

Funciones

en PLC

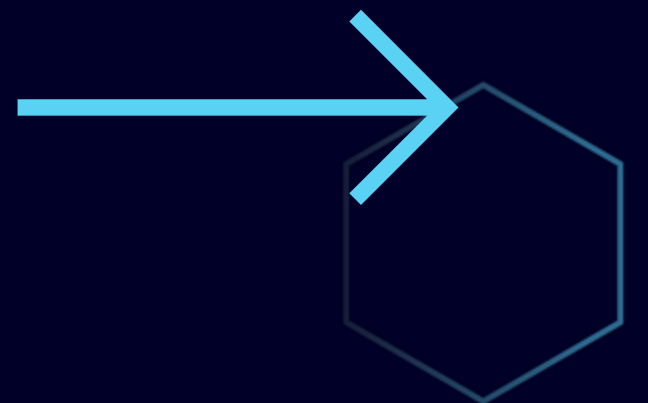


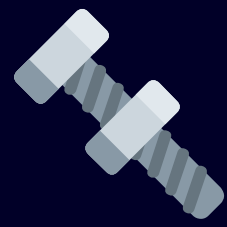
Tipos de funciones,
implementación y adaptación



⚠ Post técnico, abre tu mente...

Empecemos





Funciones



Tipos

● SÍNCRONAS

- Funciones que devuelven un resultado en el mismo ciclo de scan.

● ASÍNCRONAS

- Funciones que devuelven un resultado después de varios ciclos de scan.

● CALLBACKS

- Funciones que son pasadas como parámetro a otras funciones.

● LAMBDA O ANÓNIMAS

- Funciones sin nombre que no son definidas explícitamente, son “desechables” y para una acción concreta.

Síncronas

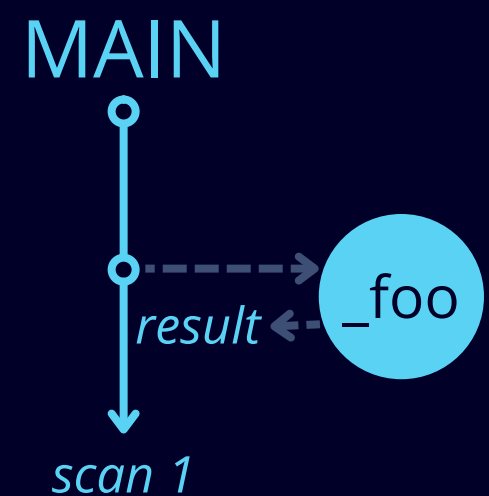


Síncronas

- Se ejecutan en el mismo ciclo de scan.
- Se pueden implementar en FC o FB.
- Debe ser para tareas específicas.

✓ VENTAJAS:


- Simplicidad.
- Fácil de depurar.
- Bajo overhead.



⚠ DESVENTAJAS:

- Desperdicio de recursos si se implementa en FB.
- Posible paro de la CPU si se anidan muchas funciones.

📌 EJEMPLO:

- ✓ `_inPosition(a, b, c): bool`
-  `_inPositionAndMove(a, b, c): {bool, real}`

Funciones Asíncronas



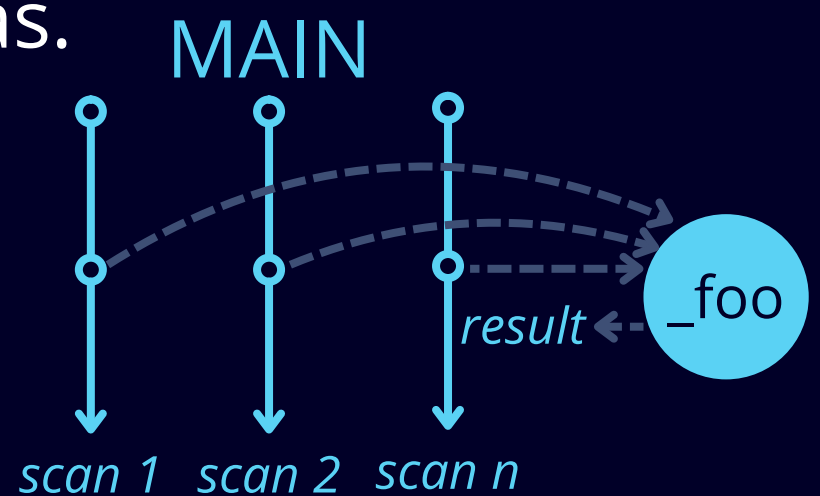


Asíncronas

- Se ejecutan en varios ciclos de scan.
- Se pueden implementar en FC o FB.
- Debe ser para tareas específicas.

✓ VENTAJAS:

- No bloquea el ciclo de scan.
- Mejora el uso de recursos.
- Ideal para tareas largas no críticas.



⚠ DESVENTAJAS:

- Complejidad al llevar estados.
- Depuración más difícil.
- Riesgo de caer en puntos muertos.

📌 EJEMPLO:

- ✓ `_timer(in, timeout): bool`
- ⚡ `_timerAndCount(in, timeout): {bool, int}`

Callbacks



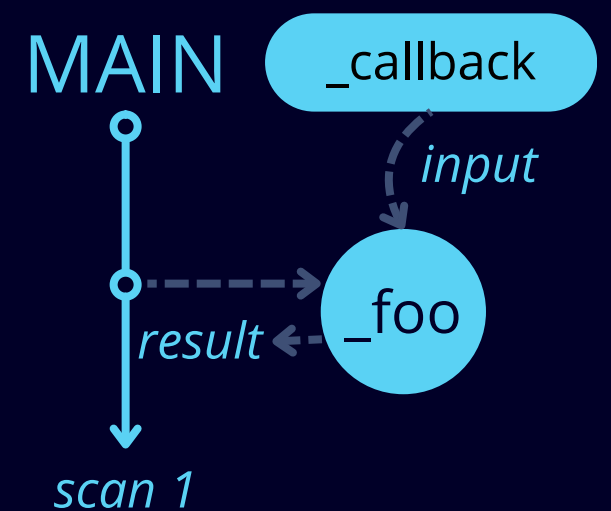


Callbacks

- Se pasan como parámetro a una función.
- Función con parámetros implícitos.
- Existe solo en PLC 300/400.

✓ VENTAJAS:

- Reutilización y especialización.
- Abstracción de problemas.
- Extiende las funcionalidades de una función.



⚠ DESVENTAJAS:

- Complejidad para pasar parámetros.
- Depuración más difícil.
- Fácil de implementar con FC, complejo con FB.

📌 EJEMPLO:

- ✓ `_callback() → _foo(a, b, _callback): bool`
- ✗ `_callback(x, y) → _foo(a, b, _callback): bool`

Funciones Anónimas





Lambdas o Anónimas

- Se declaran “in line” en el código.
- Son concisas y directas.
- Devuelven un valor.

✓ VENTAJAS:

- Código más declarativo.
- Reducción de boilerplate.
- Ideal para transformación de datos.

⚠ DESVENTAJAS:

- Solo disponible en CODESYS (Beckhoff, Wago) y parcialmente en PLCs B&R.
- Curva de aprendizaje elevada.
- Depuración compleja.

📌 EJEMPLO:

- ✓ sumLambda: `FUNCTION(a, b): REAL := (a, b) => a + b;`

Para finalizar



¿Y TÚ?

¿Sabes cuándo utilizar cada tipo de función?

En el siguiente POST técnico trataremos la jerarquía de un programa de Automatización Industrial.

👉 ¡Debate en los comentarios!

#PLC #Software #Function

✅ Finalizado