

```
FUNCTION FC 10 : VOID
```

```
TITLE =
```

```
AUTHOR : MEG
```

```
VERSION : 0.1
```

```
VAR_INPUT
```

```
  Activate : BOOL ;      //0= Function set Actual equal Null-Point
```

```
  R_max : BOOL ; //1 = Actual value approach Max_Value ( Go to Maximum )
```

```
  R_min : BOOL ; //1 = Actual value approach Min_Value ( Go to Minimum )
```

```
END_VAR
```

```
VAR_IN_OUT
```

```
  R_OK : BOOL ; //1 = Actual value reached rated value
```

```
  Max_Value : REAL ; //Maximal Value
```

```
  Null_Point : REAL ; //Null-Point, value with which the function starts
```

```
  Min_Value : REAL ; //Minimal Value
```

```
  Actual_Value : REAL ; //Actual Value
```

```
  R_plus : REAL ; //steprange for approximation Actual_Value to Max_Value
```

```
  R_minus : REAL ; //steprange for approximation Actual_Value to Min_Value
```

```
  R_Time : DWORD ; //Time for a Step
```

```
  MEG_Time : DWORD ; //Cycle time of previous OB1 scan (milliseconds)
```

```
END_VAR
```

```
VAR_TEMP
```

```
  Temp_Rated_Value : REAL ; //temporary rated value
```

```
  Temp_Step_Range : REAL ; //temporary value for actual steprange
```

```
  Proportion : REAL ; //Proportion between R_Time and Cycle Time
```

```
END_VAR
```

```
BEGIN
```

```
NETWORK
```

```
TITLE =Set Values
```

```
  CLR ;
```

```
  = #R_OK; // Reset "Done" - Signal
```

```
NETWORK
```

```
TITLE =check Activatebit
```

```
  U #Activate; // Function should run
```

```
  SPB m001; // go to m001, check input values
```

```
// the function is'nt to run
```

```
  L #Null_Point;
```

```
  T #Actual_Value; // set Actual_Value equal Nullpoint
```

```
  BEA ; // and exit
```

```
NETWORK
```

```
TITLE =check input values
```

```
m001: L 0.000000e+000;
```

```
  L #R_plus; // check R_Plus > 0
```

```
  <R ;
```

```
  SPB m01a; // if yes goto m01a, next check
```

```
// if R_Plus < 0
```

```
  NEGR ; // Make R_Plus positiv
```

```
  T #R_plus; // save R_Plus
```

```
  L 0.000000e+000; // R_Plus == 0 ?
```

```
  <>R ;
```

```
  SPB m01a; // if not , next check
```

```
  L 1.000000e+000; // else save default value ( 1 )
```

```
  T #R_plus;
```

```
// check R_minus
```

```
m01a: L 0.000000e+000;
```

```
  L #R_minus;
```

```
  <R ; // if R_minus > 0
```

```
  SPB m01b; // goto m01b, next check
```

```
    NEGR ; // else set R_minus positiv
    T    #R_minus; // Save corrected Value
    L    0.000000e+000; // R_minus == 0 ?
    <>R   ; //
    SPB  m01b; //if not go to m01b
    L    1.000000e+000; // else save default value ( 1 )
    T    #R_minus; // for R_minus

//check time for step
m01b: L    0; // load compare value
      L    #R_Time;
      <D   ; // if R_Time > 0 goto m01c
      SPB  m01c;
      L    1; // else load default value
      T    #R_Time; // save default value

m01c: NOP  0;

NETWORK
TITLE =set StepRange

      U    #R_max;
      UN   #R_min; // if function should go to max
      SPB  m02a; // go to m02a

      UN   #R_max;
      U    #R_min; // if function should go to minimum
      SPB  m02b; // goto m02b

// if R_max = R_min the function aprouch Actual_Value to Null_Point
      L    #Null_Point; //
      T    #Temp_Rated_Value; // save Null_Point as temporary rated Value
      SPA  m02c;

// the function aprouch Actual_Value to Maximum
m02a: L    #Max_Value; //
      T    #Temp_Rated_Value; // save Maximum as temporary rated Value
      SPA  m02c;

// the function aprouch Actual_Value to Minimum
m02b: L    #Min_Value;
      T    #Temp_Rated_Value; // save Maximum as temporary rated Value

// check Actual Value < rated Value
m02c: L    #Actual_Value;
      >R   ; // if Actual_Value < rated Value
      L    #R_plus; // load R_plus for Steprange
      T    #Temp_Step_Range; //
      SPB  m003; // goto m003

// check Actual Value > rated Value
      L    #Temp_Rated_Value; // load rated value
      L    #Actual_Value; //
      <R   ; // if Actual_Value > rated Value
      L    #R_minus; // load R_minus for Steprange
      T    #Temp_Step_Range; //
      SPB  m003;

// case Actual Value == rated Value
      SET  ; //
      =    #R_OK; // set Function done

      BEA  ; // exit

NETWORK
TITLE =correct steprange

m003: L    0;
```

```
L      #MEG_Time;
<D    ;
SPB   m03a;

L      0.000000e+000;
T      #Temp_Step_Range;
SPA   m004;

// check R_Time < Cycle time of previous OB1 scan
m03a: L      #R_Time; // load time for a step
      L      #MEG_Time; // load cycle time
      <D    ; //
      SPB   m03b; // if OB1Cycle > R_Time, goto m03b, adjust R_Time,
//                                     // R_plus and R_minus

// Calculate temporary Range for Step ( Temp_Step_Range ), for
// OB cycle time <= time for Step
// =====

// Calculate Proportion between R_Time and Cycle Time
      L      #R_Time;
      DTR   ;
      L      #MEG_Time;
      DTR   ;
      /R    ;
      T      #Proportion; // save Proportion

// Calculate new value for steprange
      L      #Temp_Step_Range;
      L      #Proportion;
      /R    ;
      T      #Temp_Step_Range; // save new value

      SPA   m004;

// if OB1Cycle > R_Time multiply R_Time, R_plus, R_minus by 2
// =====

m03b: L      #R_Time;
      SLD   1;
      T      #R_Time;

      L      #R_plus;
      L      2.000000e+000;
      *R    ;
      T      #R_plus;

      L      #R_minus;
      L      2.000000e+000;
      *R    ;
      T      #R_minus;

      SPA   m003;

NETWORK
TITLE =Calculate new value for Actual_Value

m004: L      #Temp_Rated_Value;
      L      #Actual_Value;
      >R    ; // if Actual_value < rated Value
      L      #Temp_Step_Range; // load calculated StepRange
      SPB   m04a; // go to m003

// aproach actual value to minimum
// =====
```

```
    NEGR    ;
    L       #Actual_Value;
    +R      ;
    T       #Actual_Value;

    L       #Temp_Rated_Value;
    >R      ;
    BEB     ; // if Actual_Value > Temp_Rated_Value exit

    T       #Actual_Value; // else save rated Value as Actual_Value
    SET     ;
    =       #R_OK; // set "done" bit

    BEA     ; // exit

// approach actual value to maximum
// =====
m04a: L     #Actual_Value;
      +R    ;
      T     #Actual_Value;

      L     #Temp_Rated_Value;
      <R    ;
      BEB   ; // if Actual_Value < Temp_Rated_Value -> exit

      T     #Actual_Value; // else save rated Value as Actual_Value
      SET   ;
      =     #R_OK; // set "done" bit

      BEA   ; // exit

END_FUNCTION
```