Structured Text

TUTORIAL IEC 61131-3

control byte

Program Flow Control Instructions

`IF <conditions> THEN`
`  <code>`
`ELSIF <conditions> THEN`
`  <code>`
`ELSE`
`  <code>`
`END_IF`
	Provides one or more options and selects one (or none) of its statement components for execution. ELEIF and ELSE are optional.

`CASE <var> OF`
`  <int>:`
`  <code>`
`  <int>:`
`  <code>`
`  ELSE`
`  <code>`
`END_CASE`
	Select one of several alternative program sections. ELSE are optional.

`FOR <var>:=<int> TO <int> BY <step> DO`
`  <code>`
`END_FOR`
	Repeat a sequence of statements as long as a control variable is within the specified range of values.

`REPEAT`
`  <code>`
`UNTIL <conditions>`
`END_REPEAT`
	Repeat a sequence of statements until condition(S) is true. Note minimum one execution.

`WHILE <conditions> DO`
`  <code>`
`END_WHILE`
	Repeat a sequence of statements as long as condition(S) is true.

`EXIT`	Terminates the FOR, WHILE or REPEAT loop in which it resides without regard to any condition.
`RETURN`	Terminates Program, Function block call.

Prefix structure

PP_ppName	PP = Data structure / IO prefix pp = Variable prefix Name = Name of object

Example	
ioi_xStartMachine	Digital input
FB_MyMachine	Data structure of function block
fbMyMachine	Instance of function block
xMachineState	Local Boolean variable

Keyboard shortcuts

F2	Input Assistant
Shift + F2	Auto Declare
F5	Start/Run
F8	Step into
F9	Toggle breakpoint
F11	Build / Compile
Ctrl + SPACE	Auto Complete
Ctrl + F7	Write online change

Prefix

Data Structure Object Prefix	
SR_	Program
FB_	Function block
FC_	Function
A_	Action
ST_	Structure
ET_	Enum
IF_	Interface

I/O prefix	
iq_	IN-OUT variable
i_	INPUT variable
q_	OUTPUT variable
p_	Persistent variable
r_	Retain variable
rp_	Persistent and retain variable
g_	Global variable
*)	Constant variable
ioi_	IO input
ioq_	IO output
cani_	CAN input
canq_	CAN output
mbi_	Modbus input
mbq_	Modbus output

Variable prefix	
x	BOOL
by	BYTE
si/usi	SINT - Short integer
w	WORD
i/ui	INT - Integer
dw	DWORD - Double word
di/udi	DINT - Double integer
li/uli	LINT - Long integer
r	REAL
lr	LREAL - Long real
s	STRING - text
tim	TIME
tod	TOD - Time of day
date	DATE
dt	DT - Date and time
ws	WSTRING - Unicode

Instance prefix	
fb	Function block
st	Struct
if	Interface

*) No Prefix – Use capital letter.
Ex: wCOLOR or g_wCOLOR

Data types

BOOL	0 – 1 (FALSE - TRUE)
BYTE	0 - 255
SINT	- 128 – 127
WORD	0 – 65535
INT	- 32768 – 32767
UINT	0 – 65535
DWORD	0 – 4294967295
UDINT	0 – 4294967295
DINT	- 2147483648 – 2147483647
LINT	- 2^63 – 2^63-1
ULINT	0 – 2^64-1
REAL	1.401e-45...3.403e+38
LREAL	2.2250738585072014e-308...1,7976931348623158e+308
STRING	'String'
WSTRING	"Long String"
TIME	T#2s, t#100ms
TOD	TOD#16:00:00
DATE	D#2023-10-18
DT	DT#2023-10-17-09:00:00

Conversion

<DataType>_TO_<DataType>

Example
BOOL_TO_INT
WORD_TO_DINT
BYTE_TO_REAL
REAL_TO_LREAL
TIME_TO_DINT

Numeric

ABS	Absolute Value
SQR	Square Root
LN	Natural logarithm
LOG	Decimal logarithm
EXP	Exponential function
SIN	Sine
COS	Cosine
TAN	Tangent
ASIN	Reciprocal of sine
ACOS	Inverse of cosine
ATAN	Inverse of tangent
EXPT	Exponentiation

Counters

CTU	Counting Up
CTD	Counting Down
CTUD	Counting up and down

Bitwise Shift Functions

ROL	Rotates counterclockwise by N
ROR	Rotates clockwise by N
SHL	Moves left by N
SHR	Shifts right by N

Operations

Arithmetic	
:=	Assignment
*	Multiplication
/	Division
MOD	Modolo division
+	Addition
-	Subtraction
Comparison	
<	Less than
>	Greather than
<=	Less than or equal to
>=	Greather than or equal to
=	Equal to
<>	Not equal to
Logical	
NOT	Negation
AND	And
XOR	Exclusive or
OR	Or
Parentheses and indexing	
()	Parentheses
[]	Array index

String processing

LEN	Length of string
CONTACT	Connect two strings
LEFT	Returns N characters from the left
RIGHT	Returns N characters from the right
MID	Returns N1 characters from N2
INSERT	Insert string at N
DELETE	Delete a substring
REPLACE	Replace a substring
FIND	Find a substring in a string

Selecting a Value

SEL	Returns a value based on a condition
MAX	Returns the maximum value
MIN	Returns the minimum value
LIMIT	Returns a value based on the limit
MUX	Returns a value based on an index

Timers

TP	Pulse timer
TON	Timer with delayed switch-on
TOF	Timer with delayed shutdown

Signal Edge Detection

R_TRIG	Rising signal
F_TRIG	Falling Signal

Conversion

TRUNC	Shortens the number to DINT format
TRUNC_INT	Shortens the number to INT format

Sample programs in ST

Lamp control

```
PROGRAM LedControl
VAR
    xButton1, xButton2, xEmergency: BOOL;    // Definition of input logical variables
    xLed: BOOL;    // Output variable for the LED
END_VAR

xLed := (xButton1 AND xButton2) OR xEmergency;    // Logic for controlling the LED

END_PROGRAM
```

Pump control

```
PROGRAM PumpControl
VAR
    xButton1: BOOL;    // Logical variable for the button
    xPump1: BOOL;    // Logical variable for the pump
END_VAR

IF xButton1 THEN
    xPump1 := TRUE;    // Turning on the pump when the button is active
ELSE
    xPump1 := FALSE;    // Turning off the pump when the button is not active
END_IF

END_PROGRAM
```

Parcel Weighing

```
PROGRAM PackageSelection
VAR
    iPackageWeight: INT;    // Weight of the package
    iWeightCategory: INT;    // Weight category of the package
END_VAR

CASE iPackageWeight OF
    0..999:
        iWeightCategory := 1;    // Light packages
    1000..1999:
        iWeightCategory := 2;    // Medium packages
    2000..2999:
        iWeightCategory := 3;    // Heavy packages
    ELSE
        iWeightCategory := 4;    // Very heavy packages
END_CASE

END_PROGRAM
```

Printing the highest temperature from the array

```
PROGRAM MaxTemperature
VAR
    aTemperatures: ARRAY[1..10] OF REAL;   // An array of temperatures
    rMaxTemp: REAL;   // Variable for the highest temperature
    i: INT;   // Loop counter
END_VAR

rMaxTemp := aTemperatures[1];   // Initialize with the first temperature

i := 1;

WHILE i <= 10 DO
    IF aTemperatures[i] > rMaxTemp THEN
        rMaxTemp := aTemperatures[i];   // Update the max temperature
    END_IF
    i := i + 1;
END_WHILE

END_PROGRAM
```