

- 1.- Programación lineal y programación modular.
- 2.- Funciones, FC.
- 3.- Bloques de Funciones, FB.

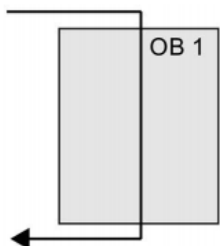
## 1.- Programación lineal y programación modular.

### Seleccionar el tipo de estructura del programa de usuario

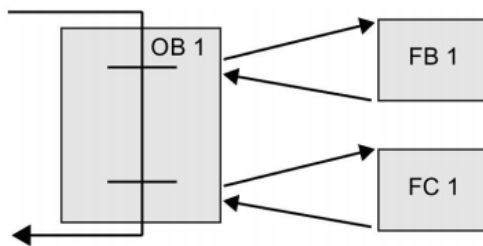
Según los requisitos de la aplicación, es posible seleccionar una estructura lineal o modular para crear el programa de usuario:

- Un programa lineal ejecuta todas las instrucciones de la tarea de automatización de forma secuencial, es decir, una tras otra. Generalmente, el programa lineal deposita todas las instrucciones del programa en el OB encargado de la ejecución cíclica del programa (OB 1).
- Un programa modular llama bloques de función específicos que ejecutan determinadas tareas. Para crear una estructura modular, la tarea de automatización compleja se divide en tareas subordinadas más pequeñas, correspondientes a las funciones tecnológicas del proceso. Cada bloque lógico provee el segmento del programa para cada tarea subordinada. El programa se estructura llamando uno de los bloques lógicos desde otro bloque.

Estructura lineal:



Estructura modular:



Creando bloques lógicos genéricos que pueden reutilizarse en el programa de usuario, es posible simplificar el diseño y la implementación del programa de usuario. La utilización de bloques lógicos genéricos ofrece numerosas ventajas:

- Es posible crear bloques lógicos reutilizables para tareas estándar, tales como el control de una bomba o motor. También es posible almacenar estos bloques lógicos genéricos en una librería, de manera que puedan ser utilizados por diferentes aplicaciones o soluciones.
- El programa de usuario puede dividirse en componentes modulares para las tareas funcionales, facilitando así su comprensión y gestión. Los componentes modulares ayudan no sólo a estandarizar el diseño del programa, sino que también pueden facilitar y agilizar la actualización o modificación del código del programa.
- La creación de componentes modular simplifica la depuración del programa. Dividiendo el programa completo en segmentos de programa modulares, es posible comprobar las funciones de cada bloque lógico a medida que se va desarrollando.
- La creación de componentes modulares para las distintas funciones tecnológicas permite simplificar y reducir el tiempo de puesta en marcha de la aplicación.

## 2.- Funciones. FC

Una función contiene un programa que se ejecuta cada vez que la función es llamada por otro bloque lógico.

Las funciones (FC) son como unas **subrutinas sin memoria**. Los datos de las variables temporales se pierden tras haberse procesado la función. Los datos temporales no se almacenan. Para almacenar los datos de forma permanente es preciso asignar el valor de salida a una posición de memoria global, p. ej. al área de marcas o un DB global.

Las funciones se pueden utilizar p. ej. para los siguientes fines:

- Contienen instrucciones a modo de subrutinas.
- Devolver valores de función al bloque que llama, p. ej. en funciones matemáticas.
- Ejecutar funciones tecnológicas, p. ej. controles individuales con operaciones lógicas binarias.

Una FC no tiene ningún bloque de datos instancia asociado (DB). La FC usa la pila de datos locales para los datos temporales utilizados para calcular la operación.

Una función se puede llamar varias veces en diferentes puntos de un programa. Esto facilita la programación de funciones complejas que se repiten con frecuencia.

## 3.- Bloques de Funciones. FB

Los bloques de función como unas **subrutinas con memoria**.

Los bloques de función son bloques lógicos que depositan sus valores de forma permanente en bloques de datos de instancia, de modo que siguen disponibles tras procesar el bloque.

Los parámetros de entrada, salida y entrada/salida se depositan de forma permanente en **bloques de datos de instancia**. Gracias a ello, continúan disponibles tras el procesamiento del bloque. Por este motivo, se conocen también como bloques con memoria.

Los bloques de función se utilizan en tareas que no se pueden realizar con funciones:

- Cuando hay que almacenar información en el programa.
- Cuando son necesarios temporizadores y contadores.
- Por ejemplo, una preselección del modo de operación con un pulsador.

Un bloque de función también se puede llamar varias veces en diferentes puntos de un programa. Esto facilita la programación de tareas complejas que se repiten con frecuencia.

Parámetros de los bloques:

- **Input, Output, InOut y Ret\_Val:** estos parámetros definen las variables de entrada, las variables de salida y el valor de retorno del bloque lógico. El nombre de la variable introducida en este punto se emplea de forma local durante la ejecución del bloque lógico. Normalmente, no se emplea el nombre de variables globales en la tabla de variables.
- **Static (solo FB; la figura de arriba corresponde a una FC):** el bloque lógico utiliza variables estáticas para almacenar resultados intermedios estáticos en el bloque de datos de instancia. El bloque retiene datos estáticos hasta que se sobrescriben, lo cual puede ocurrir después de varios ciclos. Los nombres de los bloques, que este bloque llama como multiinstancia, también se almacenan en los datos locales estáticos.
- **Temp:** estos parámetros son variables temporales que se emplean durante la ejecución del bloque lógico.
- **Constant:** son valores constantes con nombre para el bloque lógico.

### **Proyecto 5. Utilización de FC y FB. Guardar FB en librerías**

- 1) Realiza un FC para el cálculo de los mm recorridos por una rueda la cual es movida por un encoder. Indica las entradas y las salidas. *Nombre: Calculo\_mm*
- 2) Realiza un FB para el cálculo de los mm recorridos por una rueda la cual es movida por un encoder. Indica las entradas y las salidas. *Nombre: Calculo\_mm*
- 3) Realiza un FB para un arranque estrella-triángulo (ojo al temporizador, es multiinstancia). Indica las entradas y las salidas Utiliza el FB dos veces y verifica el funcionamiento. *Nombre: Arranque\_EstTri*
- 4) Realiza un FB para un telerruptor (lo realizamos en un proyecto anterior).
- 5) Realiza un FB para implementar el normalizado y escalado de la entrada analógica. El FB debe ser lo más genérico posible y servir para entradas en tensión o corriente y para cualquier valor (entre 0 .. 10V) o (0 .. 20mA). *Nombre: Analog\_IN*
- 6) Realiza un FB para implementar el normalizado y escalado de la salida analógica. El FB debe ser lo más genérico posible y servir para salidas en tensión o corriente y para cualquier valor (entre 0 .. 10V) o (0 .. 20mA). *Nombre: Analog\_OUT*
- 7) Crear un Librería Global y agregar en ella los FB creados. <https://plc-hmi-scadas.com/100.php>