

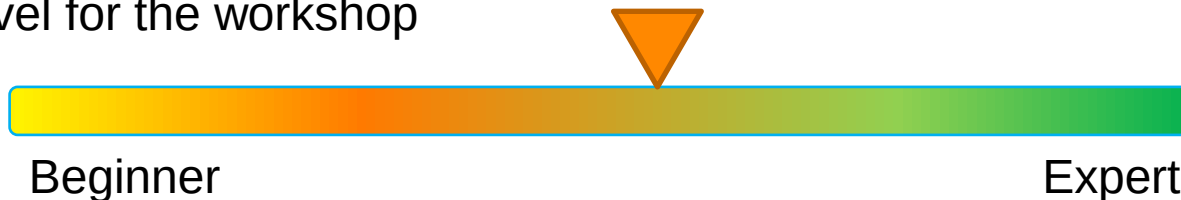


Structured Text (ST)



Time	Topic	Presenter
8:00	Introduction Structured Text	Kohlbacher
9:00	Practice Structured Text	Kohlbacher
10:30	Introduction Sequential Function Chart	Kohlbacher
Afternoon	Practice Sequential Function Chart	Kohlbacher

Aimed level for the workshop



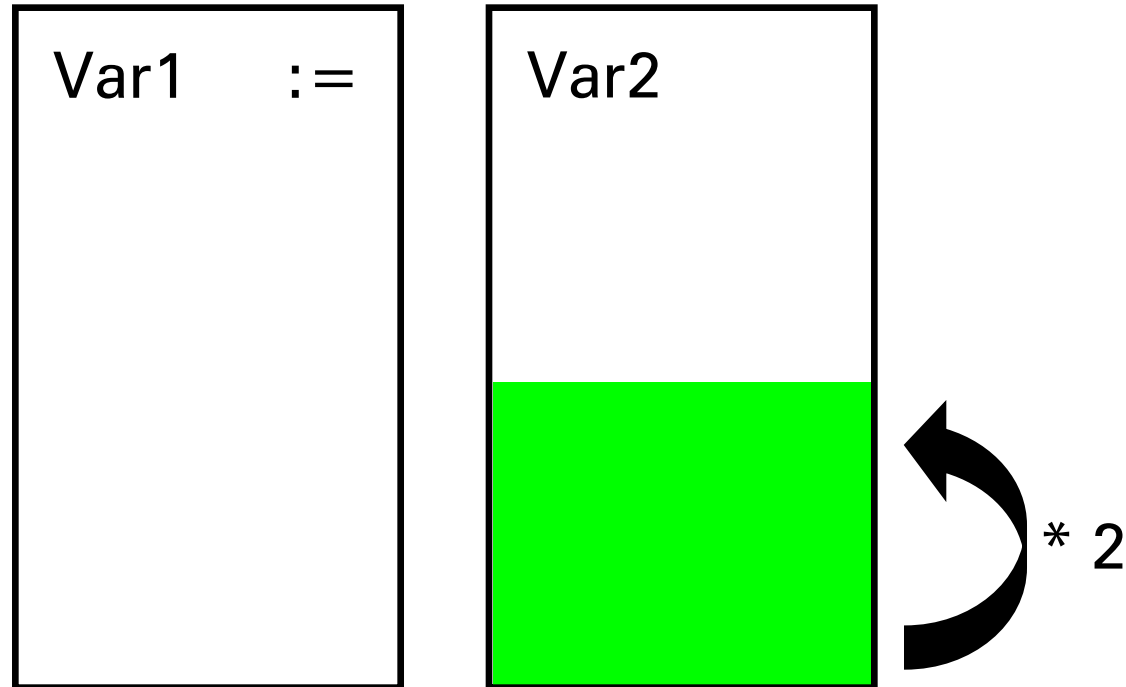


- High-level, text-based language
- Easy-to-use standard constructs
- Conforms to the IEC 61131-3 standard
- Command groups for
 - Logic
 - Decisions
 - State machines
 - Loops
- Possibility to call function/function blocks





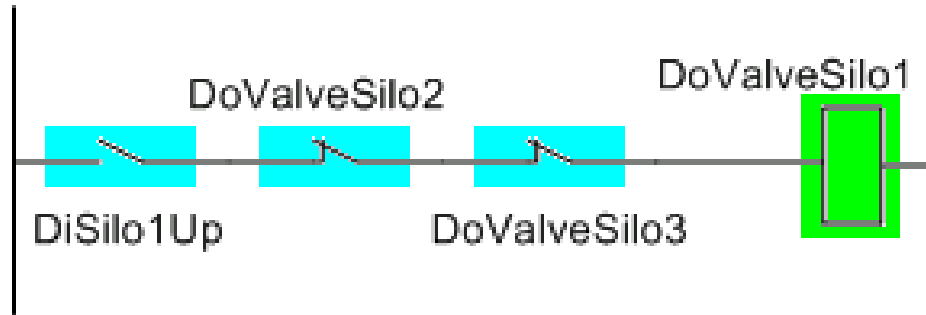
```
Var1 := Var2 * 2; (* Var1 <-- (Var2 * 2) *)
```



- The result will be assigned to the left side



Command groups – Boolean operations



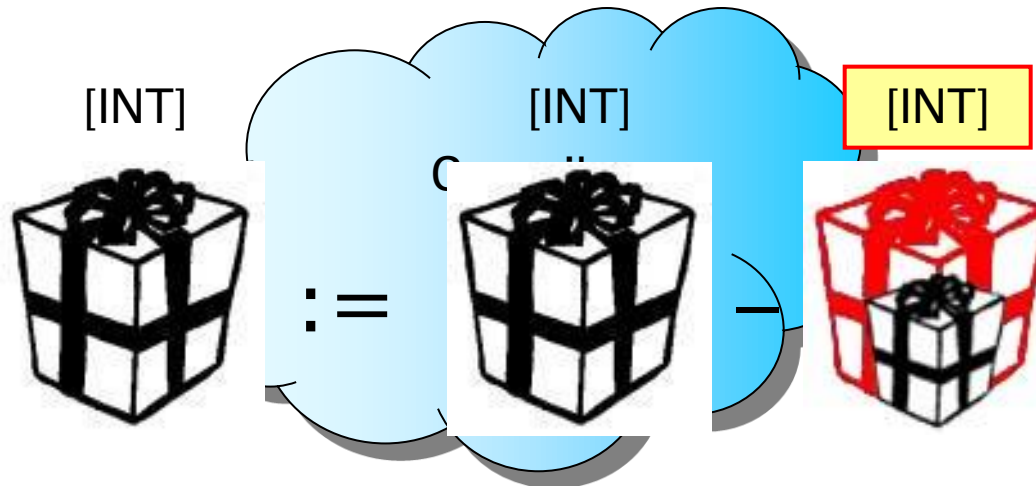
```
DoValveSilo1 := (DiSilo1Up AND (NOT DoValveSilo2) AND (NOT DoValveSilo3));
```





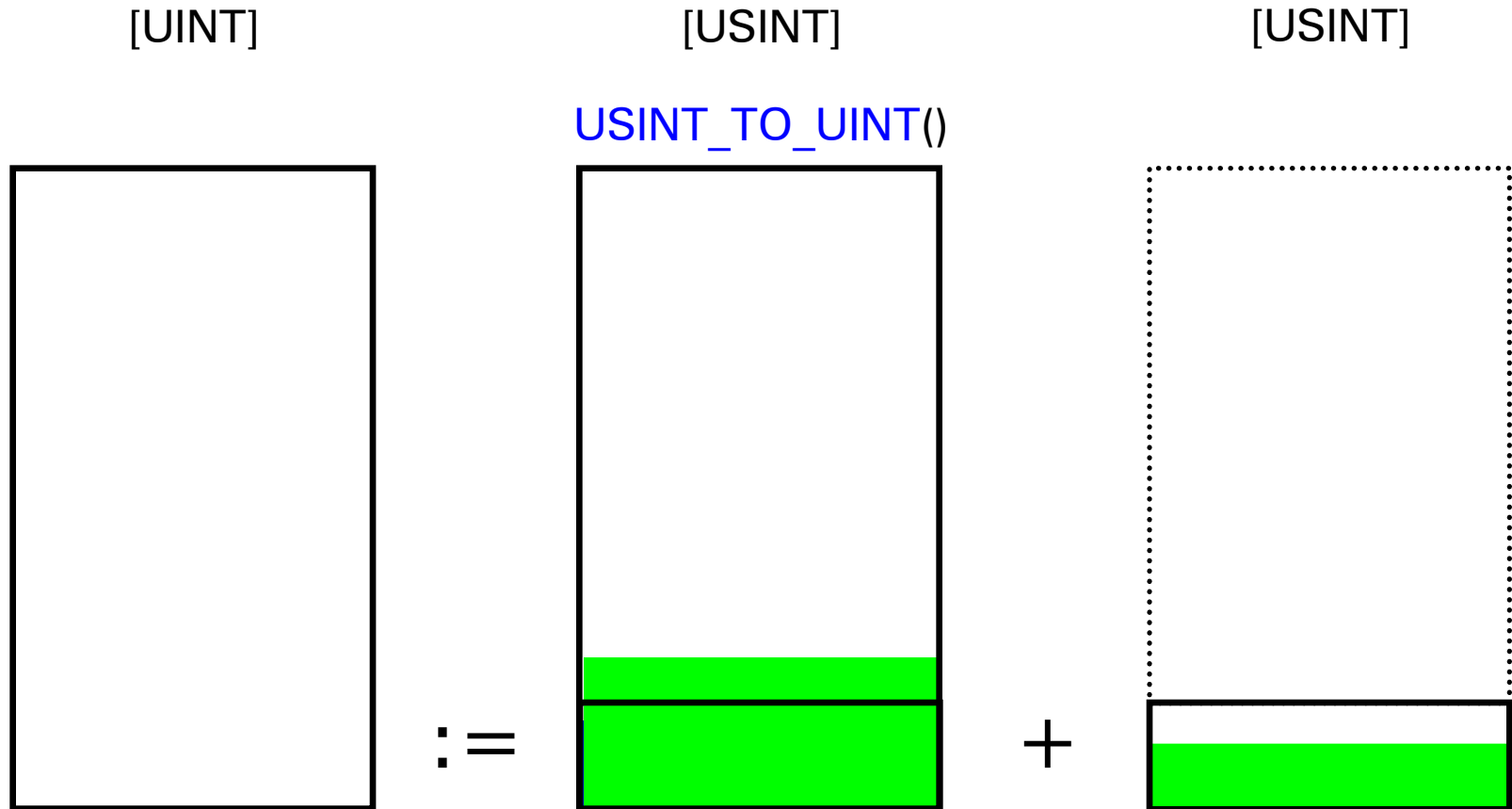
Implicit data type conversion

- Automatically performed by the compiler





Explicit data type conversion





IF – ELSIF – ELSE – statement

Examples

```
IF Var1 < Var2 THEN
```

```
    Output:= 1;
```

```
ELSIF Var1 > Var2 THEN
```

```
    Output:= 2;
```

```
ELSIF Var2 > Var3 THEN
```

```
    Output:= 3;
```

```
ELSE
```

```
    Output:= 0;
```

```
END_IF
```

■ Case 1: Var1 = 1
 Var2 = 2
 Var3 = 0

■ Case 2: Var1 = 3
 Var2 = 2
 Var3 = 0

■ Case 3: Var1 = 2
 Var2 = 2
 Var3 = 0

■ Case 4: Var1 = 5
 Var2 = 5
 Var3 = 5





CASE – statement

Examples

```
CASE Step OF
```

```
1:
```

```
    Var1:= 1;
```

```
    Var2:= 3;
```

```
2, 4, 12:
```

```
    Var1:= Step;
```

```
    Var2:= 2;
```

```
6..10:
```

```
    Var1:= 4;
```

```
    Var2:= 1;
```

```
ELSE
```

```
    Var1:= 0;
```

```
    Var2:= 0;
```

```
END_CASE
```

■ Case 1: Step = 1

■ Case 2: Step = 4

■ Case 3: Step = 9

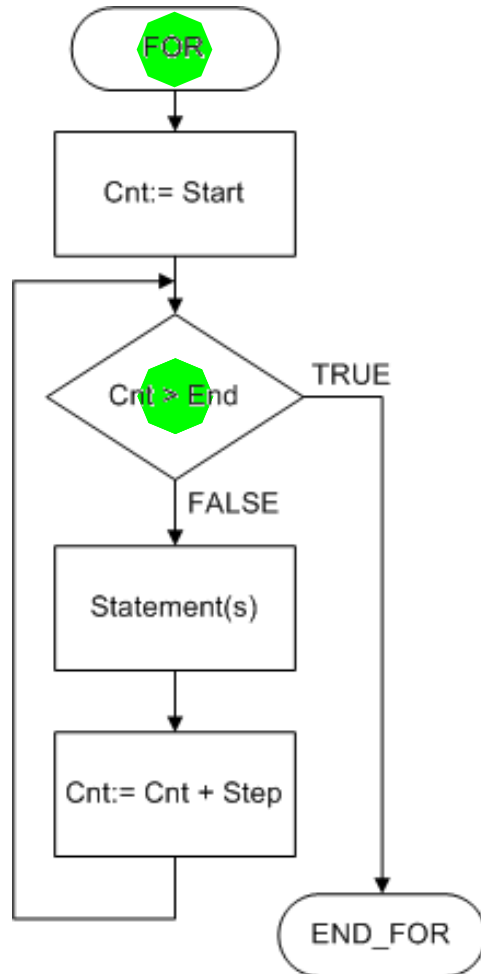
■ Case 4: Step = 5





FOR – loop

Example



```
FOR Cnt:= 3 TO 5 BY 1 DO
```

```
    Var1:= Var1 + Cnt;
```

```
END_FOR
```

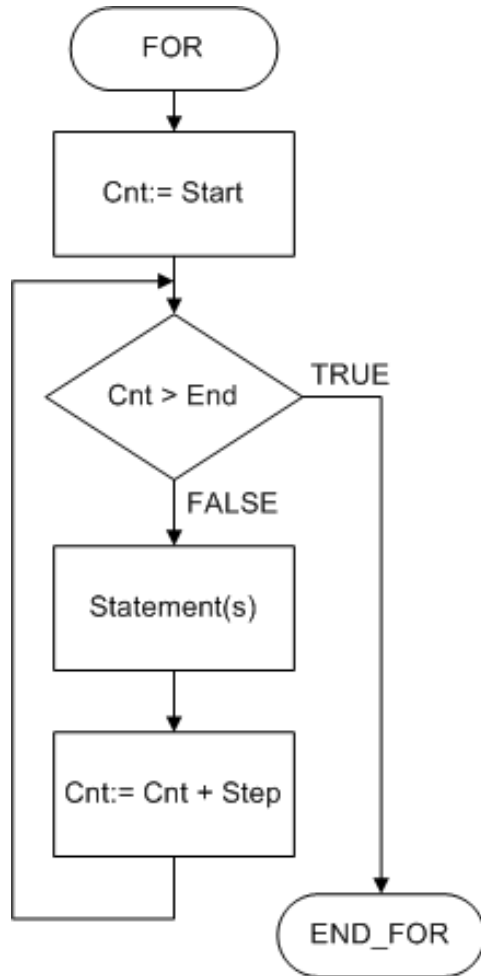
pass		1	2	3	4
Cnt	X	3	4	5	6
Var1	0	3	7	12	12



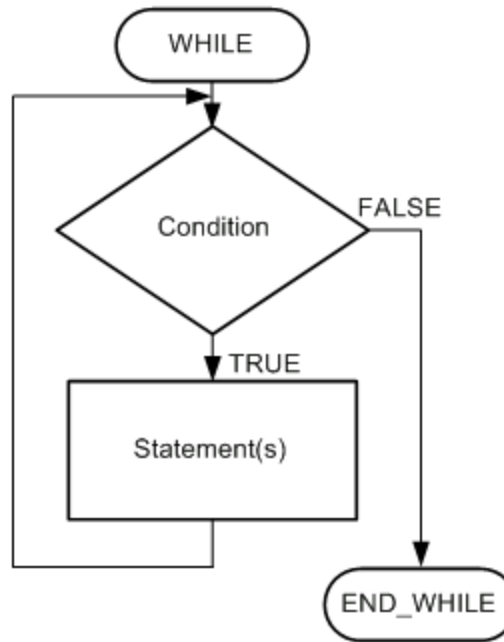
Command groups – different types of loops



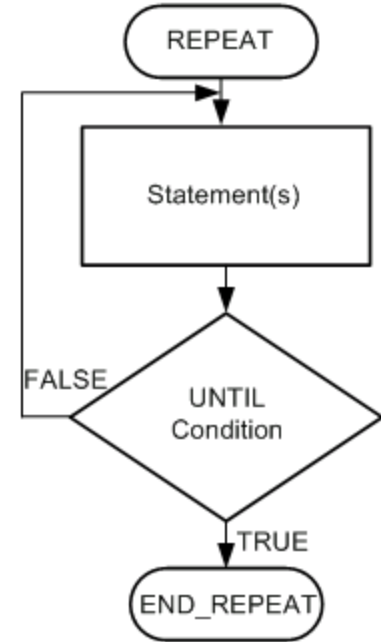
■ FOR – loop



■ WHILE – loop



■ REPEAT – loop

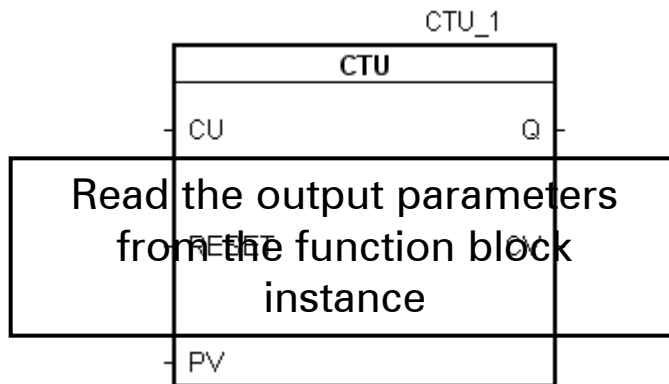


Calling function blocks I



Example: function block CTU (upward counter)

Assign the input parameters to the function block instance



(* Set inputs *)

```
CTU_1.CU:= Trigger;  
CTU_1.RESET:= Reset;  
CTU_1.PV:= 10;
```

(* Call function block *)

```
CTU_1();
```

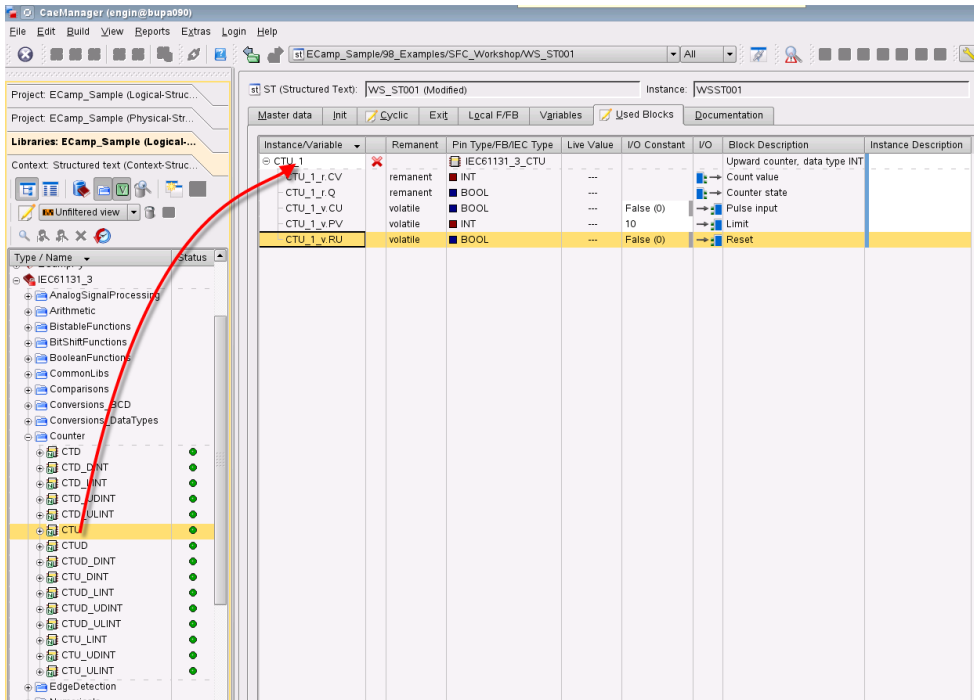
(* Read outputs *)

```
Counter:= CTU_1.CV;  
Finished:= CTU_1.Q;
```





Example: function block CTU (counter up) from IEC61131_3 Library



(* set the inputs *)

```
CTU_1.CU:= Trigger;
```

```
CTU_1.RESET:= Reset;
```

```
CTU_1.PV:= 10;
```

(* call the function block *)

```
IEC61131_3_CTU(CTU_1.v  
,CTU_1.r);
```

(* read the outputs *)

```
Counter:= CTU_1.CV;
```

```
Finished:= CTU_1.Q;
```



Live presentation with APROL ST-Editor



YOUR GLOBAL PARTNER FOR AUTOMATION EXCELLENCE