

Structered Text Reference

Contents

1. Variables	2
2. Constants.....	3
3. Decisions.....	3
IF	3
IF/ELSE	4
ELSIF.....	4
CASE.....	4
4. Loops	5
FOR	5
WHILE	5
REPEAT	5
EXIT	6
CONTINUE.....	6
5. Arrays.....	6
6. Enums.....	7
7. Structures	8
8. Programmable Organization Unit Objects (POUs)	9
Function.....	9
Program	9
Function Block	10

1. Variables

Declaration

Example:

```
VAR
    counter: INT := 0;
END_VAR
```

Description:

1. **Name** of the variable
2. **Data type** of variable
3. **Initial value** of variable

Variables are used as placeholders. Values can be assigned to or read from variables. They can also be used in comparisons.

USAGE

Example:

```
counter := counter + 1;

anotherCounter := counter;

IF counter < 100 THEN
    ...
END_IF
```

Description:

1. Reads current value of variable **counter** adds 1 and assigns this new value to **counter**.
2. Reads current value of variable **counter** and assigns it to **anotherCounter**.
3. Reads current value of variable **counter** and checks if it's smaller than 100.

Data Types

Name	Description	Range	Memory use in Byte
BOOL	Boolean	TRUE, FALSE	1
BYTE	Unsigned fixed point (same as USINT)	0 ... 255	1
WORD	Unsigned fixed point (same as UINT)	0 ... 65535	2
DWORD	Unsigned fixed point (same as UDINT)	0 ... 4294967295	4
SINT	Signed fixed point	-128 ... 127	1
USINT	Unsigned fixed point	0 ... 255	1
INT	Signed fixed point	-32768 ... 32767	2
UINT	Unsigned fixed point	0 ... 65535	2
DINT	Signed fixed point	-2147483648 ... 2147483647	4
UDINT	Unsigned fixed point	0 ... 4294967295	4
REAL	Decimal point	-3.402823E38 ... 3.402823E38	4
LREAL	Decimal point	-1.79769313486231E308 ... 1.79769313486232E308	8

Further data types which are not listed here: STRING, TIME, TIME_OF_DAY, DATE, DATE_AND_TIME

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/core_languagemodelmanager_decl_home.htm
http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/variables_initialization.htm
http://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_plc_data_types_overview.htm

2. Constants

Declaration

Example:

```
VAR_GLOBAL CONSTANT
    PI : REAL := 3.15159;
END_VAR
```

Description:

1. **Name** of the constant
2. **Data type** of constant
3. **Value** of constant

Constants can be used for values that don't change. They make code much more readable and less error prone, because they can be changed in a single location and don't have to be tracked all over the code.

USAGE

Example:

```
temp := SIN(2 * PI * x);
```

Description:

Looks up the value of the **constant** and uses it for the calculation.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/Constants_CONSTANT_Typed_Literals.htm

3. Decisions

IF

Example:

```
IF i = 100 THEN
    ...
END_IF
```

Description:

Executes the **instructions** if the **boolean condition** is true.

IF/ELSE

Example:

```
IF i = 100 THEN
    ...
ELSE
    ...
END_IF
```

Description:

Executes the **instructions** if the **boolean condition** is true, else the **other instructions** are executed.

ELSIF

Example:

```
IF i = 100 THEN
    ...
ELSIF i = 200 THEN
    ...
ELSE
    ...
END_IF
```

Description:

Executes the **instructions** if the **boolean condition** is true. Executes the next **instructions** if the **following boolean condition** is true. Executes the **last instructions** if none of the boolean conditions above are true. Multiple ELSIF instructions can be used between the first IF and the last ELSE.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/ifinstruction.htm

CASE

Example:

```
CASE foodAtHome OF
    Spagetti:
        ...
    Pizza:
        ...
    Bread, Butter, Marmelade:
        ...
ELSE:
    ...
END_CASE
```

Description:

Executes the **instructions** of the **label** that matches the **condition variable**. If no label matches, the **instructions** after ELSE are executed.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/caseinstruction.htm

4. Loops

FOR

Example:

```
i : INT;  
  
FOR i := 1 TO 10 BY 1 DO  
  ...  
END_FOR
```

Description:

1. First initializes counter
2. Checks if counter has reached the stop limit
3. Executes the instructions in the loop
4. Increases the counter a certain amount
5. Goes to 2.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/FORloop.htm

WHILE

Example:

```
WHILE time < 10 DO  
  ...  
END_WHILE
```

Description:

1. Checks boolean condition
2. If the condition is TRUE:
 - o Executes the instructions in the loop
 - o Goes to 1.
3. Leaves the loop if condition is FALSE

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/whileloop.htm

REPEAT

Example:

```
REPEAT  
  ...  
UNTIL time < 10  
END_REPEAT
```

Description:

1. Executes the instructions in the loop
2. Checks boolean condition
3. As long as the condition is TRUE:
 - o Goes to 1.
4. Leaves the loop if condition is FALSE

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/repeatloop.htm

EXIT

Example:

```
WHILE TRUE DO
    IF i = 100 THEN
        EXIT;
    END_IF
END_WHILE
```

Description:

Exits the innermost loop immediately.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/exitinstruction.htm

CONTINUE

Example:

```
WHILE TRUE DO
    IF i = 100 THEN
        CONTINUE;
    END_IF
    ...
END_WHILE
```

Description:

Ignores the following instructions in the loop and starts with the next loop cycle.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/continue_instruction.htm

5. Arrays

Declaration

Example:

```
myArray: ARRAY [1..10] OF INT;
```

Description:

4. Variable name of the array
5. Lowest and highest index.
6. Data type of array

Arrays can store a number of elements of the same data type. In the example above the array consists of 10 INT elements. One to three dimensional arrays are supported. It's a good practise to use constants to specify the indices. These constants can be used when accessing the elements later on.

Accessing elements

Example:

```
myArray[1] := 345;
temp := myArray[2];
```

Description:

1. Variable name of the array
2. Index to read from or write to.

Each element of an array can be accessed by it's index. Trying to access a nonexistent index results in an error at runtime.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/TcPlcControl/HTML/TcPlcCtrl_ARRAY.htm

6. Enums


Declaration

Example:

```
TYPE StopLight :
(
    StopLightRed,
    StopLightYellow,
    StopLightGreen
);
END_TYPE
```

Description:

1. Enum **name**
2. **Possible values** of enum

Enums allow to create a datatype that can only have certain values. If decisions are using Enums it's more obvious what the decision is about. To avoid naming conflicts the name of the Enum should precede  all it's defined values.

Using enums

Example:

```
aLight : StopLight;

aLight := StopLightRed;

IF aLight = StopLightGreen THEN
    ...
END_IF
```

Description:

1. **Name** of variable that uses an Enum as data type
2. Assigning of or comparing to an **Enum value**

Enums can be used as data types for regular variables or also with ARRAYS.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_enum.htm

7. Structures

Declaration

Example:

```
TYPE Car:
STRUCT
    MaxSpeed: REAL;
    Price: REAL;
    NumberOfSeats: INT;
END_STRUCT
END_TYPE
```

Description:

3. Structure name
4. Variable names and data types

Structures allow to compose new data types. This makes it easier to handle data that belongs together. Structures can consist of regular data types (like INT, REAL, etc.), ARRAYS or even other structures.

Using structures

Example:

```
toyotaCorolla: Car;

toyotaCorolla.MaxSpeed := 160;
toyotaCorolla.Price := 21555.5;
toyotaCorolla.NumberOfSeats := 5;
```

Description:

3. Name of variable that uses a structure as data type
4. Accessing the elements of the variable.

Structures can be used as data types for regular variables or also with ARRAYS.

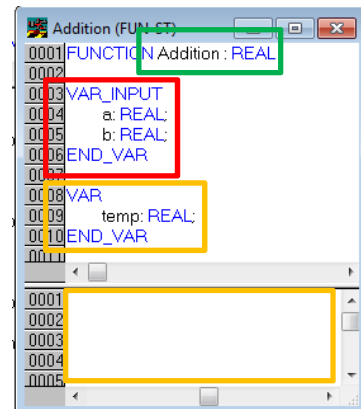
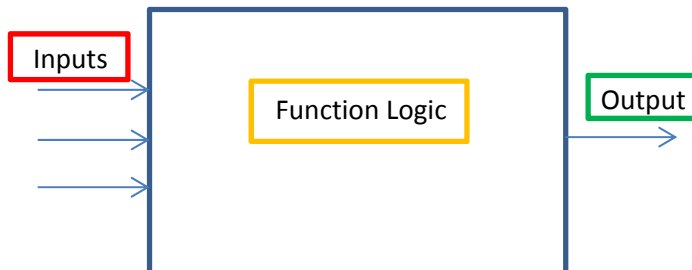
Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_struct.htm

8. Programmable Organization Unit Objects (POUs)

Function

Schema:



Description:

A function ...

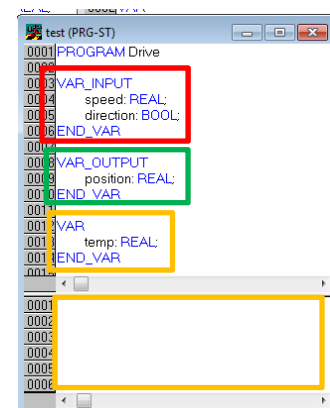
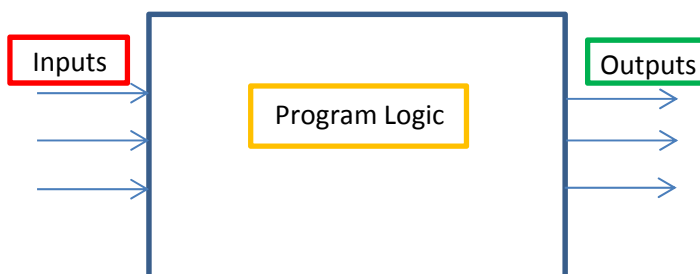
- ... returns exactly one value after it's execution.
- ... does not remember any values from the last time it was executed.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/pou_function.htm

Program

Schema:



Description:

A program ...

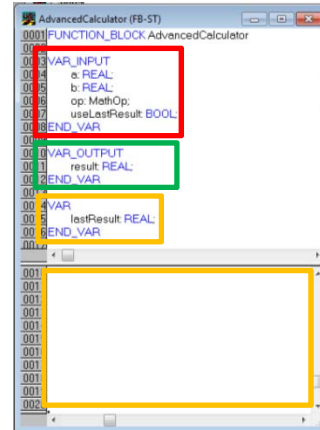
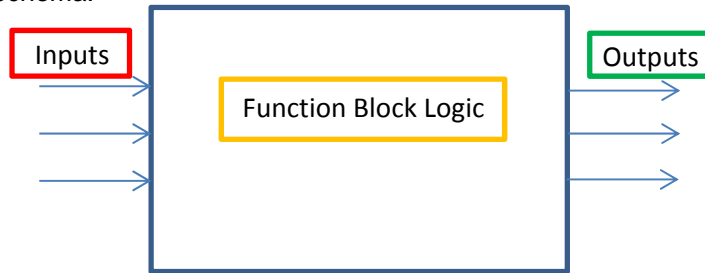
- ... returns one or several values during operation.
- ... remembers all values from the last time the program was run.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/pou_program.htm

Function Block

Schema:



Description:

A function block ...

- ... returns one or several values during operation.
- ... remembers all values from the last time the function block was run.
- ... needs to be instantiated before it can be executed (similar to a class in Java).
- ... can be instantiated multiple times.

Beckhoff:

http://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/html/POU_Function_Block.htm