

Coding Style para PLC

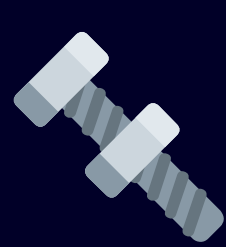
Cómo nombrar a los elementos
de un programa - Código



⚠ Post técnico, respira profundamente...

Empecemos





CÓDIGO



Conceptos clave

● INTERRUPCIONES

- Bloques de usuario donde reaccionamos a los eventos generados por el Sistema Operativo.

● SUBROUTINAS o PROCEDIMIENTOS

- Bloques usados para separar la lógica en grupos, familias, etc. Aquí va código genérico con acceso a global.

● FUNCIONES y MÉTODOS

- Bloques que realizan una función específica y puede ser reutilizada en diferentes partes del programa.

● MACROS

- Patrón que especifica cómo una secuencia de entrada debe transformarse en una secuencia de salida de reemplazo.

Interrupciones



Interrupciones



Reglas

- Snake-case
- Mayúsculas



OB1: MAIN, OB123: F_RTG1



OB001_MAIN, OB123_F_RTG1



ob_nombre_de_la_irs, Main

A veces uso esta

← nomenclatura para
ordenar la prioridad

? ¿Por qué? Para diferenciar de las subrutinas, sin más.



¡Sí! La OB1, el main, es una interrupción, es la entrada principal del programa de usuario controlada por el Sistema Operativo, tiene la prioridad más baja para poder ser interrumpida.

Subrutinas o Procedimientos





Subrutinas o Procedimientos



Reglas

- Camel-case
- Minúsculas
- Primera letra mayúscula



NombreDeLaSubrutina



nombre_de_la_sub, FC_subrutina, FB_subrutina

? ¿Por qué? A nivel semántico, las subrutinas expresan el “Título” de la agrupación.

? ¿Cuándo usar _? Para estructurar agrupaciones e instancias

OB001_MAIN:

```
├─ FC: Group1
│   ├── FB: Motor1_control, DB: MOTOR1
│   └── FB: Motor2_control, DB: MOTOR2
└─ FC: Group2
    ├── FB: Motor3_control, DB: MOTOR3
    └── FB: Motor4_control, DB: MOTOR4
```

Funciones y Métodos



⚠ ¡ Sí! una FC o FB pueden ser una subrutina.



Funciones y Métodos



Reglas

- Camel-case
- Minúsculas
- Empieza con _



_nombreDeLaFuncion



nombre_de_la_Func, FC_NOM_FUNC

? ¿Por qué? _ como prefijo es para diferenciar claramente a una función del resto del programa.

? ¿Cuándo usar _ en otra posición? Es para estructurar agrupaciones.

OB001_MAIN:

└ FC: Group1_control

└ FB: Motor1_control, DB: MOTOR1

└ FC: _add(#a, #b)

└ FB: _ackn(#i, #o), Inst: #ackn

Macros



⚠ ¡ Sí! una FC o FB pueden ser una función.

Macros

⚠ Los PLC no tienen macros, si usas el termino macro, quizás no sea correcto.

Una macro es una pieza de código que se “COPIA y PEGA” donde la macro es llamada, eso se llama EXPANSIÓN.

Existen dos clases de expansión:

1. En compilación (ensablador, c, c++)
2. En ejecución (lisp)

Todo lo que se salga de esto pasa a ser una función.

Para finalizar



¿Y TÚ?

¿Cómo diferencias a los diferentes tipos de bloque?

En el siguiente POST técnico entraremos de lleno en funciones, los tipos de funciones y cómo implementarlas en FC, FB, add-ons.

👉 ¡Debate en los comentarios!

#PLC #CodingStyle

✅ Segunda parte