

Automatización Avanzada (37800)
Máster en Automática y Robótica

Práctica 1.
Programación y Simulación de un PLC



Francisco Andrés Candelas Herías
Grupo de Innovación Educativa en Automática



Universitat d'Alacant
Universidad de Alicante

© 2011 GITE – IEA

Práctica 1. Programación y Simulación de un PLC

1. Objetivos

- Conocer como se configura y utiliza un entorno de programación de PLCs.
- Saber cómo se organizan las variables y programas en un PLC.
- Conocer lenguajes y técnicas de programación de PLCs.
- Se capaz de realizar un programa mediante los lenguajes LD, ST, y FBD.
- Saber utilizar los bloques de funciones disponibles para programar un PLC.
- Se capaz de crear nuevos bloques de funciones.
- Saber cargar un programa en un PLC, así como ejecutarlo y depurarlo.

2. Programación básica con UnityPro

2.1. Cuestiones obligatorias

- 1) Abrir la herramienta UnityPro y crear un nuevo proyecto, utilizando un bastidor como el explicado en el manual de prácticas de introducción a UnityPro.
- 2) Configurar los canales del módulo de E/S digitales para que las salidas 0 a 3 tengan los valores de retorno 0, y las salidas 5 a 7 los valores de retorno 1.
- 3) Realizar la programación mediante diagrama de contactos (LD) de un motor que se accione de la siguiente forma:
 - Dispone de dos pulsadores (el de MARCHA y el de PARO) y de dos interruptores (el de marcha IZQUIERDA y el de marcha DERECHA).
 - El motor se pone en funcionamiento con el pulsador de MARCHA y con uno de los dos interruptores de marcha, y se para únicamente con el pulsador de PARO, que será prioritario sobre el resto de órdenes.
 - Físicamente, el pulsador de PARO es normalmente cerrado por seguridad de máquina.
 - En el caso de que se intente activar el funcionamiento del motor con los dos interruptores de marcha, la prioridad la tendrá la marcha IZQUIERDA.

Los pulsadores e interruptores MARCHA, PARO, IZQUIERDA y DERECHA están conectados a las entradas %IO.1.0, %IO.1.1, %IO.1.2 y %IO.1.3 respectivamente. El motor se controla con un relé conectado a las salidas %Q0.1.16 y %Q0.1.17: la primera indica la activación del motor (a 1) y la segunda el sentido de giro (0: izquierda, 1:derecha).
- 4) Cargar y ejecutar el proyecto creado en los pasos anteriores en el simulador del PLC, y comprobar que funciona correctamente forzando los valores de las entradas.

2.2. Cuestiones opcionales

- 5) Tratar de representar el programa plateado en la cuestión 3 usando también los lenguajes de diagrama de bloques funcionales (FDB) y de texto estructurado (ST).
- 6) Crear un proyecto para resolver el automatismo del “tostador de bollos” visto en clase de teoría (Figura 1) y descrito a continuación.

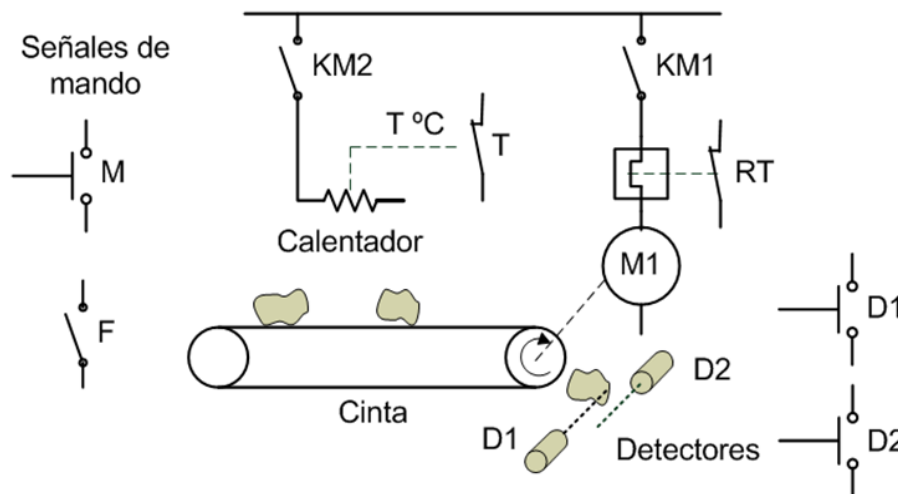


Figura 1. Esquema del tostador de bollos.

Al pulsar (y soltar) el botón de marcha M, se conecta el relé KM1 para activar el motor de la cinta transportadora. El motor se para cuando uno de los detectores laser (D1 o D2) se activa cerrando sus contactos. Aunque el detector abra sus contactos de nuevo, el motor seguirá parado hasta volver a pulsar. Además, el motor (KM1) solo funciona si el relé térmico RT tiene el contacto cerrado.

La resistencia de calentamiento (KM2) funciona cuando el motor está en marcha y el interruptor de paro F está abierto. Para la resistencia de calentamiento funcione, el termostato T debe estar cerrado.

3. Programación con bloques

3.1. Arranque de motores AC asíncronos

Cuando se trabaja con motores AC síncronos de poca potencia, su activación se puede realizar sencillamente con un solo contactor o relé, además de los equipos de protección. Pero cuando se requiere utilizar un motor de más potencia, generalmente trifásico, que proporcione un gran par, se requiere un circuito de conexión más complejo para controlar la gran demanda de corriente que es necesaria en el arranque del motor para conseguir un par de arranque suficiente. Entre las técnicas para controlar la corriente de arranque, la más usada es la conocida como estrella-triángulo por ser sencilla y barata.

Un motor AC asíncrono trifásico de seis polos está constituido por tres bobinas, como muestra la Figura 2, las cuales se pueden cablear de dos modos para ser conectadas a una toma de red trifásica. Estos modos, mostrados en la Figura 2 se denominan conexión en estrella y conexión en triángulo por las formas que adopta el cableado.

En la conexión en estrella, la diferencia de potencial entre cada par de líneas de la alimentación trifásica, denominadas fases, alimenta dos bobinas, mientras que en la conexión en triángulo, la tensión entre cada dos fases solo alimenta una bobina. Así, en la

configuración de estrella, las bobinas consumen menos corriente para hacer girar al motor, aunque éste gire a menos revoluciones. En cambio, la configuración en triángulo es adecuada para conseguir mayores revoluciones y par, aunque su consumo de corriente es alto, especialmente durante el arranque del motor.

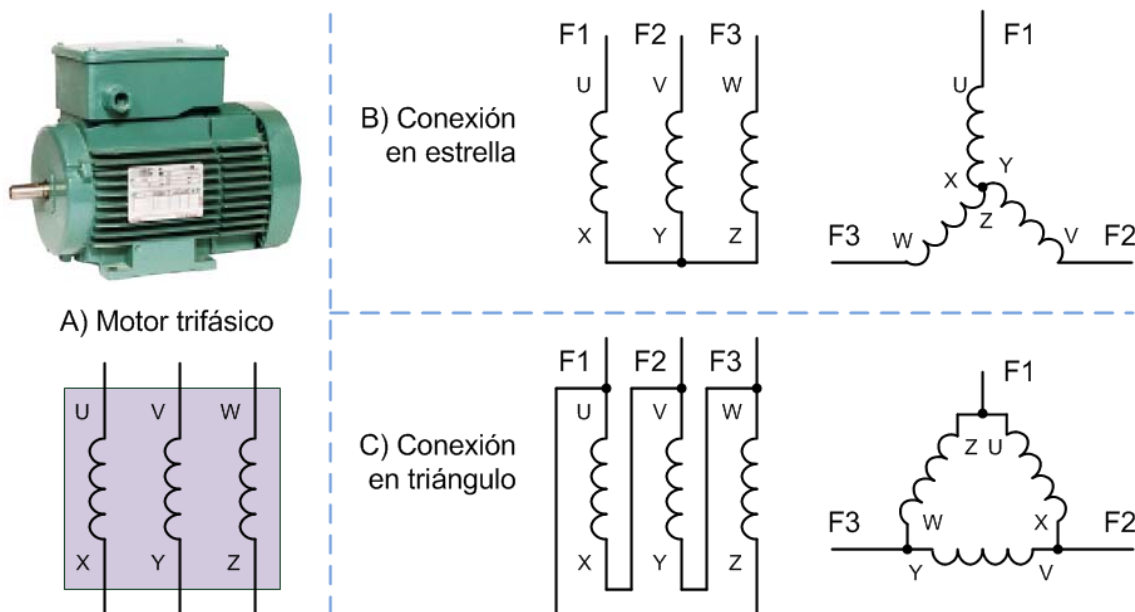


Figura 2. Conexiones de un motor trifásico.

Se puede utilizar las dos configuraciones aprovechando las ventajas de cada una. Para ello, el motor debe arrancar con una configuración de estrella, y después cambiar a una de triángulo cuando alcance un régimen de funcionamiento estable. De este modo la corriente de arranque se reduce, mientras que, en régimen de funcionamiento normal, el motor puede alcanzar su máximo rendimiento. Se suele considerar que el cambio debe realizarse cuando el motor alcanza el 80% de su velocidad nominal. El cambio de configuración se realiza conmutando las conexiones de las bobinas mediante contactores, como muestra el esquema de potencia de la Figura 3.

En el esquema de la Figura 3, el contactor KM1 controla el encendido y apagado del motor, el contactor KM2 se debe cerrar para una conexión en estrella, y el contactor KM3 conecta las bobinas en triángulo. Este esquema también incluye otros elementos, como el guarda-motor RT1 que protege el motor de corrientes excesivas y cortocircuitos, el sensor SEN 1 que mide la corriente consumida por el motor, y una transformación para obtener una alimentación de 220V, a partir de la alimentación trifásica de 380V, que sirve para el circuito de mando.

Es importante tener en cuenta que KM2 y KM3 no deben activarse a la vez ya que esto provocaría cortocircuitos en las fases. Existen modelos de contactores especiales para cablear conexiones estrella-triángulo, que incluyen todos los contactos necesarios para KM2 y KM3, y que impiden la activación de ambos a la vez. Pero si KM2 y KM3 son contactores independientes, es el circuito de mando el que debe controlar que no se activen a la vez. Entonces, lo más habitual es realizar una pequeña pausa entre la desactivación de KM2 (triángulo) y la activación de KM3 (estrella) mediante un temporizador.

Si el circuito de mando se basa en un PLC, se puede tener un esquema como el mostrado en la Figura 4. En este caso, el PLC se encarga de actuar sobre los contactores KMA, KM2 y KM3 para arrancar el motor y controlar su marcha.

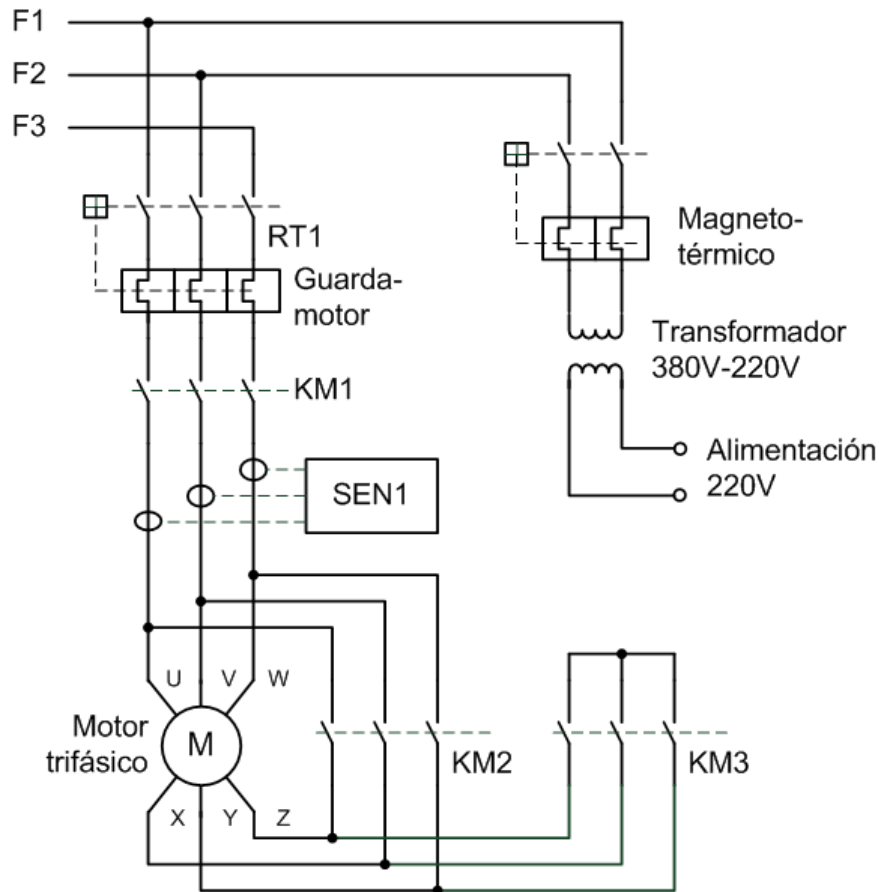


Figura 3. Esquema del circuito de potencia para un motor trifásico.

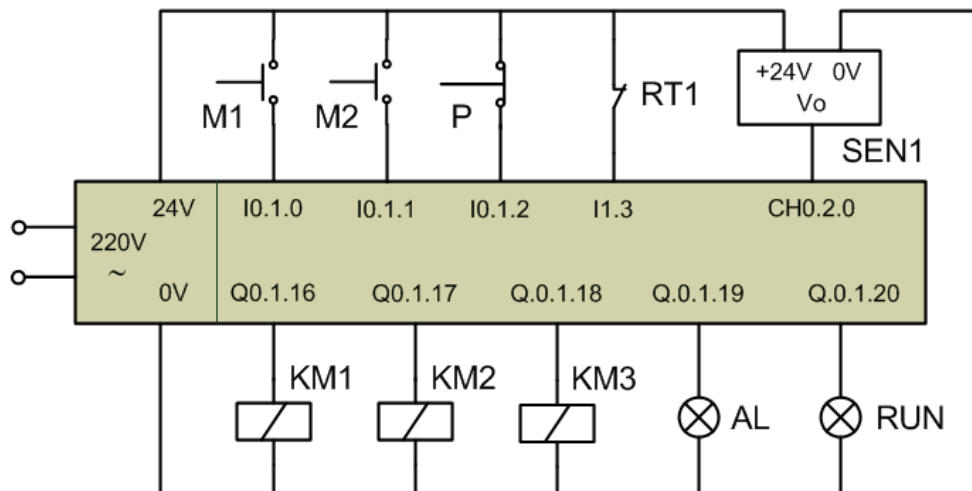


Figura 4. Esquema de mando con un PLC.

3.2. Cuestiones obligatorias

Se debe desarrollar un programa para el PLC Modicom M340 para controlar la puesta en marcha y mantenimiento de un motor asíncrono trifásico de corriente alterna con arranque estrella-triángulo, cumpliendo ciertas medidas de seguridad. El esquema de

potencia es el mostrado en la Figura 3, mientras que el esquema de mando, con las conexiones al PLC, se muestra en la Figura 4.

- 7) Crear un nuevo proyecto, con la configuración de CPU y bastidor usadas en la prácticas
- 8) Implementar un bloque de función denominado **MotorET**, programado en diagrama de bloques de funciones (FBD), que controle el arranque y funcionamiento del motor según estas especificaciones:
 - Entradas:
 - **e_marcha**: se pone a 1 para indicar que se debe poner en marcha el motor.
 - **e_paro**: se pone a 1 para indicar que se debe parar el motor.
 - **e_habilitado**: se pone a 1 para indicar que el motor puede estar en marcha.
 - Salidas:
 - **s_linea**: es la salida que se representa el relé que pone en marcha el motor.
 - **s_estrella**: es la salida que se representa el relé que activa la configuración de estrella.
 - **s_triángulo**: es la salida que se representa el relé que activa la configuración de triángulo.
 - La salida **s_linea** se pondrá a 1 cuando la entrada **e_marcha** se ponga a 1, y se pondrá a 0 cuando la entrada **e_paro** pase a ser 1.
 - Al activarse la salida, el motor debe arrancar con una configuración en estrella de sus bobinas, que se mantendrá durante **3s**, para lo que se activa la salida **s_estrella**. Posteriormente se cambiará a la configuración de triángulo activando **s_triángulo** y desactivando **s_estrella**.
 - Por seguridad, debe haber un espacio de tiempo de **200ms** entre la desactivación de la configuración en estrella y la activación de la configuración de triángulo. Las temporizaciones se pueden resolver con una función de temporización como TP (pulso), TON o TOFF.
 - En todo momento, la salida **s_linea** solo estará activa (1) sólo si la entrada **e_habilitado** vale 1.
- 9) Implementar una sección denominada **SensorI**, programado en texto estructurado (ST), que vigile la entrada analógica del sensor de corriente SEN1 y funcione de esta manera:
 - La entrada analógica debe configurarse en el rango 0-10V, que corresponden a corrientes de **0 a 100A**.
 - Si se detecta que SEN1 mide más corriente que un valor límite de corriente durante un tiempo continuado de más de **500ms**, se debe poner a 1 la variable booleana en memoria **alarma_corriente**. El valor límite de corriente estará almacenado en la variable de tipo real **corriente_limite**, que inicialmente valdrá 20,0A.
 - Esta sección se debe ejecutar cada 50ms. Esto se puede hacer mediante una tarea de evento de temporizador, o mediante una sección que se habilite sólo cuando acaba un temporizador arrancado en otra sección del programa. En el segundo caso, el temporizador se puede iniciar de nuevo en la propia sección **SensorI**.
 - Se debe guardar en la variable real **corriente_max** el máximo valor de corriente leído durante la ejecución de esta sección. Inicialmente el valor de esa variable será 0.

- 10) Implementar una sección de la tarea maestra que, programado en diagrama de contactos (LD), que haga uso de la función **MotorET** creada anteriormente para controlar el motor según las líneas de entrada y salida del PLC (ver Figura 4), cumpliendo que:
 - Inicialmente, al arrancar el PLC, el motor debe estar parado, con los contactores **KM1**, **KM2** y **KM3** abiertos, y los pilotos **AL** y **RUN** apagados.
 - Por seguridad, el motor se pondrá en marcha cuando se pulsen los botones de marcha **M1** y **M2** a la vez.
 - Se considerarán condiciones de paro que se actúe sobre el pulsador de paro **P** (normalmente cerrado) o que el guardamotor **RT1** corte la alimentación del motor.
 - Para hacer girar el motor, las salidas del bloque **MotorET** deben actuar sobre los contactores **KM1**, **KM2** y **KM3**.
 - El funcionamiento del motor solo debe estar permitido cuando la variable **alarma_corriente**, establecida en la sección **SensorI**, valga a 0. Esta variable se pondrá a 0 cuando o se pulse **P** o se inicie el programa tras un arranque en frío.
 - El piloto de alarma **AL** se debe encender cuando la variable **alarma_corriente** vale 1, o cuando el guardamotor **RT1** ha parado el motor. Este piloto sólo debe apagarse cuando se pulsa el pulsador **P**.
 - El piloto de ejecución **RUN** debe estar encendido cuando el motor está en marcha y girando con una configuración de estrella o triángulo.
- 11) Ejecutar el funcionamiento del programa sobre el PLC simulado para comprobar que funciona según las especificaciones anteriores. Para simular la entrada analógica procedente del medidor SEN1, es mejor crear una nueva variable entera en memoria (no asociada a una E/S) y usar está en vez de la entrada analógica, ya que así se podrá modificar fácilmente su valor durante la ejecución. También es conveniente probar el programa por partes, antes de ejecutarlo al completo.

3.3. Cuestiones opcionales

- 12) En vez de requerir simplemente que se pulsen los botones de marcha **M1** y **M2** a la vez para poner el motor en marcha, cambiar el programa para que el motor se active solo cuando **M1** y **M2** se pulsan con un margen de tiempo de **0,2s como máximo**, esto es, si se pulsa primero un botón, el otro deberá ser pulsado como mucho 0,2s después. Esto se puede resolver con una función de temporización como TP (pulso), TON o TOFF.
- 13) Implementar una sección de la tarea maestra denominada **Cuentas** mediante texto estructurado, que lleve la cuenta de las veces que se arranca el motor, y que almacene esa cuenta en la variable de memoria **cuenta_arranques** de tipo entero. Esta variable debe iniciarse a 0 en la tabla de variables.
- 14) Cambiar el programa para que, el cambio de estrella a triángulo en el arranque no se haga tras un tiempo fijo de **5ms**, sino que se realice cuando la corriente del motor sea inferior a un valor que estará almacenado en la variable real **corriente_nom**, que inicialmente valdrá 10,0A.
- 15) Hacer que el piloto de alarma **AL** se encienda de modo constante cuando la variable **alarma_corriente** vale 1, y que parpadee cuando el guardamotor **RT1** ha parado el motor. El piloto sólo debe apagarse cuando se pulsa el pulsador **P**.