

Basic information on STRUCT

Description

The STRUCT data type represents a data structure that consists of a fixed number of components of different data types. Components of STRUCT or ARRAY data type can also be nested in a structure. Structures can be used to group data according to the process control system and to transfer parameters as one data unit.

Structure declarations that are used directly at the tag are called anonymous structures. An anonymous structure can have the following form:

Global_DB		
	Name	Data type
1	Static	
2	Porsche	Struct
3	Motor	Struct
4	Valves	Array[0..5] of Bool
5	Speed_Min	Byte
6	Speed_Max	Byte
7	Temperature	Word
8	Axes	Array[0..4] of Bool
9	Oil	Byte
10	Carriage	Struct
11	Tuning	Struct
12	BMW	Struct
13	Audi	Struct

All of the subsequent explanations refer to this structure image.

Nesting depth and potential number of structures

A nested structure contains at least one additional structure as component. In each case 65535 FCs, FBs, DBs and PLC data types can be defined. For each CPU there is a special high limit relating to the block types and the total of all blocks.

A maximum of 16K components can be defined per hierarchy level for each structure description (for example, a DB or a PLC data type (UDT)). These components can also be an ARRAY or a STRUCT data type. When a structured component is hereby derived from an explicitly declared PLC data type, there is no restriction. For each hierarchy level, 16K components of PLC data type (UDT)/SDT are supported. However, a maximum of 252 "anonymous structures" may exist within a block.

Structures (STRUCT) can be nested to a depth of 8 hierarchy levels. This nesting depth is independent of the CPU used.

Transferring a STRUCT for parameter supply

You can transfer the STRUCT data type as a parameter. You can find additional information on parameter supply with STRUCT here:

[Transferring tags of the STRUCT data type](#)

Disadvantages of anonymous structures

The components of the structured tag can be identically addressed. This is regardless of whether the declaration has been made using a PLC data type or an anonymous structure.

The following disadvantages apply when using anonymous structures:

- Increased maintenance costs: If an anonymous structure was copied multiple times, then it must also have been changed multiple times during a change.
- Anonymous structures are not compatible with PLC data types (UDT) of the same structure
- Performance, since the matching types of all structural components are checked
- Increased memory requirement: Each anonymous structure is a separate object, whose description is loaded to the AS.

Example

If you declare the tag of the data type STRUCT in a PLC data type (UDT), more usage options are available to you (see figure on left). However, you can also declare the tag directly with the data type STRUCT (see figure on right).

Declaration of the structured tag "Motor" with or without PLC data type (UDT):

Structured tag with PLC data type (UDT)			Structured tag without PLC data type (UDT)		
Global_DB			Global_DB		
	Name	Data type		Name	Data type
1	Static		1	Static	
2	Porsche	"Vehicle"	2	Porsche	Struct
3	Motor	Struct	3	Motor	Struct
4	Valves	Array[0..5] of Bool	4	Valves	Array[0..5] of Bool
5	Speed_Min	Byte	5	Speed_Min	Byte
6	Speed_Max	Byte	6	Speed_Max	Byte
7	Temperature	Word	7	Temperature	Word
8	Axes	Array[0..4] of Bool	8	Axes	Array[0..4] of Bool
9	Oil	Byte	9	Oil	Byte
10	Carriage	Struct	10	Carriage	Struct
11	Tuning	Struct	11	Tuning	Struct
12	BMW	"Vehicle"	12	BMW	Struct
13	Audi	"Vehicle"	13	Audi	Struct

See also

[Declaring tags of STRUCT data type](#)
[Overview of the valid data types](#)