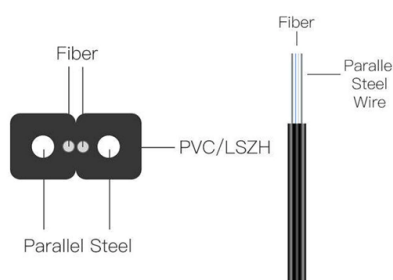


How to set interface parameters for an FC block



Overview

Passing parameters to blocks FBs and FCs receive parameters through the IN and IN/OUT interface types. The blocks process the parameters and Passing parame.

Key Takeaways

Interface parameters for a Function (FC) block are defined using IN, OUT, and IN/OUT types, with simple data passed by value and complex data passed by reference to optimize memory usage. Understanding FC Parameters

Function blocks (FCs) in PLC programming are stateless blocks that do not retain memory between calls, unlike FBs. Therefore, all data required by an FC must be passed through its interface parameters each time it is called . FCs typically use three types of interface parameters:

- IN: Input-only parameters. Values are passed by value, meaning the FC receives a copy of the data. This is suitable for simple data types like INT, DINT, or REAL .
- OUT: Output-only parameters. The FC writes results to these variables, which are then available to the calling program.



- IN/OUT: Input and output parameters. Values are passed by reference, allowing the FC to read and modify the original variable. This is recommended for complex data types such as STRUCT, ARRAY, or STRING to avoid unnecessary memory copying .

Steps to Set Interface Parameters

- Declare Parameters in the FC: At the top of the FC programming view, define the interface tags with their data types and direction (IN, OUT, IN/OUT). For example:

```
VAR_INPUT MotorSpeed : REAL; // IN parameterEND_VARVAR_OUTPUT MotorStatus :
BOOL; // OUT parameterEND_VARVAR_IN_OUT SensorData : ARRAY[1..10] OF INT; //
IN/OUT parameterEND_VAR
```

- Call the FC from a Program: When calling the FC, map the interface parameters to actual variables in your program or tag table. For example:

```
MotorControl(MotorSpeed := SpeedSetPoint, MotorStatus => StatusFlag, SensorData
:= SensorArray);
```

- Consider Memory Usage: Use call-by-value for simple types to avoid unnecessary complexity, and call-by-reference for structured or large data to reduce memory overhead .
- Optional BLOCK_FC Parameter: In some cases, you can pass an FC itself as a parameter using the BLOCK_FC type. This allows the called block to determine which FC is being passed and implement conditional logic based on the FC number, useful for library management or legacy support .

Best Practices

- Always match the parameter type (IN, OUT, IN/OUT) with the intended use to prevent unintended data overwrites.
- For arrays, structures, or strings, prefer IN/OUT to avoid copying large amounts of data.
- Keep FCs stateless; do not rely on internal memory to retain values between calls.
- Use descriptive names for interface parameters to improve readability and maintainability. By following these steps, you can efficiently set and manage interface parameters for FC blocks, ensuring proper data flow and optimized memory usage in your PLC program .

Article Content

Configuring Fibre Channel Interfaces

Physical Fibre Channel Interfaces Cisco Nexus 5000 Series switches support up to sixteen physical Fibre Channel

Fibre Channel Overview

FC is structured as a set of hierarchical functions (Figure 2). The lowest level (FC-0) defines the physical link in the system, including

4.3 Overview of Fibre Channel (FC) SAN Protocol

FC-0 Layer: It is the lowest layer in the FCP stack. This layer defines the physical interface, media, and transmission of

Related Video Reference

This video was associated with the source search result. Verify technical details against current product documentation and project requirements.

Contact Us

Continue your research online:

Website: <https://rivomultitrade.co.za>

Technical inquiry: <https://rivomultitrade.co.za/contact.html>

This document is for preliminary research. Verify specifications against current standards, product documentation and project requirements.

