado. Durante el régimen transitorio sus devanados se alambran en estrella ($\curlywedge$) y cuando su velocidad alcanza el 80 % aproximadamente se realiza la re-conexión en delta (Δ) de los devanados. Esto permiten reducir el voltaje en $\sqrt{3}$ y la corriente de arranque se reduce a un tercio de la original. La Figura 4.5 muestra el circuito de potencia para el arranque $\curlywedge - \Delta$.

4.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

4.4. Práctica en Clase

4.4.1. Actividad 1

Realice un arranque-pare secuencial para dos motores trifásicos de tres puntas. Mida la corriente de arranque y la tensión de alimentación del motor.

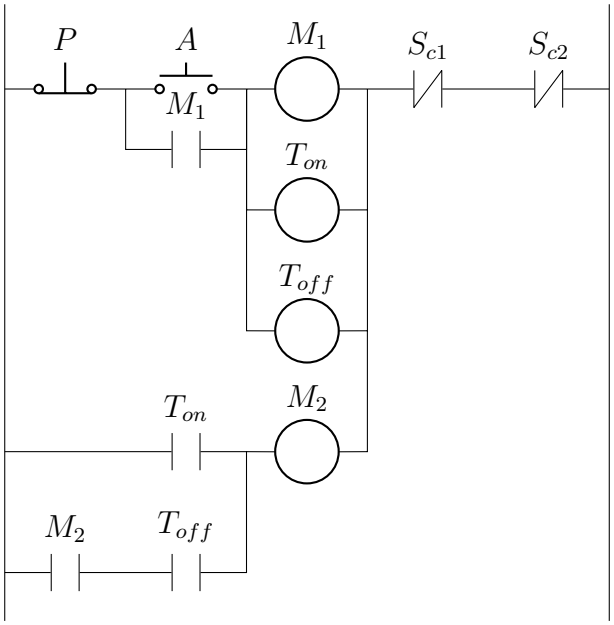


Figura 4.1: Circuito de control: arranque-pare secuencial de dos motores

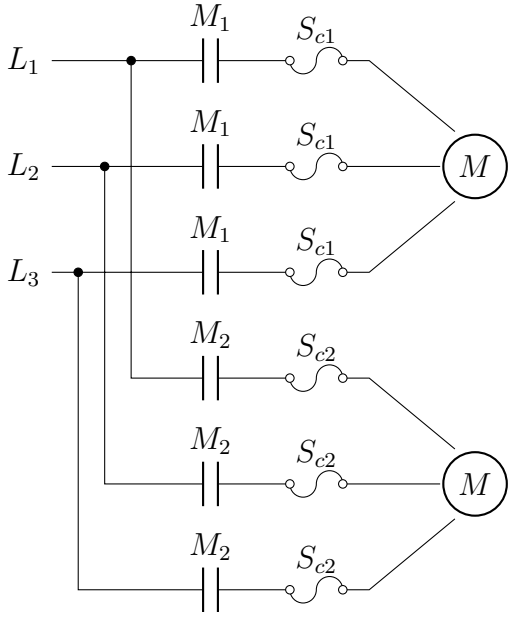


Figura 4.2: Circuito de potencia: arranque-pare secuencial de dos motores

Conteste las preguntas

¿Puede mostrar el circuito de control para ambas variantes y las ecuaciones lógicas? ¿Puede mostrar el diagrama de tiempos (oscilograma) del funcionamiento de cada bobina, y su relación con los eventos? ¿El circuito implementado funciona según lo planeado? ¿Cuanto es la corriente de arranque medida con el FLUKE 143B?

4.4.2. Actividad 2

Realice un arranque por auto-transformador para un motor trifásico de tres puntas. Mida la corriente de arranque y la tensión de alimentación del motor.

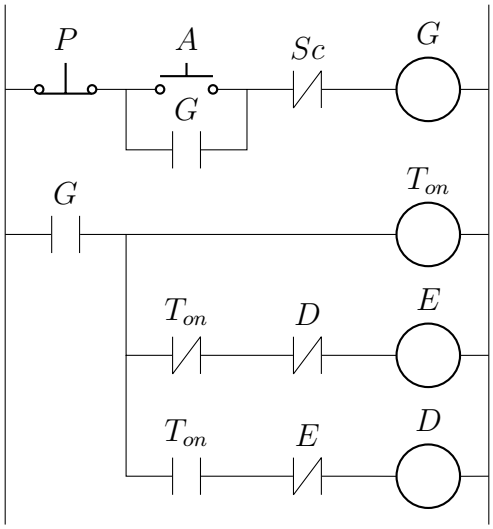


Figura 4.3: Circuito de control: arranque por auto-transformador

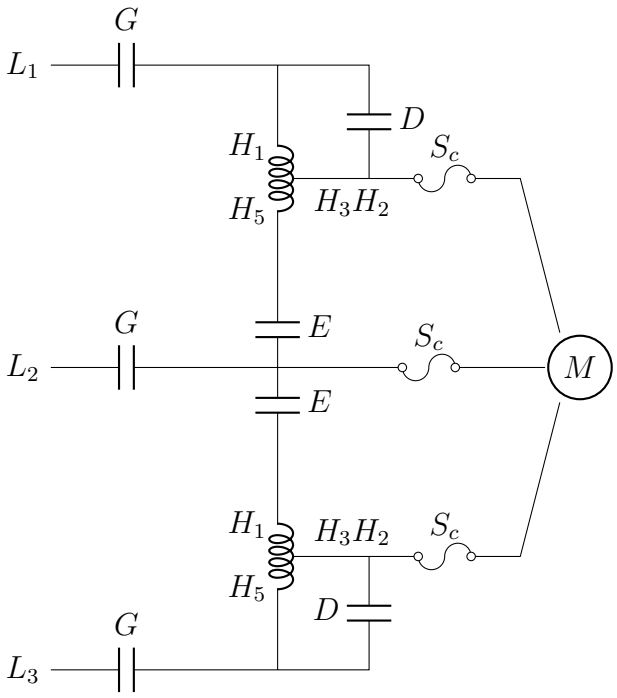


Figura 4.4: Circuito de potencia: arranque por auto-transformador

Conteste las preguntas:

¿Cual es la corriente de arranque del motor según su dato de placa y letra de código? ¿Cuanto es la corriente de arranque medida con el FLUKE 143B? ¿Cuanto se redujo la corriente en el arranque?¿Se cumple la ecuación $I_{linea} = a^2 \cdot I_{arr}$?

4.4.3. Actividad 3

Realice un arranque pare reversible para un motor que arranca en estrella-delta. Es decir, el motor debe arrancar en estrella-delta hacia la derecha, o puede arrancar en estrella-delta hacia la izquierda. Cuando se presiona el botón de pare o sobrecarga este se detiene. Solo se requiere un temporizador tipo On-Delay.

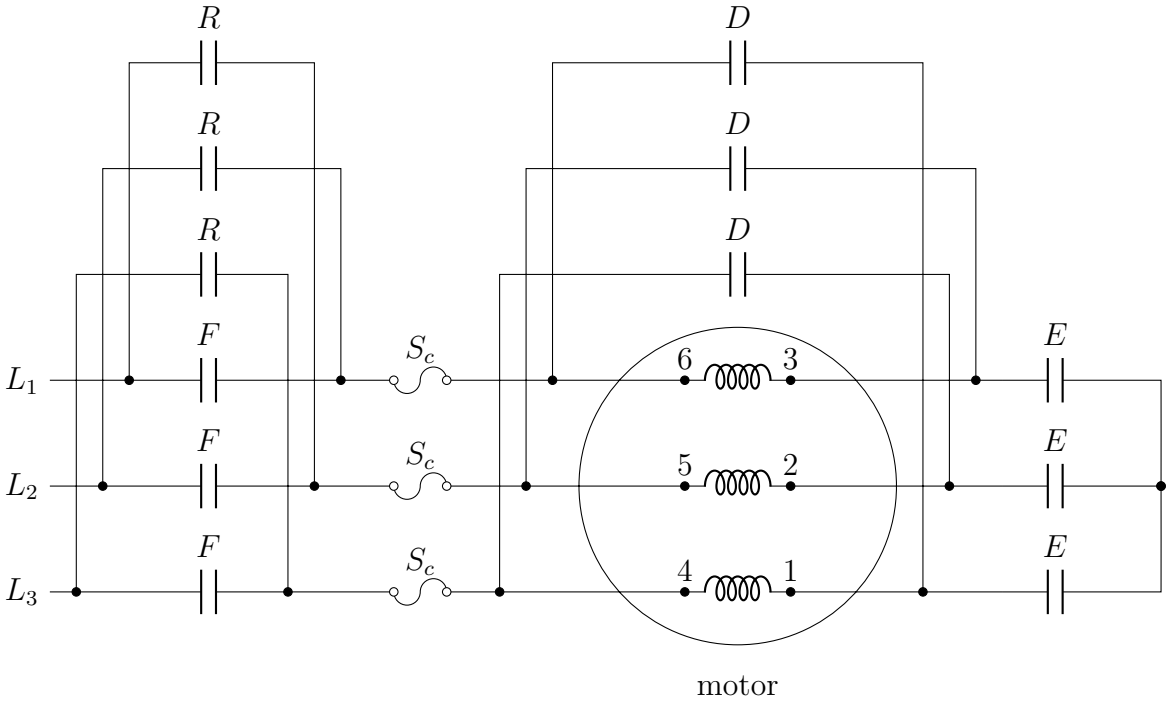


Figura 4.5: Circuito de potencia: arranque Estrella-Delta Reversible

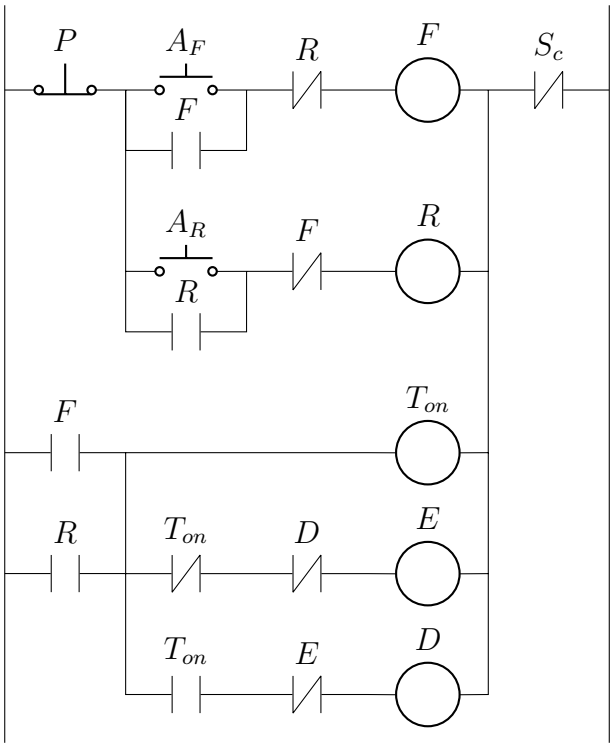


Figura 4.6: Circuito de control: arranque Estrella-Delta Reversible

Conteste las preguntas:

¿Como es el circuito implementado? ¿Cuanto es la corriente de arranque medida con el FLUKE 143B? ¿Como es el diagrama de tiempos de cada bobina? ¿La corriente de arranque se redujo un tercio del arranque a plena tensión?

Laboratorio 5

Laboratorio: Programación de variadores de frecuencia y arrancadores suaves.

5.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Alambrar el circuito de control y potencia del variador.
- Programar un variador de frecuencia segun los requerimientos.
- Alambrar el arrancador suave para un motor trifásico.
- Alambrar un arranque pare reversible usando un arrancador suave de un motor trifásico.
- Programar el variador de frecuencia.
- Medir el tiempo de las rampas de aceleración y desaceleración, y vincular con los parámetros programados.

5.2. Equipos y materiales

Para este laboratorio de necesitaran:

- 1 motor trifásico 1/2 hp de 3 puntas.
- 1 variador de frecuencia: ABB ACS355 o VFD-EL
- 1 arrancador suave SIEMENS RW3013.
- 2 contactores.
- 1 botón pulsar cerrado.
- 2 botones pulsadores abiertos.
- 3 interruptores N.O.
- 1 amperimetro de gancho con medición INRUSH.
- 1 analizador de señales trifásico FLUKE 143B.

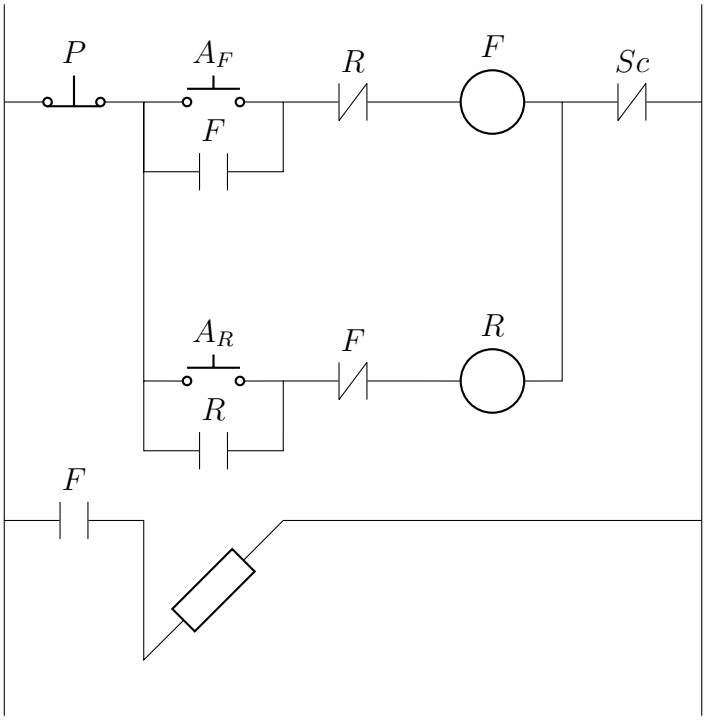
5.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

5.4. Práctica en Clase

5.4.1. Actividad 1

Suponga que el arrancador suave es para un motor de una cinta transportadora que puede operar en ambos sentidos. Según el manual del SIRIUS 3RW30 [2], sección 13.1.2, seleccione el voltaje de inicio y la rampa de tiempo recomendada. Implemente un circuito de control y potencia para que el motor arranque en dos sentidos, para esto puede usar como referencia la sección 16 de ejemplos.



Conteste las preguntas

¿Puede mostrar el circuito de control del sistema? Usando un analizador Fluke 143B mida la corriente de arranque del motor. ¿Como son las señales de corriente y voltaje a la entrada del motor con/sin arrancador suave? ¿Cuanto tiempo tardó en aparecer el voltaje de línea en las terminales?¿Coincide con la rampa de tiempo pre-establecida?

5.4.2. Actividad 2

Utilizando el manual del variador, encuentre el código que re-inicial el VDF a sus valores de fábrica. Luego ingrese los parámetros descritos en la sección 3.3 asociados al voltaje, corriente nominal y

frecuencia base del motor. Posteriormente seleccione un modo de entrada que permita operar el motor en ambos sentidos mediante cableado externo. Programe rampas de aceleración entre las frecuencias mínimas y máximas de 20 segundos y de desaceleración de 30. Programe siete frecuencias en el VDF, asumiendo torque constante en vacío,de tal forma que en el eje del motor se midan las velocidades de la Tabla 5.1.

Tabla 5.1: Velocidades solicitadas

Velocidad	Rev/min
0	900
1	1050
2	1200
3	1350
4	1500
5	1650
6	1800
7	1950

Conteste las preguntas:

Realice la gráfica de voltaje en función de la frecuencia ($V = \mathfrak{F}(f)$), ¿Es como se esperaba? Realice una tabla con las mediciones de las velocidades medidas, ¿Cuál fue el porcentaje de error relativo de cada medición? Si se programa el variador con frecuencias calculadas con torque variable, los errores mejoran o empeoran? ¿Por qué? ¿Son las gráficas PWM a la salida del variador son como se esperan? ¿Puede mostrar un gráfico? Si se desconecta una fase de entrada en VDF, ¿que sucedió con el VDF? Mida las rampas de aceleración y desaceleración, cumple lo programado? Cuanto tarda de pasar de la velocidad 3 a la 5 y viceversa, tienen estos tiempos sentido? Que pasaría si se programan rampas de 1 segundo de aceleración y des-aceleración?

Laboratorio 6

Laboratorio: Programación con LOGO!

6.1. Objetivos

Este laboratorio busca:

- Programar e implementar un arranque en Estrella -Delta usando un el relé inteligente LOGO!.
- Programar e implementar un alternador de dos bombas.
- Programar un alternador de bombas con señales de alarma y temporizadores semanales.
- Aprender las conexiones eléctricas de las entradas y salidas del controlador.
- Calcular el ciclo de trabajo de cada programa de cada programa

6.2. Equipos y materiales

Para este laboratorio de necesitaran:

- 2 motores trifásicos 1/2 hp de 3 puntas.
- 1 motor trifásico de 6 puntas.
- 1 Logo! 230RC de Siemmens.
- 1 Software LOGO! Confort.
- 4 contactores.
- 1 botón pulsar cerrado.
- 3 botones pulsadores abiertos.
- 1 Manual del LOGO! 8.
- 1 Manual LOGO!Soft Comfort.

6.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

6.4. Práctica en Clase

6.4.1. Actividad 1

El profesor realizara el programa de control para un arranque en estrella-delta en LOGO!Soft Comfort y explicará todo lo concerniente a: simulación del programa, y a la configuración de la red local (LAN), Mascara, pasarela de salida (gateway), y carga en el programa.

Conteste las preguntas

¿Que información brinda el comando IPCONFIG en COMMAND PROMPT de Windows?, ¿Que información brinda el comando PING?.

6.4.2. Actividad 2

Diseñe el programa para el controlador LOGO!, que resuelva el siguiente problema.
Se necesita un alternador de bombas para el llenado de un tanque de almacenamiento. El taque posee tres niveles, NA nivel alto, NB nivel bajo y Nivel Crítico NC, además existen 2 bombas B1 y B2. Cuando el agua activa NB la bomba B1 se activa y se apaga hasta alcanzar el nivel NA. La próxima vez que se active el sensor NB se activa la segunda bomba B2 y se apaga hasta alcanzar NA. Este ciclo se repite hasta que se presione el apagado general del sistema o se active una de las dos sobrecargas. Si el nivel crítico se presiona NC se encienden ambas bombas, además el sistema debe recordar el orden de la alternancia. El sistema cuenta con dos botoneras para el arranque y pare general del sistema. El sistema deberá mostrar en pantalla en todo momento, el estado de la bombas y la etapa lógica en que se encuentra el sistema.
Una vez simulada la solución, procederá a cargarla y alambrar el LOGO!. Para esto requerirá de las botoneras pulsadoras, contactores y finales de carrera que emulan los niveles de la boyas, etc.

Conteste las preguntas

¿Puede mostrar el diagrama lógico de la solución?, ¿Puede mostrar el diagrama de la implementación con sus ecuaciones lógicas?, ¿Puede mostar el diagrama de conexión eléctrico?.¿Cuando es el ciclo de escaneo de la solución? La ultima pregunta se contesta implementando el código que se aporta en apéndice B del manual [3].

6.4.3. Actividad 3

Diseñe el programa para el controlador LOGO!, que resuelva el siguiente problema.
La empresa **Automation Solutions** necesita controlar dos moto-bombas de forma alternada.

Las entradas del sistema son: botón del arranque motor A, botón de paro del motor A, presostato de confirmación del motor A, botón del arranque motor B, botón de paro del motor B, presostato de confirmación del motor B.
Las salidas del controlador son las señales a los contactares de la moto-bomba A y la moto-bomba B.
El sistema debe funcionar con la siguiente lógica:

- Si el sistema esta en modo manual, el arranque y pare se realiza desde las botoneras. Si a los 3 segundos no existe la señal de confirmación de presión, aparece en pantalla del LOGO! un mensaje indicando que la bomba no levanta presión, y se desactiva el enclave-miento de la señal del motor.
- Por otra parte, si la maneta esta en la posición de modo automático, la bomba X se enciende a las 5PM y se apaga a las 10PM. El arranque y pare de las moto-bombas es de forma alternada. Es decir, si el Lunes arranca la moto-bomba B1, el Martes deberá arrancar la moto-bomba B2 y así sucesivamente. Si la confirmación de presión no se recibe en un lapso de 3 segundo, el sistema cambiará a la otra bomba y mostrara en pantalla el aviso del cambio.
- Adicionalmente, existe un contador que indica en pantalla del LOGO!, la cantidad de ciclos de la bomba 1 y de la bomba 2. Cuando los ciclos realizados sean igual a 10 aparecerá en pantalla un mensaje indicando : “La bomba Bx requiere mantenimiento”. En pantalla se debe indicar la etapa lógica del sistema.

Conteste las preguntas

¿Puede mostrar el diagrama lógico de la solución?, ¿Puede mostrar el diagrama de la implementación con sus ecuaciones lógicas?, ¿Puede mostar el diagrama de conexión eléctrico?.¿Cuando es el ciclo de escaneo de la solución? La ultima pregunta se contesta implementando el código que se aporta en apéndice B del manual [3].

Laboratorio 7

Laboratorio: Programación de Autómatas Lógicos

7.1. Objetivos

Este laboratorio busca:

- Entender la implementación de la norma IEC 61131-3.
- Programar de manera estructura el PLC S7-1500 usando bloques OB, FB y DB.
- Utilizar tres lenguajes estandarizados para la programación de bloques lógicos.
- Solucionar problemas comunes de automatización.

7.2. Equipos y materiales

Para este laboratorio de necesitaran:

- 1 PLC SIEMENS 1500.
- 1 Software TIA Portal V17.
- 1 Computadora.
- 1 Manual del software TIA Portal.
- 1 Colección de Manuales S7-1500

7.3. Metodología

Este laboratorio tiene una duración de 4 lecciones, repartidas en dos semanas. Los estudiantes deben mostrar durante las clases programadas las tres actividades propuestas. Deben recabar fotografías y resultados de los equipos de medición para elaborar las evidencias. Las evidencias se subirán al TecDigital la semana siguiente finalizadas las actividades.

7.4. Práctica en Clase

7.4.1. Actividad 1

Se requiere programar un arranque estrella-delta con multiples instancias. Para realizar esta actividad, utilize el lenguaje de contactos (KOP) para programar un estrella - delta dentro de un Bloque de función (FB). El FB tendrá como parámetros de entrada: la señal de arranque, pare, sobre carga y tiempo del temporizador. Como señales de salida tendrá la de la bobina general, la bobina que realiza el estrella y la bobina en delta.
Luego en el bloque de organizacion uno (OB1), se llama el bloque FB multiple veces. Por ejemplo, llame el FB tres veces y ponga tiempos de arranque distintos {3, 6, 9} segundos al parámetro tiempo de cada instancia.

Conteste las preguntas

Si se presiona el arranque al mismo tiempo, ¿como fue diagrama de tiempos de las 9 señales? ¿Puede mostrar el diagrama KOP del FB y OB1 implementado?

7.4.2. Actividad 2

Se necesita automatizar un parque de pozos profundos, el parque posee tres pozos y y tres tanques de captación. Se desea un sistema general que inicialice y pares los tres subsistemas, es decir, cuando se presione el arranque inicien el sistema de los tres pozos y cuando se presione el pare se detengan. Cuando se active una sobrecarga de algún motor ese subsistema en específico deja de funcionar, pero los otros dos subsistemas de pozos y tanque siguen funcionando normalmente. Los subsistemas se llamarán S1, S2 y S3 respectivamente.
Cada subsistema de pozo profundo se compone de dos bombas que funcionan en alternancia, un pozo de captación y un tanque de almacenamiento. Las señales del tanque de almacenamiento indican el nivel alto **T1-Na**, y nivel bajo **T1-Nb** de agua, el pozo profundo posee también dos sensores de nivel, nivel alto **P1-Na** y nivel bajo **P1-Nb**, además existen las dos señales de salida para las dos bombas: **S1-B1** y **S1-B2**.
EL funcionamiento del subsistema es de la siguiente forma: las bombas toman el agua del pozo y la deposita en el tanque de captación. El nivel de agua **T1-Nb** activa la bomba **S1-B1** siempre y cuando la señal de **P1-Na** este activa. La bomba **S1-B1** se apaga hasta alcanzar el nivel **T1-Na** o que se quede sin agua indicado por el sensor del pozo **P1-Nb**. Dicho de otra forma, si antes de alcanzar el sensor **T1-Na** se activase el nivel **PI-Nb**, se apaga la bomba. La próxima vez que se active el sensor **T1-Nb** y este activa el sensor **P1-Na** se activa la segunda bomba **S1-B2** y se apaga hasta alcanzar **T1-Na** o **P1-Nb**. Este ciclo se repite hasta que se presione el apagado general del sistema o se active una de las dos sobrecargas del subsistema.
Para los subsistemas 2 y 3 simplemente cambie los subindices en los sensores del tanque, pozo, bombas y sobrecargas son: **T#-Nx**, **P#-Nx**, **S#-B#**, **S#-Sc#**. Por ejemplo, la codificación de los sensores de nivel alto del tanque y el pozo se indica de la forma: **T3-Na** y **P3-Na**.
Cada subsistema posee un contador de número de ciclos. El valor del contador es un parámetro de entrada. Cuando se llega al valor se activa una señal de alarmar para el mantenimiento. La alarma se restablece un boton pulsador.

- Realice la programación con el Gráfico de Función Secuencial de SIEMENS llamado GRAPH.

- Cree un Bloque de función FB1 genérico usando el lenguaje GRAPH.
- Cree un Bloque de función FB2 general en diagrama de contactos (KOP), donde se llame 3 veces al bloque anterior y renombre todos los parametros de entrada.
- En el Bloque de organización OB1, inserte el bloque FB2.
- Verifique el funcionamiento la solución del problema mediante simulación.
- Cargue el programa en el PLC S7-1500.
- Active el monitoreo de las entradas y salidas del PLC y su visualización en la pantalla.

Conteste las preguntas

¿Como es la implementación de contadores y temporizadores en GRAPH? ¿Como se declaran los parámetros de entrada y salida de un bloque de función (FB)? ¿Como se comportó el sistema implementado? ¿Puede mostrar el diagrama del FB1 y FB2?

7.4.3. Actividad 3

Repita la actividad 2, pero implemente el ejercicio en texto estructurado (ST), con la variante que cada bomba del pozo requiere arrancar en estrella-delta. Para esta actividad podrá usar el FB programado en la actividad 1, y llamar este bloque múltiples veces.

Apéndice A

Marcos de referencia

A.1. Arduino

Un micro-controlador es un circuito integrado pequeño que contiene un microprocesador, memoria y periféricos como entradas/salidas (IO), comunicación, almacenamiento y sensores. Estos componentes están integrados en un solo chip y pueden ser programados para controlar diferentes dispositivos y sistemas.
Entre las partes que pueden contener los micro-controladores se encuentran:

- Microprocesador: es la unidad aritmética lógica, registros asociados y controladores que realizan las operaciones del sistema y es responsable de ejecutar las instrucciones del programa.
- Memoria: donde se almacena el programa y los datos.
- Entradas/Salidas (IO): permite la comunicación entre el micro-controlador y el mundo exterior.
- Comunicación: permite que el micro-controlador se comunique con otros dispositivos a través de diferentes protocolos como USB, Ethernet, etc.
- Almacenamiento: permite almacenar datos en el dispositivo, como una EEPROM o una memoria flash.
- Sensores: permiten medir diferentes variables ambientales como la temperatura, la humedad, la presión, etc.

Los micro-controladores tienen muchos usos, incluyendo:

- Control de motores,
- Automatización de procesos industriales,
- Dispositivos de medida y monitoreo,
- Control de sistemas de iluminación y calefacción,
- Control de sistemas de seguridad,
- Control de electrodomésticos y dispositivos de consumo,

- Control de sistemas de comunicación,
- Control de sistemas de vehículos,
- Control de robots y sistemas automatizados.

A.1.1. Arduino

El Arduino es una plataforma de desarrollo de hardware y software libre basada en un micro-controlador. La programación de Arduino se basa en el lenguaje de programación C++ y utiliza un entorno de desarrollo integrado (IDE) específico llamado Arduino IDE. El Arduino IDE es una aplicación de escritorio que se utiliza para escribir, depurar y cargar código en el micro-controlador. El procedimiento básico de programación del Arduino es el siguiente:

a. Conectar el Arduino a la computadora mediante un cable USB.

b. Abrir el Arduino IDE y seleccionar el tipo de Arduino y la tarjeta que se va a utilizar en el menú Herramientas.

c. Escribir el código en el editor de código del Arduino IDE, utilizando el lenguaje de programación C++.

d. Verificar el código compilando el código utilizando el botón verde de compilación del Arduino IDE.

e. Cargar el código en el Arduino utilizando el botón azul de carga del Arduino IDE.

f. Observar el comportamiento del código en los pines de entrada y salida del Arduino, si es necesario realizar cambios en el código para obtener el resultado deseado.

g. Repetir los pasos 3-6 hasta que el código funcione correctamente.

El esquema básico de programación de Arduino utiliza dos funciones esenciales [4]: setup() y loop(). La función setup(): Es la primera función que se ejecuta cuando el Arduino se enciende o se reinicia. En esta función se configuran los pines de entrada y salida, se establecen las velocidades de comunicación, se inicializan las variables, entre otras tareas de configuración.
La función loop(): Es la función que se ejecuta continuamente después de que se ha ejecutado la función setup(). En esta función se escriben las instrucciones que se deben ejecutar continuamente, como la lectura de sensores, el control de actuadores, la comunicación con otros dispositivos, entre otras tareas.
Las instrucciones y funciones del lenguaje C++ para Arduino pueden ser consultadas en el siguiente vínculo.

A.2. Combinacionales

Un circuito combinacional digital es un circuito que produce una o varias salidas en función de sus entradas actuales. Es decir, no mantiene ningún estado interno y su salida depende exclusivamente de sus entradas en el momento actual. Existen muchas formas de implementar circuitos combinacionales,

a partir de contactos eléctricos, válvulas neumáticas, transistores, software, etc. Sin embargo cualquier circuito lógico combinacional se puede expresar mediante cuatro formas equivalentes que describen su funcionamiento: ecuación lógica, tabla de verdad, diagrama de tiempo, y circuito eléctrico. Puede encontrar información adicional en el siguiente enlace https://es.wikipedia.org/wiki/Sistema_combinacio

A.2.1. Implementan de expresiones Booleanas

Un circuito combinacional puede ser representado de distintas formas, pero la más común es mediante una ecuación lógica, por ejemplo:

$$F(a,b,c) = a \cdot b + \bar{a} \cdot \bar{b} \cdot c \tag{A.1}$$

La expresión anterior posee una representación en suma de productos (SDP), puede ser implementada en C de la siguiente forma:

```
bool F(bool a,bool b,bool c){
  return  (a && b) || (!a && !b && c);
}
```

Existen ocho formas estándar de implementar la ecuación (A.1) digital-mente y por consecuencia en software . Otras maneras de implementar las funciones lógicas son con el producto de sumas (PDS), las implementaciones a dos niveles NAND/NAND y NOR/NOR, todas las implementaciones deben arrojar los mismos resultados.
Por otra parte, la ecuación (A.1) es equivalente a la expresión booleana NAND/NAND si se niega dos veces y se aplica el Teorema de Morgan.

$$F(a,b,c) = a \cdot b + \bar{a} \cdot \bar{b} \cdot c \tag{A.2}$$

$$F(a,b,c) = \overline{\overline{a \cdot b + \bar{a} \cdot \bar{b} \cdot c}} \tag{A.3}$$

$$F(a,b,c) = \overline{\overline{a \cdot b} \cdot \overline{\bar{a} \cdot \bar{b} \cdot c}} \tag{A.4}$$

Notece que la ecuacion (A.4) necesita conectivas lógicas (funciones) de 2 y 3 entradas. Esto no es un problema cuando se implementa en SOFTWARE, pero sí se requiere hacer una implementación fisica hay que aplicar el Algebra Booleana para dejar la expresión en términos de conectivas de solamente dos entradas. Lo que se hace es que el término de tres literales, le aplicamos una doble negación tal como sigue.

$$F(a,b,c) = \overline{\overline{a \cdot b} \cdot \overline{\bar{a} \cdot \bar{b} \cdot c}} \tag{A.5}$$

La expresión (A.5) se encuentra en términos de conectivas NAND de dos entradas y se puede implementar en ARDUINO de la siguiente forma:

```
bool F(bool a, bool b, bool c){
  bool result = false;
  result = NAND(NAND(a,b),NAND(NAND(NAND(NAND(a,true),NAND(b,true)),true),c));
  return result;
}
```

donde la NAND de dos entradas es:

```
bool NAND(bool x, bool y){
  return !(x && y);
}
```

A.2.2. Multiplexores y Decodificadores

Un multiplexor (Mux) es un circuito combinacional con 2^n entrada, más n entradas selectoras o de control y una salida. Este circuito combinacional selecciona con las n entradas control, el valor booleano de una entrada entre $[0, 2^n - 1]$ y coloca dicho valor en la salida. La ecuación general de cualquier multiplexor es la siguiente,

$$MUX = \sum_{i=0}^{2^n-1} I_i \cdot \mathbb{S}_i \tag{A.6}$$

donde $\mathbb{S}_i$ es la i-ésima permutacion de las entradas de selección. A modo de ejemplo un Mux 2×1 posee dos entradas digitales, una de control y una salida. Un Mux 4×1 posee cuatro entradas digitales, dos de control y una salida, Un Mux 8×1 posee ocho entradas digitales, tres de control y una salida.
La ecuación de un Mux 4×1 es la siguiente,

$$F(I_0, I_1, I_2, I_3, S_1, S_0) = I_0\bar{S}_1\bar{S}_0 + I_1\bar{S}_1S_0 + I_2S_1\bar{S}_0 + I_3S_1S_0 \tag{A.7}$$

El código para implementar dicho multiplexor es similar al siguiente,

```
bool F(bool I0,bool I1,bool I2,bool I3,bool S1,bool S0){
  return (I0&&!S1&&!S0 || I1&&!S1&&S0 || I2&&S1&&!S0 || I3&&S1&&S0) ;
}
```

Los circuitos que realizan función inversa del multiplexor se llama Demultiplexores y usualmente tiene una entrada digital más n entradas de selección y 2^n salidas digitales.
Otro ejemplo de circuito combinacional son los Decodificadores y Codificadores. Un codificador es un circuito combinacional con 2^n entradas y n salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activada. Mientras que un decodificador hace el trabajo inverso, recibe un código binario y lo transforma en cualquier otro código, ya sea binarios como el Gray o exceso 3; o otras reprentaciones numericas: octal, decimal, hexagecimal. La figura A.1 muestra la implementan digital de un decodificador de 2 a 4 líneas, la tabla de verdad del circuito y las ecuaciones características de cada salida.
Una posible implementación para Arduino de este decodificador es la siguiente.

```
byte  DECO (bool A0, bool A1)
{
  int i=0;
  bool D[4]={false, false, false, false};
  byte deco=0;

  D[0]=!A1 && !A0;
  D[1]=!A1 && A0;
  D[2]=A1 && !A0;
  D[3]=A1 && A0;
```

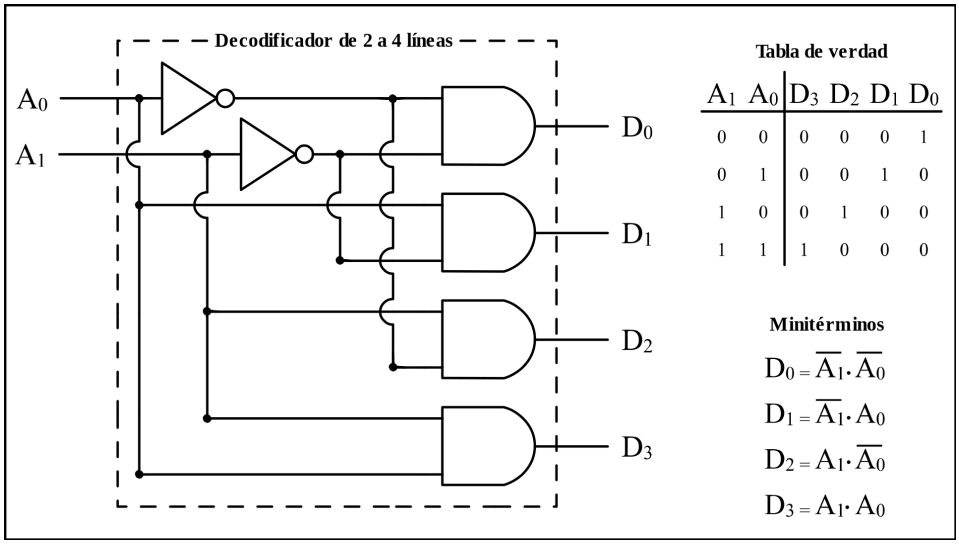


Figura A.1: Decodificador de código binario de dos bits a cuatro líneas de salida.

```
  for (i=0; i<sizeof(D);i++)
  {
    bitWrite(deco, i, D[i]);
  }
return deco;
}
```

A.3. Lógica cableada

El control eléctrico de motores es un conjunto de técnicas y dispositivos utilizados para administrar la operación de un motor eléctrico. El control eléctrico de motores se utiliza, para regular la velocidad, el par, la dirección de rotación, el arranque y el frenado del motor.
En general, el control eléctrico de motores implica la utilización de circuitos electrónicos y sistemas de control para ajustar el suministro de energía eléctrica al motor y así lograr la operación deseada. El control eléctrico de motores se aplica en una amplia gama de aplicaciones, desde equipos de producción industrial hasta electrodomésticos y sistemas de transporte.

A.3.1. Contactor y relevador térmico

Un contactor es un dispositivo electromecánico utilizado para controlar la conexión y desconexión de motores eléctricos u otros equipos eléctricos de alta potencia como; calentadores eléctricos, luminarias, banco de capacitores, etc. Los contactores consisten en un electroimán que activa los contactos eléctricos para permitir o interrumpir el flujo de corriente eléctrica hacia el motor, un ejemplo de un contactor de uso general se muestra en la Figura A.2a. Los contactores suelen utilizarse en combinación con los relés de sobrecarga.
Un relé de sobrecarga es un dispositivo de protección utilizado para evitar el sobrecalentamiento del motor eléctrico; la Figura A.2b muestra un relé de sobrecarga electrónico. Estos se clasifican en dos grandes tipos: Los térmicos y los electrónicos. Los relés térmicos detectan el aumento de temperatura del motor y, si la temperatura excede un cierto umbral, desconectan el motor para evitar daños. Los

(a) Contactor de proposito general Eaton [5].

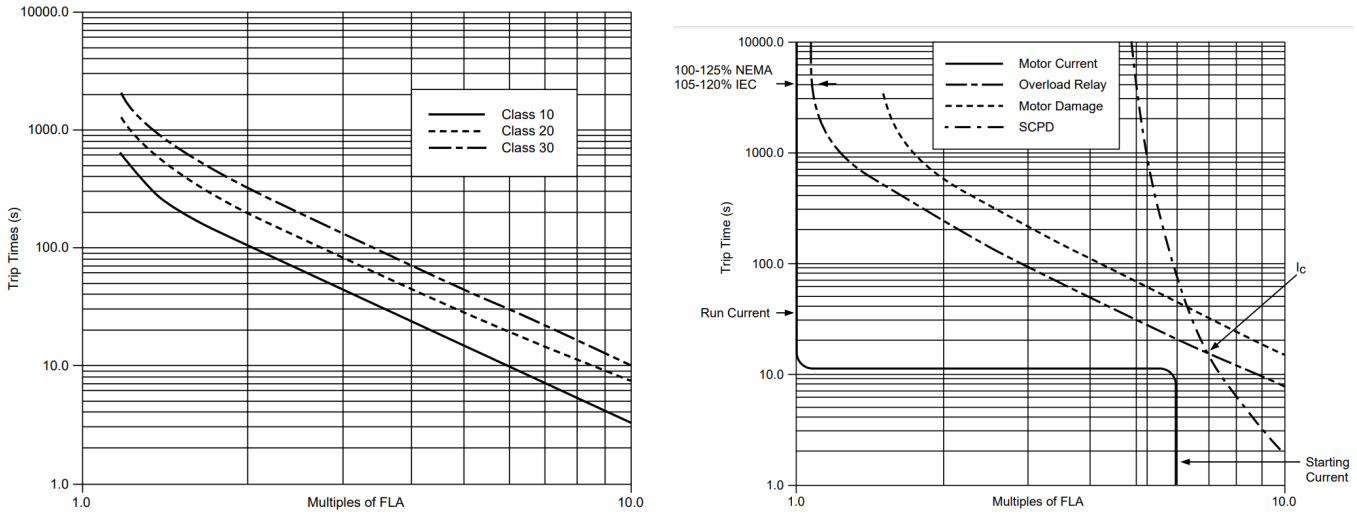
(b) Relé de protección térmica Eaton [6].

Figura A.2: Elementos de control y protección de motores.

relés térmicos funcionan mediante el uso de elementos de medición de temperatura, como fusibles de alección eutéctica o bimetálicos, que desconectan los contactos del relé en caso de sobrecalentamiento. Por otra parte los relés electrónicos posee una electrónica auto-alimentada con termistores que posee las siguientes características: poseen disparo ajustables para una protección óptima en diferentes condiciones de arranque, detectan perdida de fase y desbalance, poseen liberación automática/manual para un rápido reinicio del proceso para una mayor disponibilidad, poseen una sensibilidad monofásica ajustable para cargas simétricas y asimétricas, detección integrada de la corriente de defecto a tierra en algunos casos y no requieren compensación térmica como los elementos térmicos.
En conjunto, el contactor y el relé térmico se le llama arrancador electromagnético y se utilizan comúnmente en el control eléctrico de motores para proporcionar una interrupción segura y eficiente. Cuando se utiliza un contactor y un relé térmico en combinación, el contactor controla la conexión y desconexión del motor, mientras que el relé térmico proporciona protección contra el sobrecalentamiento del motor.

A.3.2. Curvas de disparo

Las curvas clase 10, 20 y 30 son curvas de tiempo de disparo utilizadas en los elementos térmicos del relés según la norma NEMA (National Electrical Manufacturers Association) para la protección de motores eléctricos, ver Figura A.3a.
Cada curva indica el tiempo tiempo de apertura del circuito en función de la corriente de sobrecarga. La norma NEMA especifica los valores de corriente para cada curva son:
Curva clase 10: Esta curva se dispara en un tiempo de 10 segundos a una corriente de sobrecarga igual a 600 % de la corriente nominal del motor.
Curva clase 20: Esta curva se dispara en un tiempo de 20 segundos a una corriente de sobrecarga igual a 600 % de la corriente nominal del motor.
Curva clase 30: Esta curva se dispara en un tiempo de 30 segundos a una corriente de sobrecarga igual a 600 % de la corriente nominal del motor.



(a) Curvas NEMA Clase 10 , 20 , 30.

(b) Coordinacion de curvas y zona de daño del motor.

Figura A.3: Curvas de protección térmica [7].

Es importante tener en cuenta que estas curvas de tiempo de disparo son solo una guía para la selección de los elementos térmicos, y la selección final dependerá de las características específicas del motor y la aplicación. Adicionalmente es importante que exista un correcta coordinación con el dispositivo de protección contra corto-circuito (SCPD) o breaker. La Figura A.3b muestra el comportamiento de la corriente de un motor, la curva de protección térmica, la curva del SCPF y la curva que define la zona en que el motor se daña.

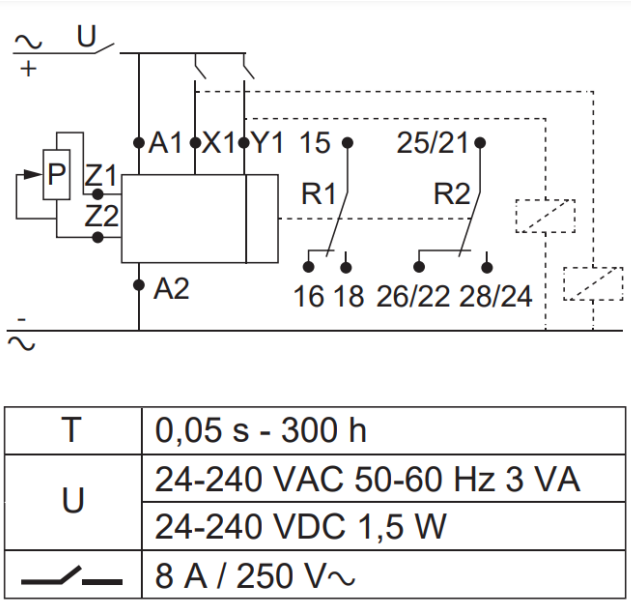
A.3.3. Selección de componentes

Usualmente cada fabricante propone su procedimiento de selección para contactores o relevadores, sin embargo los procedimientos tienen criterios comunes. Usualmente los contactores se seleccionan por: potencial del motor, voltaje de línea, voltage de la bobina, frecuencia de la red, vida útil (o cantidad de operaciones mínima requerida) y tipo de protección ambiental que se requiera, esto último se refiere al tipo de gabinete donde se monta el contactor. Por otra parte, los relevadores se seleccionan según la corriente nominal, la aplicación determina el tiempo de disparo de la curva, el tamaño del contactor al que debe ser acoplado y la compensación ambiental sólo en relevadores térmicos.

A.4. Arranques a tensión reducida

El arranque de motores trifásicos a tensión reducida es una técnica utilizada para reducir la corriente de arranque y el par de arranque, lo que ayuda a evitar problemas de sobrecarga en el sistema eléctrico.
Existen diferentes técnicas, pero las más comunes son: de arranque a tensión reducida, como el arranque estrella-triángulo, el arranque con autotransformador, arranque con resistencias estatóricas o rotóricas, arrancadores suaves, etc. Cada técnica tiene sus propias ventajas y desventajas y debe seleccionarse en función de las características del motor, la aplicación y las limitaciones del sistema eléctrico.

(a) Temporizador RE22R2MYMR [9]



T	0,05 s - 300 h
U	24-240 VAC 50-60 Hz 3 VA
	24-240 VDC 1,5 W
	8 A / 250 V~

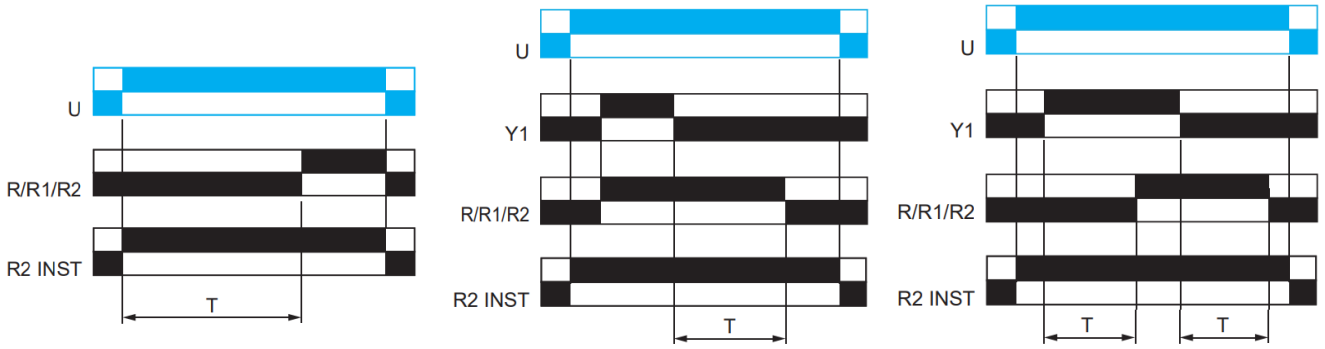
(b) Diagrama de conexión del RE22R2MYMR [8]

Figura A.4: Temporizador de 8 funciones RE22R2MYMR.

A.4.1. Temporizadores Electrónicos

Los temporizadores son dispositivos de control de tiempo que permiten activar o desactivar algún actuador un tiempo después de activarse o desactivarse una señal de control. Los temperorizadores pueden ser construidos de forma mecánica, neumática, electromecánica, electrónica, programada pero todos tienen funcionamientos estandarizados como: retardo a la conexión (On-delay), retardo a la desconexión (Off-delay), retardo a la conexión y desconexión (On/Off-Delay), relés de pulsos simétricos, entre otros.

La Figura A.4a muestra un temporizador electrónico y la Figura A.4b su diagrama de conexión. Note que el dispositivo se alimenta mediante las borneras A1-A2, las señales de control se declaran como X1 y Y1, y los contactos auxiliares normalmente cerrados y abiertos son las borneras 15-16-18 y 21-22-24. Este temporizador posee cuatro potenciometros o diales: el primer dial selecciona el rango de tiempo R, con el segundo dial se selecciona un valor V entre 1 y 10, el tiempo de activacion T se calcula como $T = R \cdot V/10$. El tercer dial selecciona una de las ocho funciones preestablecidas: a modo de ejemplo la Figura A.5 muestra las funciones A, C, Ac que representan temporizador operando como: On-delay, Off-delay y On-Off delay. Finalmente el cuarto dial, permite establecer un tiempo de retardo entre los contactos auxiliares R1 y R2, más información se detalla en [8].



(a) A: On delay (b) C: Off delay (c) Ac: On-Off delay

Figura A.5: Diagramas de tiempo de funciones a escoger en el selector del temporizador [8].

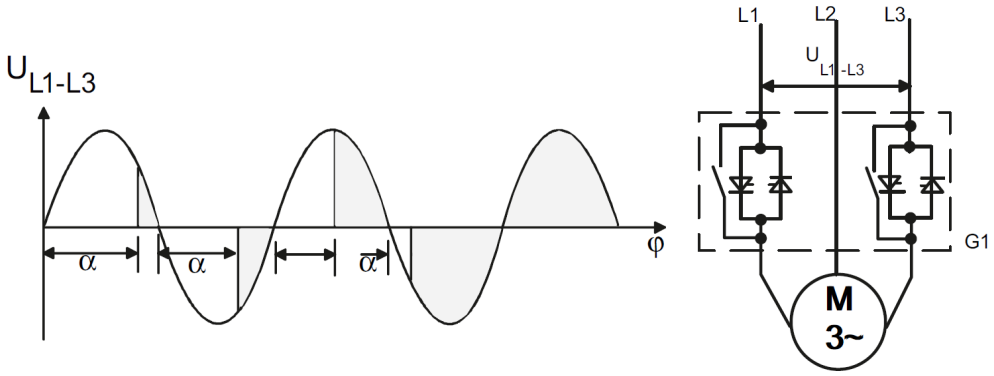
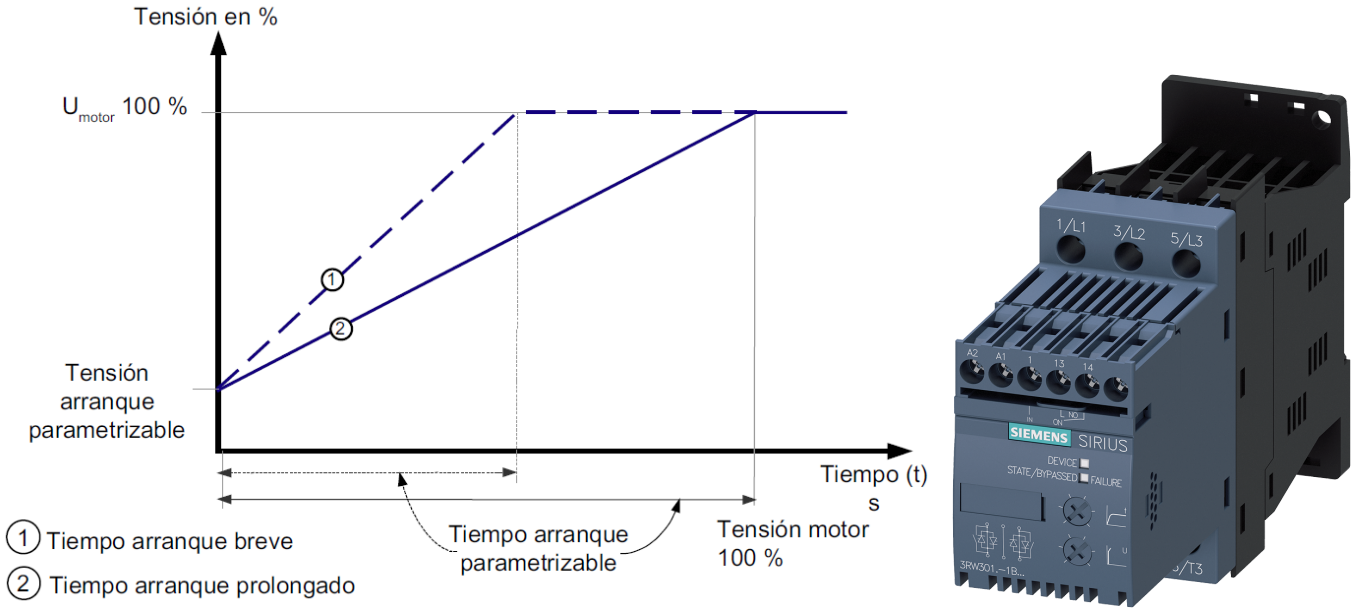


Figura A.6: Control por recorte de fase y esquema de un arrancador suave con control bifásico. [2]

A.5. Variadores de frecuencia

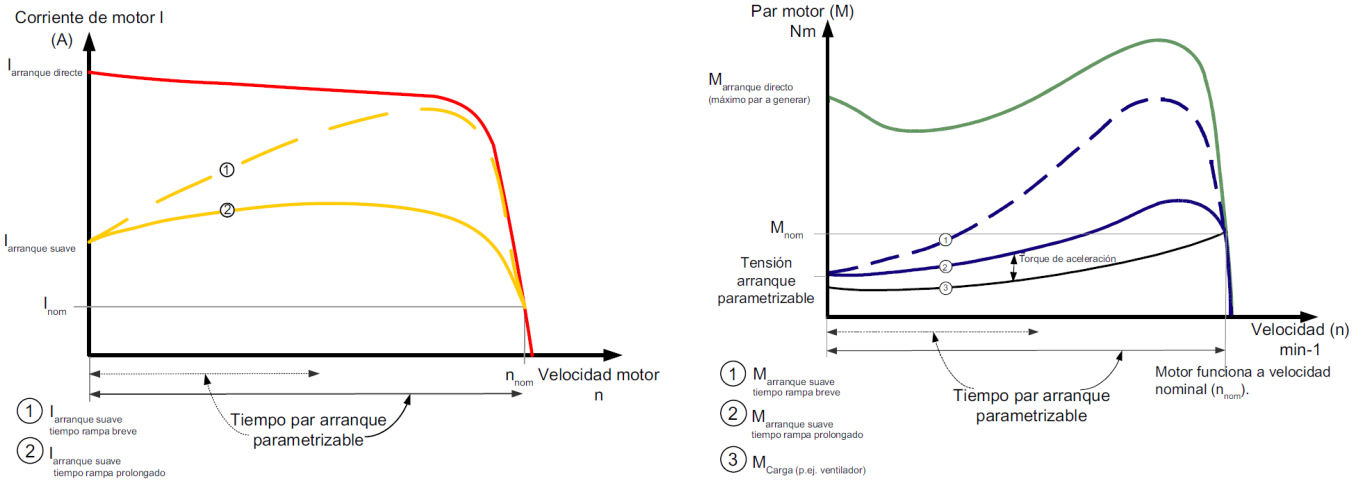
Los arrancadores suaves son sistemas electrónicos de control de motores que permiten arrancar y parar motores asíncronos de inducción. Los arrancadores suaves disponen de dos tiristores conectados en antiparalelo en dos de las tres fases como se muestra en la Figura A.6. La corriente en la tercera fase no controlada, es una suma de las corrientes de las dos fases controladas. El arrancador suave realiza un recorte del angulo α en cada fase, por lo tanto, la tensión del motor aumenta desde un voltaje inicial hasta la tensión a plena carga en un tiempo definido por el usuario. La tensión de inicio y el tiempo de la rampa se relacionan según la Figura A.7a y estos parámetros son ajustados mediante los dos potenciometros de la Figura A.7b .

La intensidad del motor tiene un comportamiento proporcional a la tensión aplicada al motor. De este modo, la corriente de arranque se reduce en el mismo factor que la tensión aplicada al motor. La Figura A.8a muestra la reducción en la corrientes. Por otra parte, el torque tiene un comportamiento cuadrático respecto a la tensión aplicada al motor. La Figura A.8b muestra como el par de arranque se reduce de forma cuadrática con la tensión aplicada al motor. En ambos casos se aprecia las curvas de corrientes y torque con rampas de tiempo cortas o prolongadas.



(a) Principio de funcionamiento (b) Imagen del arrancador

Figura A.7: Arrancador suave SIEMENS 3RW3013.[2]



(a) Corrientes del motor (b) Torque del motor

Figura A.8: Curvas de corrientes y torques versus velocidad.[2]

A.5.1. Variadores de Frecuencia

Un variador de frecuencia (VDF) es un dispositivo que se utiliza para controlar la velocidad y el par de un motor eléctrico alterando la frecuencia y el voltaje suministrado al mismo. El VDF es capaz de convertir la corriente alterna (CA) de entrada, en una señal de pulsos alternos con frecuencia y voltaje adecuados para el motor. Esta señal de salida se llamada modulación por ancho de pulso (PWM). La Figura A.9 muestra señales PWM que equivalentes a señales sinosoidales de: a) 60 Hz y 120V, b)30Hz y 60V, c) 20Hz y 40V.
El variador de frecuencia permite variar la velocidad del motor eléctrico en función de las necesidades de la aplicación, lo que puede resultar en un ahorro de energía y un mayor control de la velocidad. Además, el variador de frecuencia también puede proporcionar protección al motor, ya que puede detectar y proteger contra situaciones de sobrecarga, cortocircuito y fallos de fase, entre otros problemas.
Un VDF puede operar bajo dos principios: el control escalar y control vectorial [11]. El control escalar, también conocido como control V/f (voltaje/frecuencia), es el método de control más simple y comúnmente utilizado en los variadores de frecuencia. En este método, se ajusta la frecuencia y el voltaje de salida del variador en función de la carga del motor para mantener una relación de flujo magnético constante equivalente a la relación V/f. El control escalar es más adecuado para aplicaciones que requieren un control de velocidad básico, como ventiladores y bombas.
Por otra parte, el control vectorial, conocido como control de campo orientado, es una técnica de control avanzada que permite un mayor grado de precisión en el control del motor. En este método, se utiliza un modelo matemático del motor para controlar tanto la velocidad como el par del motor de manera independiente. Esto se logra mediante el control de la corriente y el voltaje en el estator del motor, lo que permite un control preciso de la orientación del campo magnético del motor. El control vectorial es más adecuado para aplicaciones que requieren un control de velocidad preciso y una alta dinámica, como en máquinas herramienta y robots industriales.

A.5.2. Partes del variador

La Figura A.10 muestra las principales partes del VDF y se detallan a continuación:

a. Rectificador: Es la parte que convierte la energía eléctrica de entrada en una señal de corriente directa (CD) que se utiliza como entrada para el inversor.

b. Inversor: Es la parte que convierte la señal de corriente directa (CD) en una señal PWM con la frecuencia y el voltaje adecuados para el motor.

c. Microprocesador: Es el cerebro del variador y controla todas las funciones y operaciones del mismo. El microprocesador recibe las señales de entrada y, en función de los parámetros de programación establecidos, envía las señales adecuadas al rectificador e inversor para controlar la velocidad y el par del motor. Adicionalmente el controlador se encarga de monitorear el variador, incluyendo la protección del motor contra sobrecargas, cortocircuitos, fallos de fase, etc.

d. Interfaz de usuario: Es la parte que permite a los usuarios interactuar con el variador, configurar los parámetros de programación y monitorear su operación.

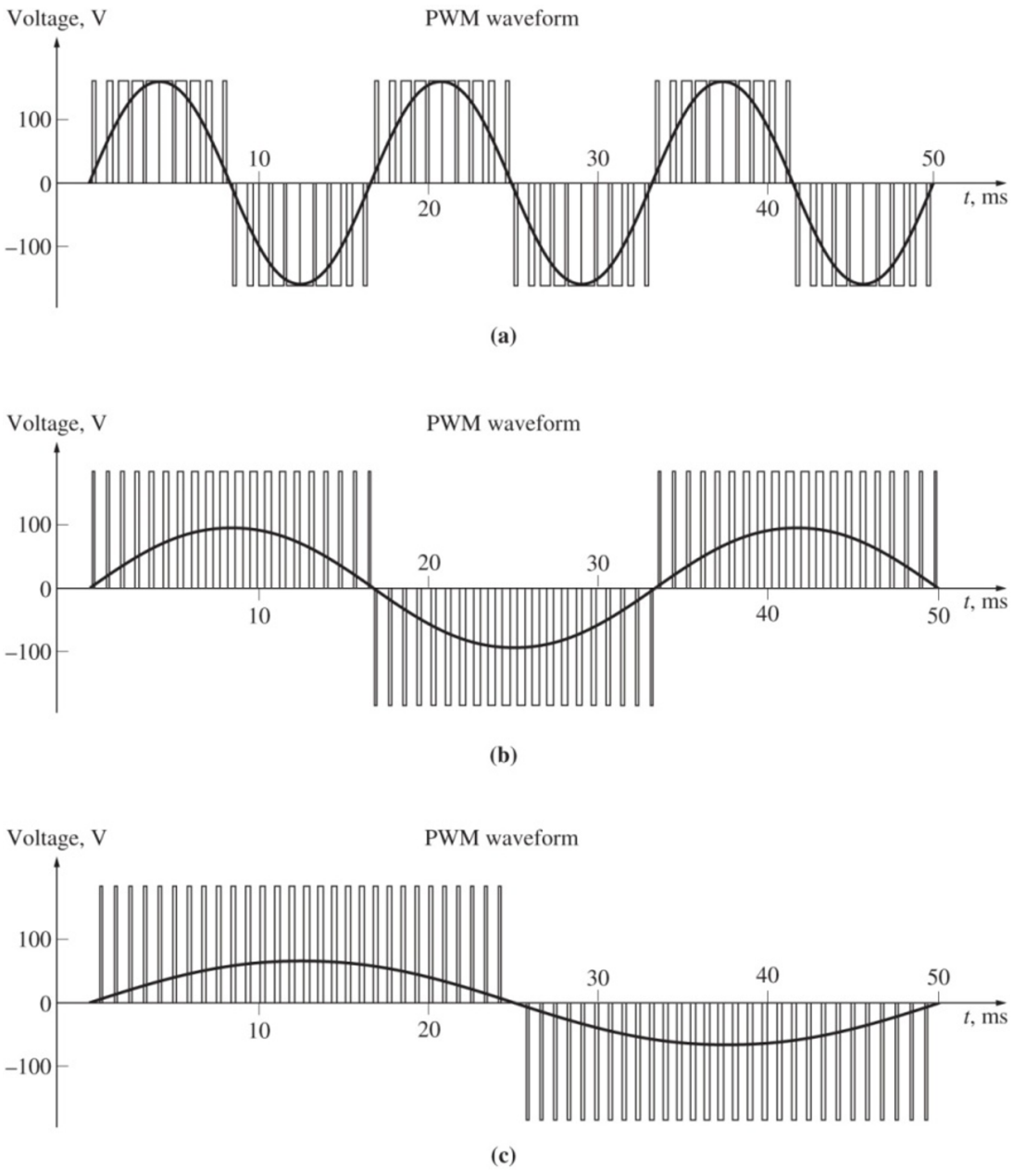


Figura A.9: Señales PWM equivalentes a señales sinosoidales de distintas amplitud y frecuencia [10].

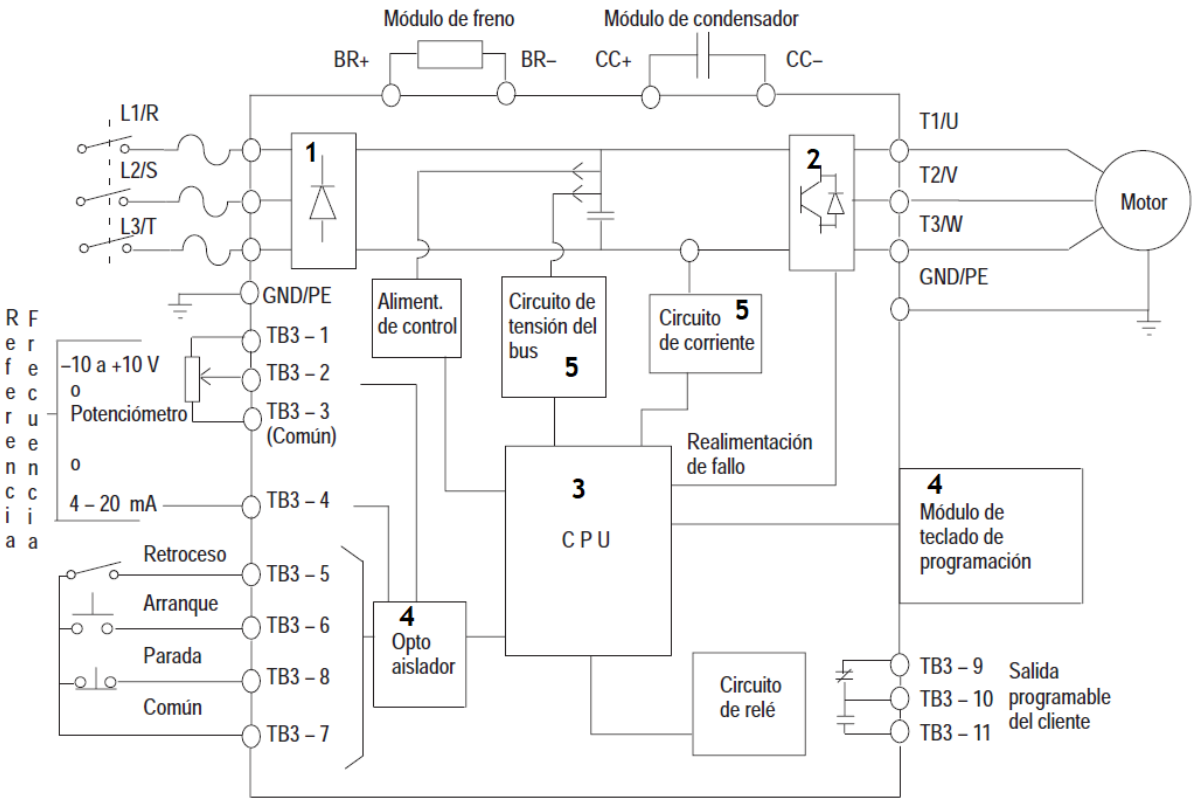


Figura A.10: Esquema del Variador de Frecuencia AB160.[12]

e. Circuitos de medición: son los circuitos encargados de medir las variables eléctricas del variador y el motor, incluyendo la protección del motor contra sobrecargas, cortocircuitos, fallos de fase, etc.

A.5.3. Parámetros de programación

Los parámetros para configurar un VDF puede oscilar desde unos pocos parámetros básicos hasta cientos. Sin embargo los obligatorios son definidos por la red eléctrica, los datos de placa del motor y la aplicación. Entre ellos destaca: Tensión base o tensión nominal del motor, Frecuencia nominal del motor, frecuencia mínima y máxima de operación del motor, modos de parada del motor , modos de entrada (cableado de entrada), modos de salida (relés de alarmas), tiempo de aceleración y desaceleración del motor y frecuencias restablecidas activadas por señales digitales.
Para el calculo de los tiempos de aceleración y desaceleración se requiere conocer el torque del motor (T_m), el torque de la carga (T_r), la inercia total del sistema mecánico J, la velocidad angular de inicio (ω_i), la velocidad angular final (ω_f) y el delta de tiempo (Δt) que es el tiempo que se tarda entre ambas velocidades angulares. Por lo tanto el torque de aceleración (T_A) debe ser siempre menor al $150\,\%T_m$ de lo contrario existirán problemas de sobrecarga.

$$T_A = J\frac{(\omega_f - \omega_i)}{\Delta t} + T_r \tag{A.8}$$

Si se asume que un motor de inducción puede entregar un torque máximo de $150\,\%$ de su torque, entonces las rampas de aceleración se calculan como:

$$\Delta t = J\frac{(\omega_f - \omega_i)}{1{,}5T_m - T_r} \tag{A.9}$$

El tiempo natural en que se frena la inercia esta dado por (A.10), si se requiere frenar el sistema de forma más acelerada, es necesario colocar en el VDF resistencias que disipan la potencia eléctrica entregada por el motor, el cual se encuentra operando como alternador.

$$\Delta t = J\frac{\omega_i}{T_r} \tag{A.10}$$

Por otra parte, la frecuencia de operación del VDF se selecciona según las necesidades mecánicas de la carga; por ejemplo, el motor de un reductor de velocidad debe operar a 950 revoluciones por minuto (rpm). Para determinar la frecuencia que se debe programar, se requiere conocer el deslizamiento del motor (s), sus polos (p), la frecuencia nominal (f_n), el tipo o comportamiento de la carga (torques constantes, torques variables).
Cuando el torque de la carga es constante, Figura A.11a, las velocidades de deslizamiento son contantes no importa la frecuencia en que opera el motor. Esto implica que el deslizamiento en el nuevo punto de operación(s_2) está dado por, $s_2 = s \cdot f_n/f_2$. Sabiendo que la velocidad mecánica está dada por:

$$\eta_{mec} = \frac{120f_n}{p}(1 - s) \tag{A.11}$$

entonces, la nueva frecuencia (f_2) a programar el en VDF en función de la velocidad mecánica deseada (η_{mec2}) se determina de la siguiente forma,

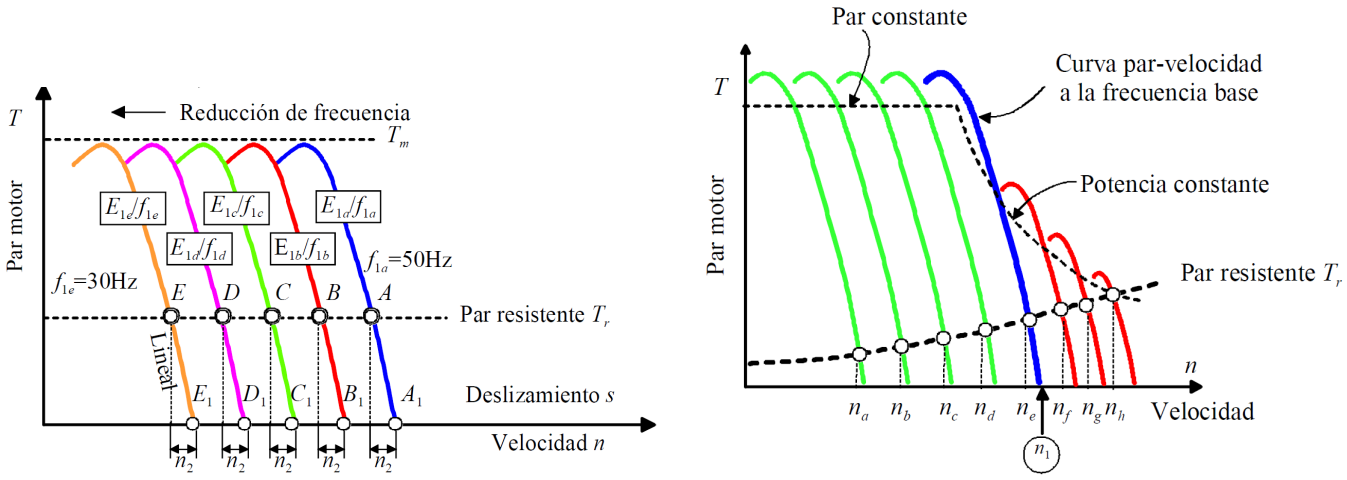
$$f_2 = \frac{p\left(\eta_{mec2} + \dfrac{120sf_n}{p}\right)}{120}. \tag{A.12}$$

Las bombas centrifugas o abanicos son ejemplos de par resistente variable, tal y como se aprecia en la figura A.11b. En estos casos la velocidad de deslizamiento varia según la carga, pero una aproximación con error menor al 1 % asume que los deslizamientos son constantes, por tanto $s_2 = s$.

A.6. Logo!

Como se indica en el manual [3], el LOGO! un módulo lógico universal para el control de procesos industriales o automatización, que incorpora funciones lógicas, temporizadores, contadores, controladores PID, funciones aritméticas, fuente de alimentación, interfaces gráficas, módulos de ampliación y el software de programación. La figura muestra un LOGO!8 24CE, de 24 voltios y salidas a transistor.

El alambrado del dispositivo esta explicado en la sección 2.3 del Manual del LOGO! [3]. Hay que tener en cuenta que dependiendo del modelo del LOGO, la forma de como se alambra la alimentación, las señales de entradas y las señales de salida podrían cambiar. Respecto a la alimentación, es importante la colocación de fusibles o varistores (MOV) dependiendo del modelo, ver pagina 43 del manual. Un ejemplo básico de un programa de control en LOGO, con el alambrado físico se muestran en la sección 3.3 del manual [3].
El LOGO es programado con el software llamado *LOGO!Soft Comfort*. En el manual del software [14], en el capítulo 3, página 164, se muestra un Tutorial del uso del programa. Por otra parte, dicho manual brinda los siguientes cinco ejemplos resueltos para su estudio:



(a) Torque carga constante (b) Torque de carga variable

Figura A.11: Curvas de torque inducido del motor y par de la carga. [13]

Figura A.12: LOGO!8 24CE.

- Bomba de agua no potable (página 219).
- Sistema de ventilación (página 235).
- Portón corredizo (página 236).
- Control de calefacción (página 238).
- Estación de llenado (página 241).

A.7. Autómatas programables

La programación de los autómatas programables ha sufrido cambios en la semántica y sintaxis de programación con la implementacion de la norma IEC 611131-3 [15] por parte de los fabricantes. La normalización de la semántica mediante la estructuración de los programas y la normalización de sintaxis mediante la implementacion de cinco lenguajes de programación estandarizados permiten la portabilidad del código entre fabricantes.
La programación estructura de código informático para autómatas programables, permite organizar y descomponer las soluciones en multiples subsistemas con distintos niveles de jerarquía. Esta nueva forma de programar estructuradamente permite ordenar y dar mantenimiento al código de forma más sencilla, así como re-utilizar código existente para otras soluciones. Al respecto la norma IEC-61131-3 [16] indica que los autómatas modernos estructuran sus programas en *Program Organisation Units* (POU) los cuales los divide en tres tipos: *Program*, *Function Block* y *Funtion*.
SIEMENS indica en su manual de Software [17], que el programa de usuario se conforma por: bloques de organización (OB), bloques de función (FB), funciones (FC), y bloques de datos (DB), la Figura A.13 muestra la relación entre los distintos bloques. Por favor leer entre las páginas 5625-5631 para una descripción detallada de los bloques.

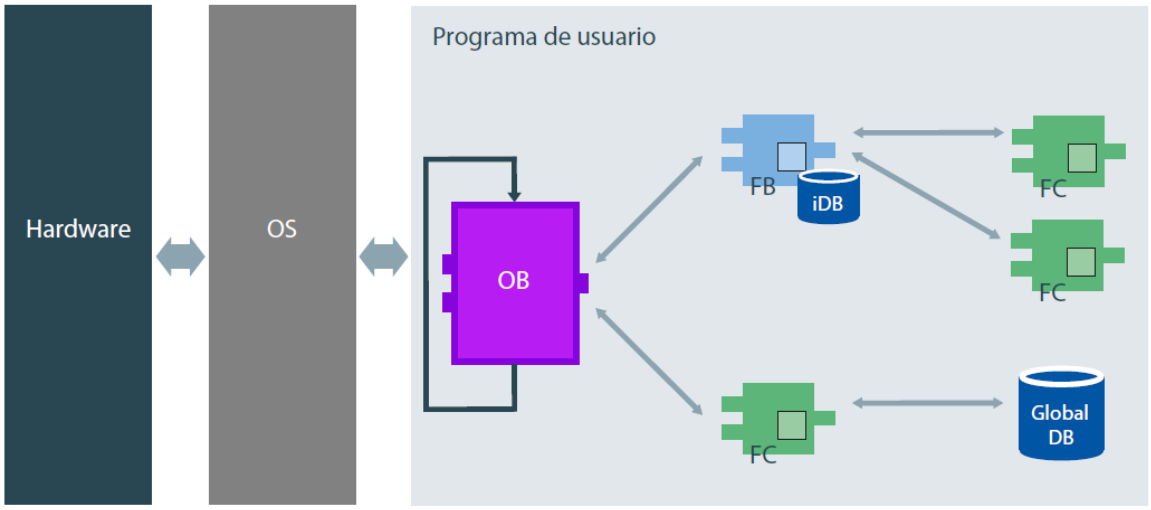


Figura A.13: Programación jerarquizadas en TIA Portal.

Cada bloque OB, FB o FC puede ser programado con varios lenguajes estandarizados, es decir un FB puede ser programado de cinco formas distintas y también es permitido mesclar distintos lenguajes en la solución jerarquizada. La Figura A.14 muestra los cinco lenguajes estandarizados por

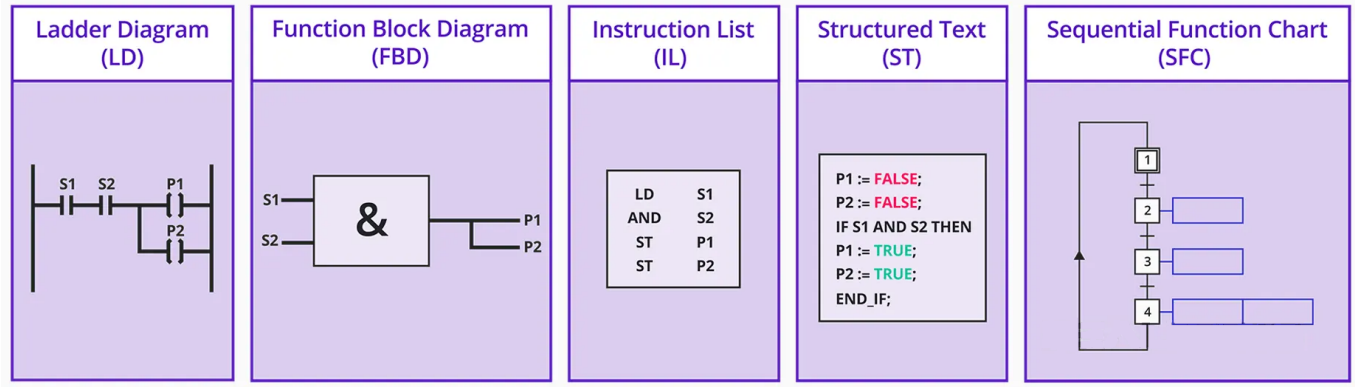


Figura A.14: Lenguajes estandarizados por la norma IEC 61131-3.

la norma: Diagrama en escalera (LD), Diagrama en Bloque de Funciones (FBD), Lista de Instrucciones (IL), Texto estructurado (ST) y Gráfico de Función Secuencial (SFC). Notese en la Figura A.14, que los primeros cuatro lenguajes muestran la respectiva implementación de una AND lógica entre las señales booleanas S1 y S2. Al respecto SIEMENS nombra dichos lenguajes como: KOP o esquemas de contactos, FUP o diagrama de funciones, AWL o lista de instrucciones, SCL o lenguaje de control estructurado y GRAPH o lenguaje de programación gráfico. La descripción de detallada de cada lenguaje, sus funciones, ejemplos, etc., se encuentran en el manual del sistema [17], para la funcionalidad del KOP incie en página 14161, para el FUP ver página 14223, para AWL inicia en la página 14283, el SCL se encuentra en la página 14333 y para el GRAPH ver la página 14411.
Los diagramas electricos de conexión del CPU S7-1500, de sus módulos de entrada y salida se muestran en el manual del hardware [18]. La creación de un proyecto con el Software TIA portal se encuentra en la página 1280 de dicho manual, sin embargo, un video informativo con dicho procedimiento esta disponible en [19].

Apéndice B

Repositorio de código

B.1. Código actividad 1

```
/*
Instituto Tecnológico de Costa Rica
Laboratorio de Control Eléctrico.
Lab #1: Introducción a Arduino
Este programa prende el LED cuando se presiona el botón.
*/

// Declaraciones de entradas y salidas
const int BUTTON = 2;  // Boton conectado al pin 2
const int LED = 4;     // LED conectado al pin 4

// Configuración
void setup() {
    pinMode(LED, OUTPUT);    // configuramos el pin del LED como salida
    digitalWrite(LED, HIGH);  // Encendemos el LED
    pinMode(BUTTON, INPUT);  // configuramos el pin del botón como entrada
}
//  Programa principal
void loop() {
    bool buttonState = !digitalRead(BUTTON);  // leemos el estado del botón
    digitalWrite(LED, buttonState); // Actualizamos el estado del LED
}
```

B.2. Código actividad 2

```
/*
Instituto Tecnológico de Costa Rica
Laboratorio de Control Eléctrico.
Lab #1: Introducción a Arduino
Implementación de funciones lógicas AND, OR, XOR, NAND, NOR de 2 entradas
*/


// Declaracione de constantes
  const int PinEntrada[2]={2,3};
  const int PinSalidas[5]={4,5,6,7,8};
  const int PinAnalogico[2]={A0,A1};
  int ValorAnalojLeido[2]={0,0};
  float ValorVoltage[2]={0.0,0.0};
  boolean Mapa_entradas[2];
  boolean ResultadoLogico[5]={false, false, false, false, false};

// Configuracion de Pines de entrada y salida
void setup(){
  Serial.begin(115200);
```

```
  Serial.println("-----");
// Initializa los pines:
for (int i = 0; i < 2; i++) {
  pinMode(PinEntrada[i], INPUT);
}
for (int i = 0; i < 5; i++) {
  pinMode(PinSalidas[i], OUTPUT);
}
delay(2);
}

void loop()
{
    //Lectura de entradas
    for(int i=0; i < 2; i++){
        Mapa_entradas[i] = digitalRead(PinEntrada[i]);
        ValorAnalojLeido[i] = analogRead(PinAnalogico[i]);
        ValorVoltage[i]= (map(ValorAnalojLeido[i], 0, 1023, 0, 500)/100.0);
    }
    //Escritura de salidas
    for(int i=0; i < 5; i++){
        digitalWrite(PinSalidas[i],ResultadoLogico[i]);
    }
//EJECUCION DEL PROGRAMA
  ResultadoLogico[0]=AND(Mapa_entradas[1],Mapa_entradas[0]);
  ResultadoLogico[1]=OR(Mapa_entradas[1],Mapa_entradas[0]);
  ResultadoLogico[2]=XOR(Mapa_entradas[1],Mapa_entradas[0]);
  ResultadoLogico[3]=NAND(Mapa_entradas[1],Mapa_entradas[0]);
  ResultadoLogico[4]=NOR(Mapa_entradas[1],Mapa_entradas[0]);

//IMPRESION DE RESULTADOS ENTRADA Y SALIDAS

  Serial.print(Mapa_entradas[0]);
  Serial.print(",");
  Serial.print(ValorVoltage[0]);
  Serial.print(",");
  Serial.print(Mapa_entradas[1]);
  Serial.print(",");
  Serial.print(ValorVoltage[1]);
  Serial.print(",");
  Serial.print(ResultadoLogico[0]);
  Serial.print(",");
  Serial.print(ResultadoLogico[1]);
  Serial.print(",");
  Serial.print(ResultadoLogico[2]);
  Serial.print(",");
  Serial.print(ResultadoLogico[3]);
  Serial.print(",");
  Serial.println(ResultadoLogico[4]);
  delay(10);
}

//DEFINICION DE LAS FUNCIONES LÓGICAS
// Forma de programacion Booleana
bool AND (bool X, bool Y ){
  return (X & Y);
}

// Forma de programacion con estructuras de control
bool OR (boolean X, bool Y ){
  return (X | Y);
}

// Completar código
bool XOR (bool X, bool Y ){
  return (X ^ Y);
}

bool NAND (bool X, bool Y ){
  return !(X & Y);
```

```
}

bool NOR (bool X, bool Y ){
  return !(X | Y);
}
```

Bibliografía

[1] S. Electric, "Wiring diagram book," 2023. Accessed: 02/03/2023.

[2] SIEMENS, "Manual de producto - arrancadores suaves sirius 3rw30/3rw40," 2023. Accessed: 20/03/2023.

[3] SIEMENS, "Manual de sistema - logo! 8 6ed1052-xxx08-0ba1," 2023. Accessed: 13/04/2023.

[4] M. Margolis, B. Jepson, and N. Weldin, *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. O'Reilly Media, 2020.

[5] E. Corp., "Definite purpose motor control," 2023. Accessed: 02/03/2023.

[6] E. Corp., "Relés de protección térmica: C396a2ap05seldc," 2023. Accessed: 02/03/2023.

[7] S. Electric, "Class 9065 catalog for overload relays and thermal unit selection," 2023. Accessed: 02/03/2023.

[8] S. Electric, "Rele temporizador 8 funciones 0.05...1s -24vac/dc," 2023. Accessed: 09/03/2023.

[9] S. Electric, "Temporizadores re22: Descripcion," 2023. Accessed: 09/03/2023.

[10] S. J. Chapman, *Máquinas Eléctricas*. McGraw-Hill, 5 ed., 2011.

[11] J. Posada Contreras, "Modulación por ancho de pulso (pwm) y modulación vectorial (svm)," 2005. Accessed: 02/03/2023.

[12] A. Bradley, "Manual de usuario, 160 ssc serie b," 2023. Accessed: 20/03/2023.

[13] J. F. Mora, *Máquinas Eléctricas*, vol. 5. McGraw-Hill, 2008.

[14] SIEMENS, "Ayuda en pantalla de logo!soft comfort," 2023. Accessed: 13/04/2023.

[15] "Iec 61131-3:2023. programmable controllers - part 3: Programming languages," 2023. Accessed: 19/04/2023.

[16] M. Tiegelkamp and J. Karl-Heinz, *IEC 61131-3: Programming Industrial Automation Systems*, vol. 166. Berlin/Heidelberg, Germany: Springer, 2010.

[17] SIEMENS, "Simatic step 7 basic/professional v18 y simatic wincc v18," 2023. Accessed: 19/04/2023.

[18] SIEMENS, "Simatic s7-1500/et 200mp manual collection," 2023. Accessed: 19/04/2023.

[19] M. Pérez-Cabezas, "Configurar, detectar dispositivos y guardar librería en tia portal," 2023. Accessed: 20/04/2023.