LamaPLC: Simatic S7 SCL commands: Conversions

ROUND: Round numerical value

Round numerical value.

_FC_ Result := **ROUND** (Value);

Value: function input (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers)

```
1
2  #result_int := ROUND(3.5);
3  #result_real := ROUND(3.5);
4  #result_lreal := ROUND(3.5);
5
6  #result_int := ROUND(-3.5);
7  #result_real := ROUND(-3.5);
8  #result_lreal := ROUND(-3.5);
9
10
```

#result_int	4
#result_real	4.0
#result_lreal	4.0
#result_int	-4
#result_real	-4.0
#result_lreal	-4.0

A yellow underline in the code indicates that the result of the function is not completely accurate for INT types. For REAL, LREAL type, precision is complete.

>> Back to LamaPLC main menu (SCL commands)

ROUND_x

On the TIA portal, you can specify the type of variable used for the function by entering ROUND_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

CEIL: Generate next higher integer from floating-point number

Generate next higher integer from floating-point number.

_FC_ Result := **CEIL** (Value);

Value: function input (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers)

```
1
2  #result_int := CEIL(3.5);
3  #result_real := CEIL(3.5);
4  #result_lreal := CEIL(3.5);
5
6  #result_int := CEIL(-3.5);
7  #result_real := CEIL(-3.5);
8  #result_lreal := CEIL(-3.5);
9
10 #result_real := CEIL_REAL(-3.5);
11
```

#result_int	4
#result_real	4.0
#result_lreal	4.0
#result_int	-3
#result_real	-3.0
#result_lreal	-3.0
#result_real	-3.0

A yellow underline in the code indicates that the result of the function is not completely accurate for INT types. For REAL, LREAL type, precision is complete.

>> Back to LamaPLC main menu (SCL commands)

CEIL_x

On the TIA portal, you can specify the type of variable used for the function by entering CEIL_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

FLOOR: Generate next lower integer from floating-point number

Generate next lower integer from floating-point number.

_FC_ Result := **FLOOR** (Value);

Value: function input (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers)

```
1
2  #result_int := FLOOR(3.5);
3  #result_real := FLOOR(3.5);
4  #result_lreal := FLOOR(3.5);
5
6  #result_int := FLOOR(-3.5);
7  #result_real := FLOOR(-3.5);
8  #result_lreal := FLOOR(-3.5);
9
10 #result_real := FLOOR_REAL(-3.5);
11
```

#result_int	3
#result_real	3.0
#result_lreal	3.0
#result_int	-4
#result_real	-4.0
#result_lreal	-4.0
#result_real	-4.0

A yellow underline in the code indicates that the result of the function is not completely accurate for INT types. For REAL, LREAL type, precision is complete.

>> Back to LamaPLC main menu (SCL commands)

FLOOR_x

On the TIA portal, you can specify the type of variable used for the function by entering FLOOR_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

TRUNC: Truncate numerical value

Truncate numerical value.

_FC_ Result := **TRUNC** (Value);

Value: function input (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers), Default: DINT

```
2  #result_int := TRUNC(3.5);
3  // default DINT:
4  #result_dint := TRUNC(3.5);
5
6  #result_real := TRUNC(3.5);
7  #result_lreal := TRUNC(3.5);
8
9  #result_int := TRUNC(-3.5);
10 #result_real := TRUNC(-3.5);
11 #result_lreal := TRUNC(-3.5);
12
13 #result_real := TRUNC_REAL(-3.5);
14
```

#result_int	3
#result_dint	3
#result_real	3.0
#result_lreal	3.0
#result_int	-3
#result_real	-3.0
#result_lreal	-3.0
#result_real	-3.0

A yellow underline in the code indicates that the result of the function is not completely accurate for INT types. For REAL, LREAL type, precision is complete.

>> Back to LamaPLC main menu (SCL commands)

TRUNC_x

On the TIA portal, you can specify the type of variable used for the function by entering TRUNC_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

SCALE_X: Scale

Use the “Scale” instruction to scale a floating-point number by mapping it to a specific value range. You specify the value range with the MIN and MAX parameters. The result of the scaling is an integer. The inverse of the scale function is the norm (NORM_X) function.

_FC_ Result := **SCALE_X** (MIN := minimum value, VALUE:= act. value, MAX := maximal value);

MIN: function input, low limit of range (integers,floating-point numbers)
MAX: function input, high limit of range (integers,floating-point numbers)
VALUE: function input, value to scale (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers), Default: INT

```
1
2  #minVal := 0.0;
3  #maxVal := 400.0;
4  #actVal := 0.5;
5
6  #result := SCALE_X(MIN := #minVal, VALUE := #actVal, MAX := #maxVal);
7
8
```

#minVal	0.0
#maxVal	400.0
#actVal	0.5
#result	200.0

>> Back to LamaPLC main menu (SCL commands)

SCALE_X_x

On the TIA portal, you can specify the type of variable used for the function by entering SCALE_X_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

NORM_X: Normalize

You can use the instruction “Normalize” to normalize the value of the tag at the VALUE input by mapping it to a linear scale. You can use the MIN and MAX parameters to define the limits of a value range that is applied to the scale. The result at the OUT output is calculated and stored as a floating-point number depending on the location of the value to be normalized within this value range. If the value to be normalized equals the value at input MIN, the instruction returns the result “0.0”. If the value to be normalized equals the value at input MAX, the instruction returns the result “1.0”. The inverse of the scale function is the scale (SCALE_X) function.

_FC_ Result := **NORM_X** (MIN := minimum range value, VALUE:= value to be normalized, MAX := maximal range value);

MIN: function input, low limit of range (integers,floating-point numbers)
MAX: function input, high limit of range (integers,floating-point numbers)
VALUE: function input, value to be normalized (floating-point numbers)
Result: the return value of the function (integers, floating-point numbers), Default: REAL

```
1
2  #minVal := 0.0;
3  #maxVal := 400.0;
4  #actVal := 0.5;
5
6  #result := SCALE_X(MIN := #minVal, VALUE := #actVal, MAX := #maxVal);
7
8  #result2 := NORM_X(MIN := #minVal, VALUE := #result, MAX := #maxVal);
9
```

#minVal	0.0
#maxVal	400.0
#actVal	0.5
#result	200.0
#result2	0.5

>> Back to LamaPLC main menu (SCL commands)

NORM_X_x

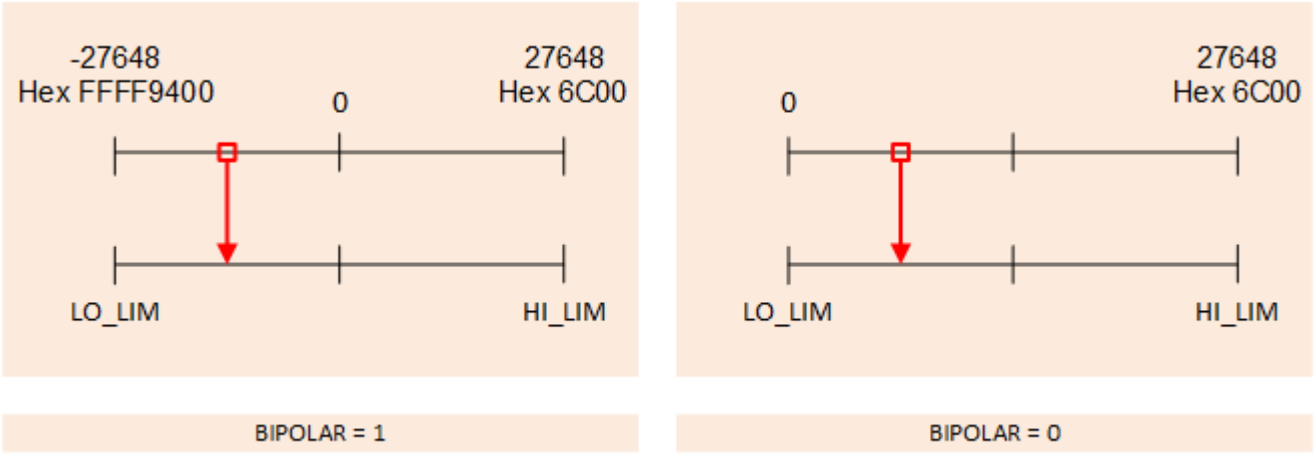
On the TIA portal, you can specify the type of variable used for the function by entering NORM_X_:INT, DINT, SINT, LINT, REAL, LREAL.

>> Back to LamaPLC main menu (SCL commands)

SCALE: Scale

Use the “Scale” instruction to convert the integer at the IN parameter into a floating-point number that can be scaled in physical units between a low limit value and a high limit value. You can use the LO_LIM and HI_LIM parameters to specify the low limit and high limit of the value range to which the input value is scaled. The result of the instruction is output at the OUT parameter.

The “Scale” instruction works with the following logic:



SCALE function BIPOLAR parameter, source: https://www.ob121.com/doku.php?id=de:s7:scl_reference_conversions

_FC_ RET_VAL := **SCALE** (IN := value to be scaled, HI_LIM:= maximal range value, LO_LIM := minimal range value, BIPOLAR := bipolar/unipolar, OUT ⇒ result);

IN: function input, value to be scaled (INT)
HI_LIM: function input, high limit of range (REAL)
LO_LIM: function input, low limit of range (REAL)
BIPOLAR: create bipolar/unipolar field (BOOL)
OUT: function output, result (REAL)
RET_VAL: functionstate (error / ok) 0: no error (WORD

```
1
2  #minVal := 0.0;
3  #maxVal := 100.0;
4  #actVal := 44;
5
6 #retval := SCALE(
7                   IN := #actVal,
8                   HI_LIM := #maxVal,
9                   LO_LIM := #minVal,
10                  BIPOLAR := FALSE,
11                  OUT => #result);
12
13#retval := SCALE(
14                  IN := #actVal,
15                  HI_LIM := #maxVal,
16                  LO_LIM := #minVal,
17                  BIPOLAR := TRUE,
18                  OUT => #result);
19
```

#minVal	0.0
#maxVal	100.0
#actVal	44
#retval	16#0000
#actVal	44
#maxVal	100.0
#minVal	0.0
#result	0.1591435
#retval	16#0000
#actVal	44
#maxVal	100.0
#minVal	0.0
#result	50.07957

>> Back to LamaPLC main menu (SCL commands)

Simatic, SCL, TIA, ROUND, CEIL, FLOOR, TRUNC, SCALE X, NORM X, SCALE, conversion