



UNI-PRO

**DEVELOPMENT ENVIRONMENT FOR
PROGRAMMABLE CONTROLLERS**



HVAC LIBRARIES MANUAL

CODE 114UPROHLE10

Important notice

This Instruction Manual should be read carefully before use, and all warnings should be observed; the Manual should then be kept for future reference.

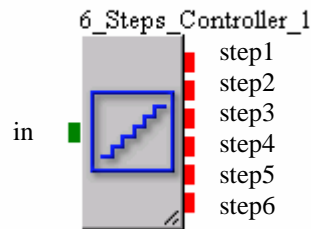
Summary

1 HVAC LIBRARIES	4
1.1 General	4
Six_Steps_Controller	4
Analog_Output_Filter	6
Analogue_Averaging	7
Dew_Point.....	8
Enthalpy	8
Enthalpy_Select	9
Holiday.....	10
HoursCounter	11
Lookup_Table	12
Pulsed_Pair.....	14
Rotation_2Bits	16
Rotation_3Bits	16
Rotation_4Bits	17
Sample and Hold.....	18
Scaling.....	18
Time_Schedule_4Modules.....	20
1.2 Regulators	24
Direct_Controller	24
Reverse_Controller	25
Two_Stages_Controller	26

1 HVAC LIBRARIES

1.1 General

Six_Steps_Controller



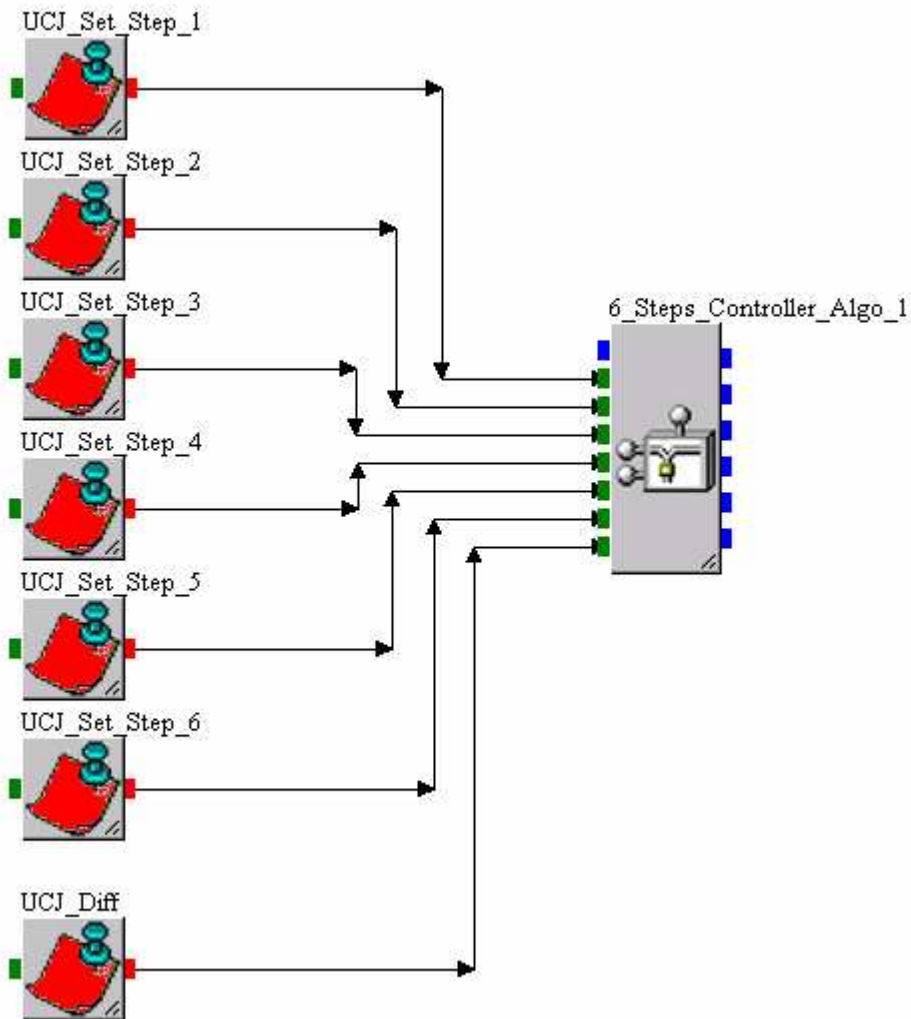
<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in</i>	CJ_SHORT	-1000.0..1000.0	Input value

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>step1</i>	CJ_BIT	0..1	Step 1 output
<i>step2</i>	CJ_BIT	0..1	Step 2 output
<i>step3</i>	CJ_BIT	0..1	Step 3 output
<i>step4</i>	CJ_BIT	0..1	Step 4 output
<i>step5</i>	CJ_BIT	0..1	Step 5 output
<i>step6</i>	CJ_BIT	0..1	Step 6 output

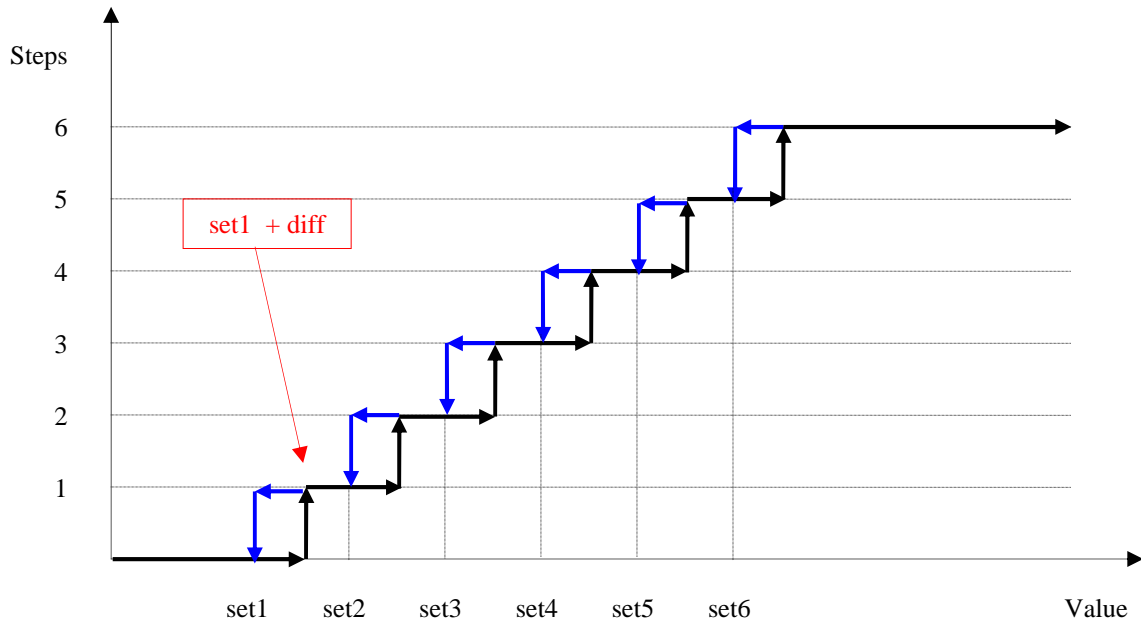
<i>Parameter</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>set_step1</i>	CJ_SHORT	-1000.0..1000.0	10.0	Step 1 set-point
<i>set_step2</i>	CJ_SHORT	-1000.0..1000.0	20.0	Step 2 set-point
<i>set_step3</i>	CJ_SHORT	-1000.0..1000.0	40.0	Step 3 set-point
<i>set_step4</i>	CJ_SHORT	-1000.0..1000.0	60.0	Step 4 set-point
<i>set_step5</i>	CJ_SHORT	-1000.0..1000.0	80.0	Step 5 set-point
<i>set_step6</i>	CJ_SHORT	-1000.0..1000.0	90.0	Step 6 set-point
<i>diff</i>	CJ_SHORT	-1000.0..1000.0	5.0	Differential. For a correct control, the differential value must be lower than the lowest steps distance.

Descrizione

The *Six_Steps_Controller* library allows to calculate the request of six outputs based on the requested value. The corresponding output to a step is activated when the input value exceeds the definitive limit by the sum between set-point and differential. The output is switched off when the value falls below the same set-point. The library is a template that contains the parameters and the control algorithm. To configure the library, double click on it and set up the parameters with appropriate values.



To better understand the functioning, see the chart below:



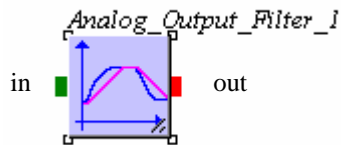
Notes

For a correct library functioning, the steps values must respect the following order:

$$set_step1 < set_step2 < set_step3 < set_step4 < set_step5 < set_step6$$

Also the differential value must be less than the lowest distance between the steps.
The *set-point* and *diff* parameters have one decimal unit accuracy.

Analog_Output_Filter



Input	Type	Limits	Description
<i>in</i>	CJ_WORD	0..100.00	Input

Output	Type	Limits	Description
<i>out</i>	CJ_WORD	0..100.00	Filtered output

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
AOF_Time	CJ_WORD	0..999.0	10.0	Time spent in tenths of a second to pass from 0.00% to 100.00%.

Description

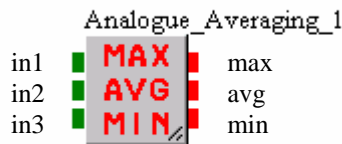
The *Analog_Output_Filter* library is usually inserted between a regulator and an analog output. This library allows to insert a filter to avoid abrupt output variations.

Inside the library there is a parameter that allows to specify the output “Adaptation time”, allows to modify the time spent by the output to pass from 0% to 100%.

If for example a 10% variation is requested, the filter will move the output to a new value in a time equal to 10% of the *AOF_Time* parameter.

If the parameter is set to zero, the output is equal to input.

Analogue_Averaging



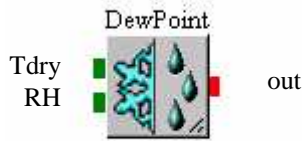
<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in1</i>	CJ_SHORT	-32768..32767	Data 1 inputs
<i>in2</i>	CJ_SHORT	-32768..32767	Data 2 inputs
<i>in3</i>	CJ_SHORT	-32768..32767	Data 3 inputs

<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>max</i>	CJ_SHORT	-32768..32767	Returns to maximum value
<i>avg</i>	CJ_SHORT	-32768..32767	Returns to medium value
<i>min</i>	CJ_SHORT	-32768..32767	Returns to minimum value

Description

The *Analog_Averaging* library allows to calculate the maximum, medium and minimum value of 3 CJ_SHORT inputs.

Dew_Point



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Tdry</i>	CJ_SHORT	-40.0..150.0	Dry bulb temperature (°C)
<i>RH</i>	CJ_SHORT	0..100	Relative humidity (0% - 100%)

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_SHORT	-40.0..150.0	Returns the condensation temperature value (°C).

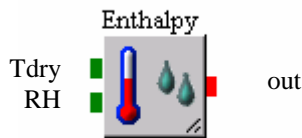
Description

The *Dew_Point* library allows to calculate the dew point (temperature when air reaches saturation conditions). In all the elements with temperature lower than calculated values the water vapor condense into water by creating dew.

Notes

The dry bulb temperature and the output temperature calculated by the library have a one decimal unit accuracy.

Enthalpy



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Tdry</i>	CJ_SHORT	-40.0..150.0	Dry bulb temperature (°C)
<i>RH</i>	CJ_SHORT	0..100	Relative Humidity (0% - 100%)

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_SHORT	-3000.0..3000.0	Air enthalpy value in KJ/Kg.

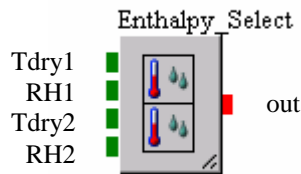
Description

The *Enthalpy* library allows to calculate the air energy content by the temperature and relative humidity that characterize it.

Notes

The dry bulb temperature and the output temperature calculated by the library, have a one decimal unit accuracy.

Enthalpy_Select



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Tdry1</i>	CJ_SHORT	-3276.8..3276.7	Dry bulb temperature 1
<i>RH1</i>	CJ_SHORT	0..100	Relative humidity 1 (0% - 100%)
<i>Tdry2</i>	CJ_SHORT	-3276.8..3276.7	Dry bulb temperature 2
<i>RH2</i>	CJ_SHORT	0..100	Relative humidity 2 (0% - 100%)

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_BIT	0..1	Comparing two enthalpies. 1 = enthalpy 1 > enthalpy 2 0 = enthalpy 2 > enthalpy 1

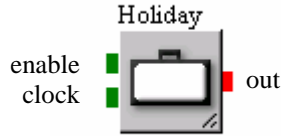
Description

The *Enthalpy_Select* library allows to compare two different enthalpies calculated according to two pairs of values of temperature and humidity.
If the enthalpy 1 is greater than enthalpy 2, the library output assumes the value 1, otherwise it assumes the value 0.

Notes

The dry bulbs temperature have a one decimal unit accuracy.

Holiday



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>enable</i>	CJ_BIT	0.1	Enabling input
<i>clock</i>	CJ_DATETIME	0..2147483647	System clock input

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_BIT	0..1	Holiday condition.

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>beginDay</i>	CJ_BYTE	1..31	1	Begin day
<i>beginMonth</i>	CJ_BYTE	1..12	1	Begin month
<i>endDay</i>	CJ_BYTE	1..31	1	End day
<i>endMonth</i>	CJ_BYTE	1..12	1	End month

Description

The *Holiday* library allows to specify a holiday period. Connecting to clock input the system clock and enabling the input is possible to specify a holiday period using the template library internal parameters. Particularly if the system clock indicates a date included in the specified period between the begin day/month and the end day/month, the library output will be on (1), otherwise will be off (0).

Notes

The library is a template.

HoursCounter



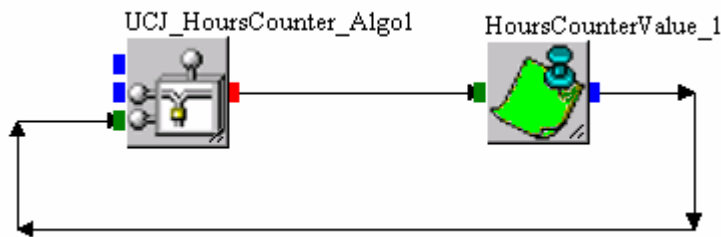
<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>cmp</i>	CJ_BIT	0..1	Compressor input
<i>reset</i>	CJ_BIT	0..1	Reset input

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_DWORD	0..4294967295	Counted hours

<i>Persistent</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>HoursCounterValue</i>	CJ_DWORD	0..4294967295	Counted hours. The values is saved in EPROM in the persistent values area.

Description

The *HoursCounter* library allows to count the working time of a device (i.e. Compressor). The device has to be connected in the *cmp* input. To reset the hour counter is necessary to enable the *reset* input. The counter value is saved in a persistent object that is stored in Eprom and keep the value even with the loss of power supply.

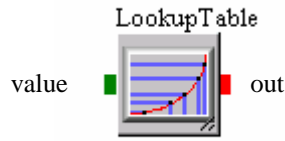


Inside the library can be visualized the *HoursCounterValue* persistent entity that for example can be connected to a EIML page or used for other purposes.

Notes

The library is a template.

Lookup_Table



<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>value</i>	CJ_SHORT	-32768..32767	Input value

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_SHORT	-32768..32767	Converted value

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>x0</i>	CJ_SHORT	-10000..10000	0	First input value
<i>x1</i>	CJ_SHORT	-10000..10000	10	Second input value
<i>x2</i>	CJ_SHORT	-10000..10000	20	Third input value
<i>x3</i>	CJ_SHORT	-10000..10000	30	Fouth input value
<i>x4</i>	CJ_SHORT	-10000..10000	40	Fifth input value
<i>x5</i>	CJ_SHORT	-10000..10000	50	Sixth input value
<i>x6</i>	CJ_SHORT	-10000..10000	60	Seventh input value
<i>x7</i>	CJ_SHORT	-10000..10000	70	Eighth input value
<i>x8</i>	CJ_SHORT	-10000..10000	80	ninth input value
<i>x9</i>	CJ_SHORT	-10000..10000	90	Tenth input value
<i>x10</i>	CJ_SHORT	-10000..10000	100	Eleventh input value
<i>y0</i>	CJ_SHORT	-10000..10000	0	First output value
<i>y1</i>	CJ_SHORT	-10000..10000	2	Second output value
<i>y2</i>	CJ_SHORT	-10000..10000	3	Third output value
<i>y3</i>	CJ_SHORT	-10000..10000	4	Fouth output value
<i>y4</i>	CJ_SHORT	-10000..10000	6	Fifth output value
<i>y5</i>	CJ_SHORT	-10000..10000	10	Sixth output value
<i>y6</i>	CJ_SHORT	-10000..10000	16	Seventh output value

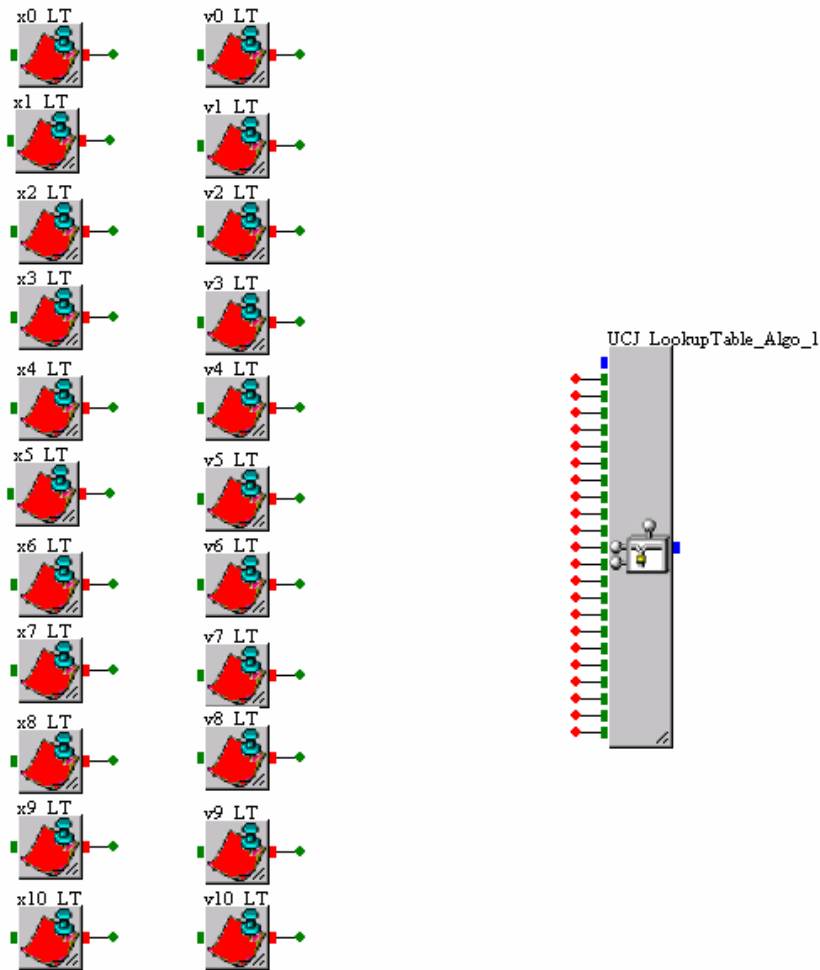
y7	CJ_SHORT	-10000..10000	25	Eighth output value
y8	CJ_SHORT	-10000..10000	40	ninth output value
y9	CJ_SHORT	-10000..10000	63	Tenth output value
y10	CJ_SHORT	-10000..10000	100	Eleventh output value

Descrizione

The *Lookup_Table* allows to convert input values through a specified curve by a values scale configured in the parameters inside the library.

For example this library is used to convert values read by a pressure sensor (4-20 mA) in corresponding pressure values.

The library is a template type and inside are 11 x-y parameters pairs that describe the curve (up to 10 line segments) for the value conversion.



Notes

The library is a template.
 In case is not necessary to define all the point pairs, the not used values must be equal to the last valid points value.

Pulsed_Pair



<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in</i>	CJ_WORD	0..100.00	Requested position (0% - 100.00%)

<i>Outputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>up</i>	CJ_BIT	0..1	Up output
<i>down</i>	C_BIT	0..1	Down output

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>PP_StrokeTime</i>	CJ_WORD	0..9999	60	Run time in seconds. Set the total time in seconds which actuator needs to pass from 0% to 100%.
<i>PP_Run_On_Time</i>	CJ_WORD	0..9999	600	Security time in seconds. Maximum time that a output can stay activated.

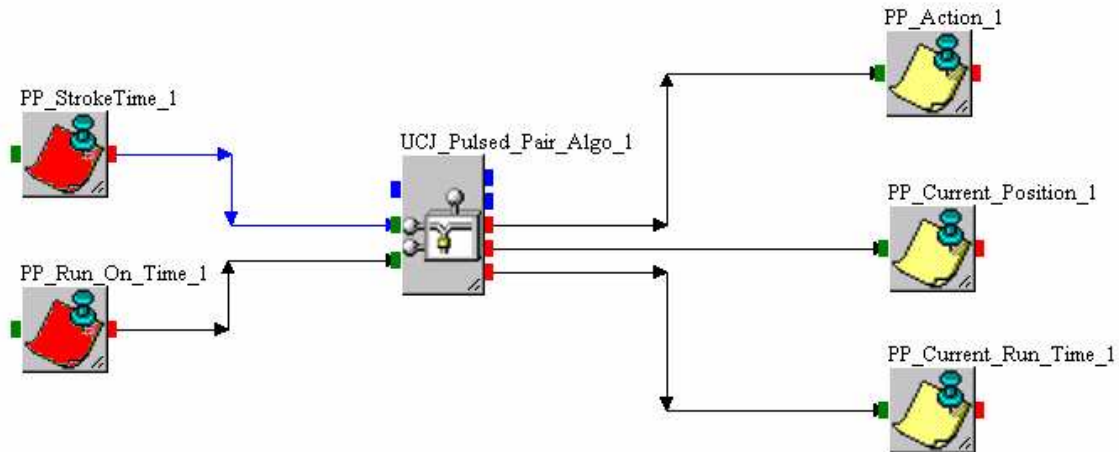
<i>Status</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>PP_Action</i>	CJ_BYTE	0..4	0	Current control status. 0 = Stopped 1 = Increase 2 = Decrease 3 = Minimum 4 = Maximum
<i>PP_Current_Position</i>	CJ_WORD	0..100.00	0	Actuator current position (0%-100%).
<i>PP_Current_Run_Time</i>	CJ_WORD	0..9999	0	Current run time in seconds. Number of seconds that the actuator worked in one direction. Is reset with the movement direction inversion.

Description

With the *Pulsed_Pair* library is possible to drive an actuator by two digital outputs.

To configure the library set the appropriate values of run time and the security time by the set up parameters inside.

Specifying the input pin wanted position value, based on the actuator current position, will be activated for necessary time the output corresponding to the direction in which the actuator must be moved to bring it to the desired position.



Within the library there are two parameters:

- *StrokeTime*: define the time in seconds (with a tenth of second accuracy) needed by the actuator to run along the entire stroke, to pass from the minimum position (0%) to the maximum position (100%).
- *Run_On_Time*: Allows to define a time in seconds while the actuator must continue working when the maximum (100%) or minimum (0%) value is reached. If the value is set 0 the outputs are switched off immediately when when the maximum or minimum value is reached.

There are also 3 read only state that allow to monitoring every time the library status.

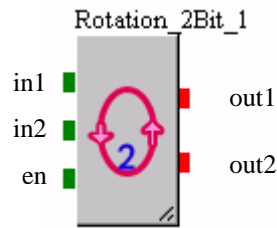
- *Action*: shows the library current status; can have values from 0 to 4 (0 = Stopped, 1 = Increase, 2 = Decrease, 3 = Minimum, 4 = Maximum).
- *Current_Position*: shows the library current position in percentage compared to the device total run.
- *Current_Run_Time*: shows the time in seconds where the actuator moves in one direction; it will be reset when the direction is reversed.

Note

The library is a template.

Part of the library is inside 100ms task. If the actuator command time is slow, could be necessary to command the corresponding outputs inside 100ms task, must consider the on/off relay times and these times should not be overly stressed.

Rotation_2Bits



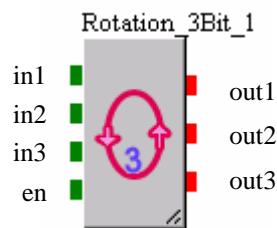
<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in1</i>	CJ_BIT	0..1	Input 1
<i>in2</i>	CJ_BIT	0..1	Input 2
<i>en</i>	CJ_BIT	0..1	Enabling rotation input

<i>Outputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out1</i>	CJ_BIT	0..1	Output 1
<i>out2</i>	CJ_BIT	0..1	Output 2

Description

The *Rotation_2Bits* library allows to manually rotate the two digital devices connected to *in1* and *in2* inputs. Rotation is enabled by a digital pulse on *en* input. This library is used to balance the working time of modular single parts as compressors, boilers, fans, etc.

Rotation_3Bits



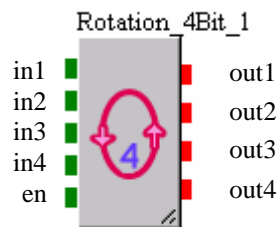
<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in1</i>	CJ_BIT	0..1	Input 1
<i>in2</i>	CJ_BIT	0..1	Input 2
<i>in3</i>	CJ_BIT	0..1	Input 3
<i>en</i>	CJ_BIT	0..1	Enabling rotation input

<i>Outputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out1</i>	CJ_BIT	0..1	Output 1
<i>out2</i>	CJ_BIT	0..1	Output 2
<i>out3</i>	CJ_BIT	0..1	Output 3

Description

The *Rotation_3Bits* library allows the manually rotate 3 digital devices connected to *in1*, *in2* and *in3* inputs. The rotation is enabled by a digital pulse on *en* input. When the *en* input is enabled, the *in1* moves to *out2* output, *in2* to *out3* and *in3* to *out1*. At the next pulse in *en*, *in1* will move to *out3*, *in2* to *out1* and *in3* to *out2*. This library is usually used to balance the working time of the modular single parts as compressors, boilers, fans, etc.

Rotation_4Bits



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in1</i>	CJ_BIT	0..1	Input 1
<i>in2</i>	CJ_BIT	0..1	Input 2
<i>in3</i>	CJ_BIT	0..1	Input 3
<i>in4</i>	CJ_BIT	0..1	Input 4
<i>en</i>	CJ_BIT	0..1	Enabling rotation input

<i>Outputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out1</i>	CJ_BIT	0..1	Output 1
<i>out2</i>	CJ_BIT	0..1	Output 2
<i>out3</i>	CJ_BIT	0..1	Output 3
<i>out4</i>	CJ_BIT	0..1	Output 4

Description

The *Rotation_4Bits* library allows to manually rotate 4 digital devices connected to *in1*, *in2*, *in3* and *in4* inputs. The rotation is enabled by a digital pulse on *en* input. When the *en* input is enabled the *in1* input moves to *out2* output, *in2* to *out3*, *in3* to *out4* and *in4* to *out1*. At the next pulse in *en*, *in1* will move to *out3*, *in2* to *out4*, *in3* to *out1* and *in4* to *out2*.

This library is usually used to balance the working time of modular single parts as compressors, boilers, fans, etc.

Sample and Hold



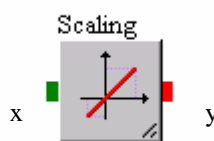
<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>in</i>	CJ_SHORT	-32768..32767	Input value
<i>sample</i>	CJ_BIT	0..1	Sampling input

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_SHORT	-32768..32767	Sampled value.

Descrizione

The *Sample_And_Hold* library allows to sample a *in* input value enabling the *sample* input. Particularly while *sample* input is enabled, the library output value corresponds to the *in* input, but when is turned off the last *sample* input is kept in output.

Scaling



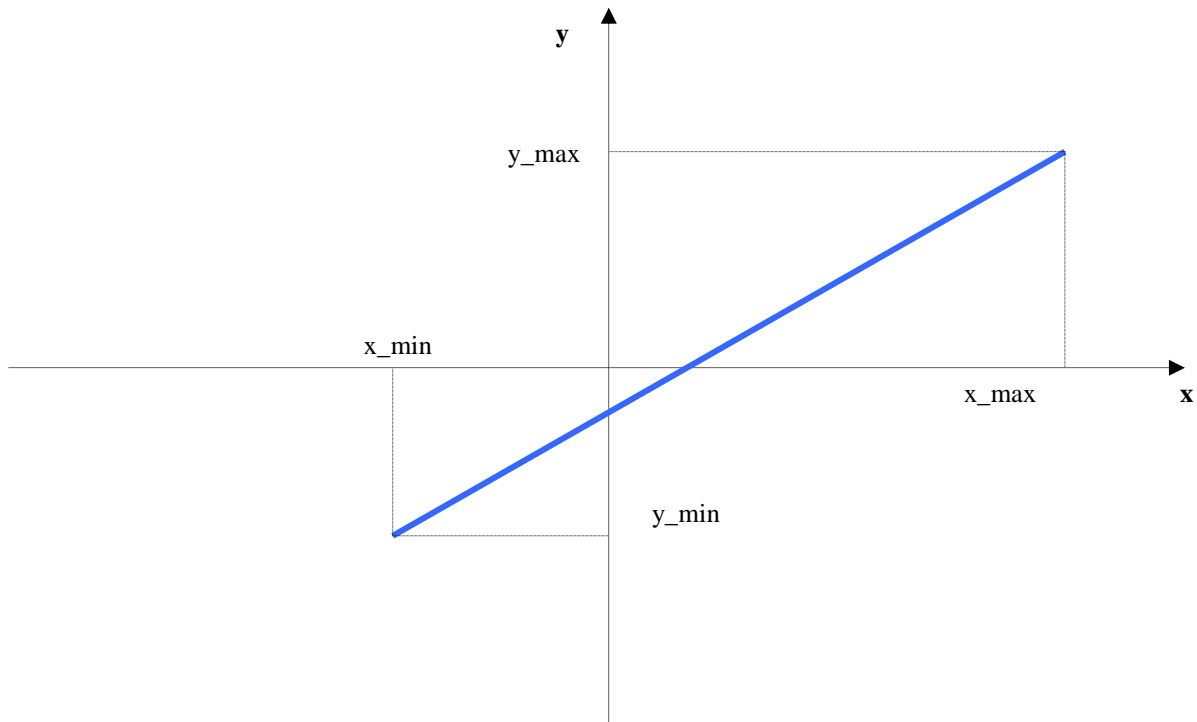
<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>x</i>	CJ_SHORT	-32768..32767	Input value

<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>y</i>	CJ_SHORT	-32768 .. 32767	Output value

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>x_min</i>	CJ_SHORT	-32768..32767	0	Input minimum value
<i>x_max</i>	CJ_SHORT	-32768..32767	100	Input maximum value
<i>y_min</i>	CJ_SHORT	-32768..32767	0	Output minimum value
<i>y_max</i>	CJ_SHORT	-32768..32767	100	Output maximum value

Description

The *Scaling* library is used to recalculate a input on a linear scale of values between two values. Exploring the library can be configured the input values interval (*x_min* e *x_max*) and the output values interval (*y_min* e *y_max*).



The output value corresponds to the Y coordinate of the line that passes from the two points defined by the parameters corresponding to X input value.

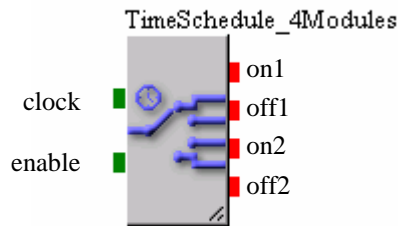
Note: it is worthwhile to assign the precision property to the parameters *x_min*, *x_max*, *y_min* and *y_max* in analogy to the input *x* and to the output *y*.

For example if wanted to convert an input in current 4-20 mA with precision=2 in a pressure 0-7 bar with precision=1 it will be opportune to set the following values:

$x_{min} = 4.00$
 $x_{max} = 20.00$
 $y_{min} = 0.0$
 $y_{max} = 7.0$

The values out of limits will be forced inside the limits.

Time_Schedule_4Modules



<i>Inputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Clock</i>	CJ_DATETIME	0..2147483647	System clock input
<i>enable</i>	CJ_BIT	0..1	Enabling input

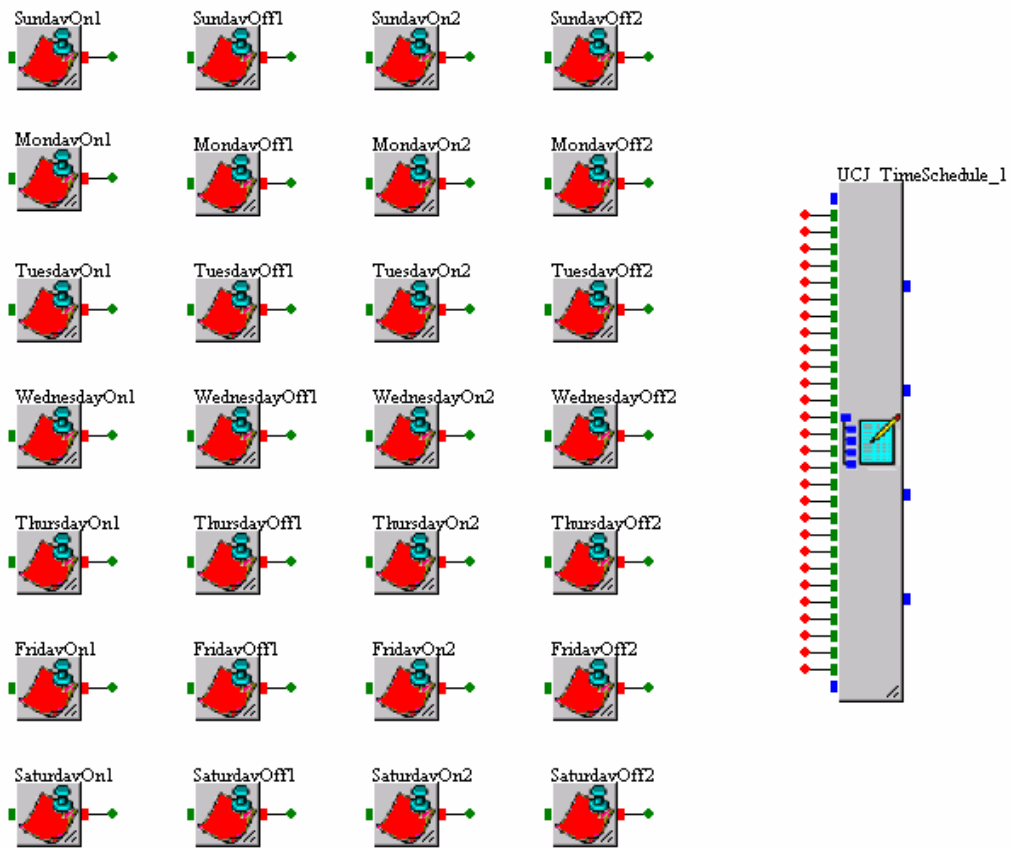
<i>Outputs</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>on1</i>	CJ_BIT	0..1	First on
<i>off1</i>	CJ_BIT	0..1	First off
<i>on2</i>	CJ_BIT	0..1	Second on
<i>off2</i>	CJ_BIT	0..1	Second off

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>SundayOn1,</i> <i>MondayOn1,</i> <i>TuesdayOn1,</i> <i>WednesdayOn1,</i> <i>ThursdayOn1</i> <i>FridayOn1,</i> <i>SaturdayOn1</i>	CJ_TIME	00:00..23:59	08:00	First on of Sunday, Monday Tuesday, Wednesday, Thursday, Friday, Saturday.

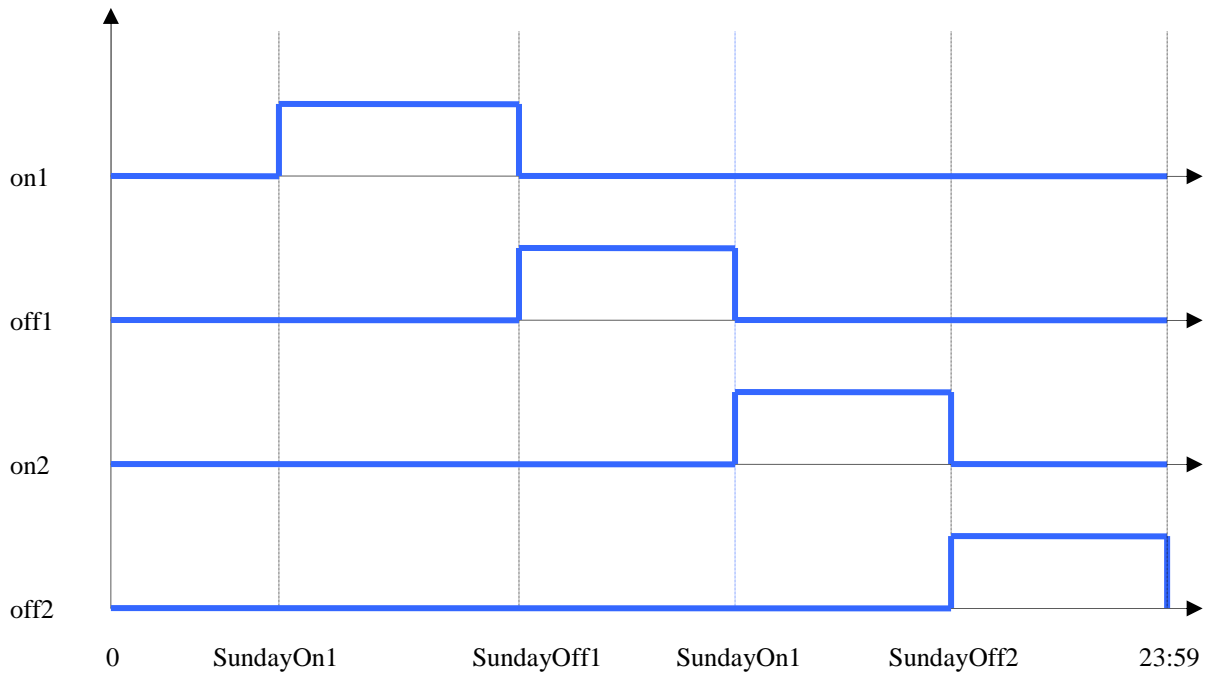
<i>SundayOff1, MondayOff1, TuesdayOff1, WednesdayOff1, ThursdayOff1 FridayOff1, SaturdayOff1</i>	CJ_TIME	00:00..23:59	17:00	First off of Sunday, Monday Tuesday, Wednesday, Thursday, Friday, Saturday.
<i>SundayOn2, MondayOn2, TuesdayOn2, WednesdayOn2, ThursdayOn2 FridayOn2, SaturdayOn2</i>	CJ_TIME	00:00..23:59	17:00	Second on of Sunday, Monday Tuesday, Wednesday, Thursday, Friday, Saturday.
<i>SundayOff2, MondayOff2, TuesdayOff2, WednesdayOff2, ThursdayOff2 FridayOff2, SaturdayOff2</i>	CJ_TIME	00:00..23:59	17:00	Second off of Sunday, Monday Tuesday, Wednesday, Thursday, Friday, Saturday.

Descrizione

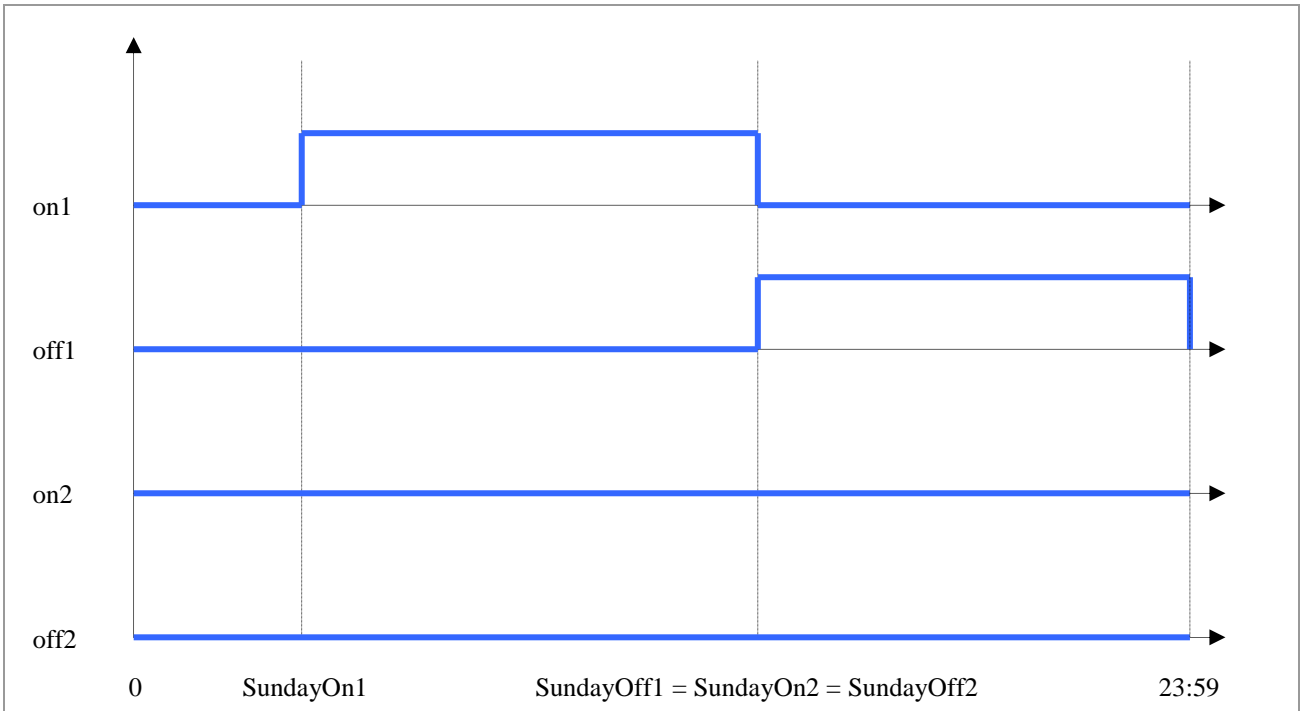
The *TimeSchedule_4Modules* library have two inputs: the *clock* input is connected to system clock and the *enable* input allows to enable or disable the clock.
 The four digital outputs represents the first on, the first off, the second on and the second off in a day.
 Inside the template type library there are 4 parameters per day that allow to specify the times that the on/off must happen in the day.



If 2 daily cycles are defined, the library works as the following chart:



If is requested only one off cycle, after setting the first on and first off value, the second on and the second off must be set at the same value as the first off.

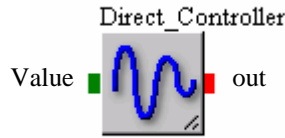


Notes

The library is a template.

1.2 Regulators

Direct_Controller



<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Value</i>	CJ_SHORT	-3276.8..3276.7	Input value

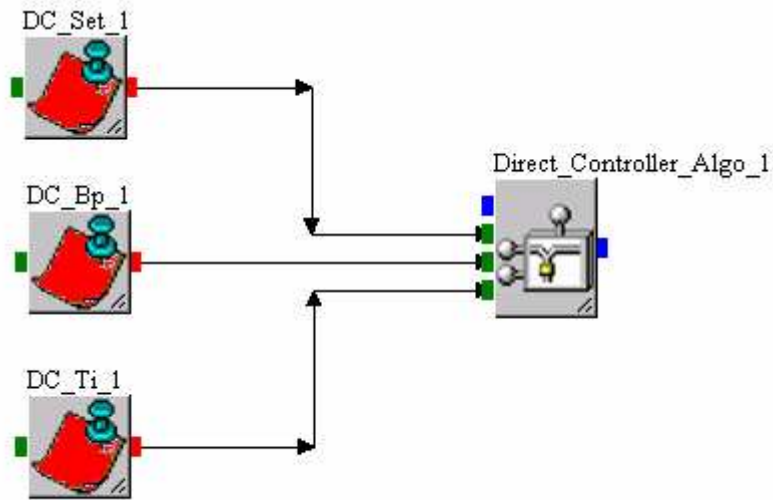
<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_WORD	0..100.00	Returns the value of analog output (0%-100%)

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>DC_Set</i>	CJ_SHORT	-1000.0..1000.0	20.0	Set-point
<i>DC_Bp</i>	CJ_WORD	0..1000.0	10.0	Proportional band.
<i>DC_Ti</i>	CJ_WORD	0..1000	300	Integral time (seconds)

Description

The *Direct_Controller* library manage the feed forward proportional-integrative control with analog output function.
 From an input value it calculates the corresponding analog output with values between 0.00% to 100.00%.

This library is a template where can be configured the corresponding parameters.

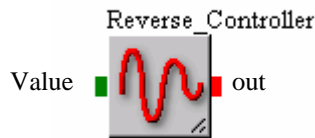


For further information check the standard *PI* library.

Note

The *DC_Set* e *DC_BP* parameters have a one decimal unit accuracy.

Reverse_Controller



<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Value</i>	CJ_SHORT	-3276.8..3276.7	Input value

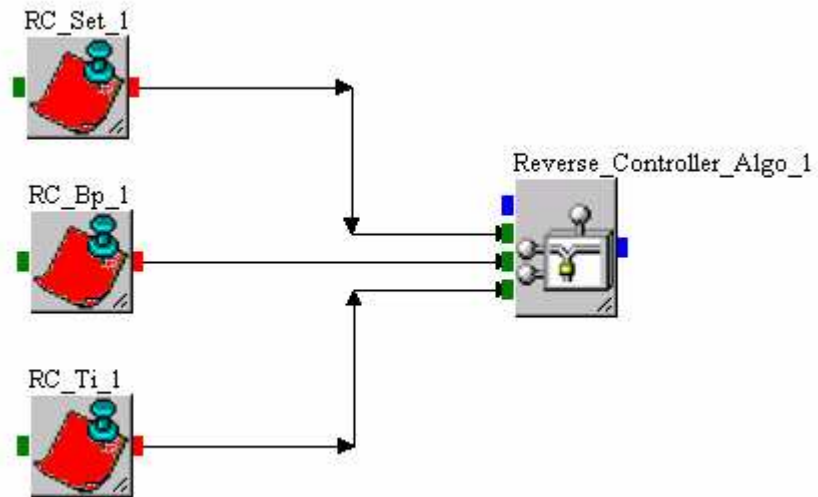
<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>out</i>	CJ_WORD	0..100.00	Returns an analog output value (0%-100%)

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
<i>RC_Set</i>	CJ_SHORT	-1000.0..1000.0	20.0	Set-point
<i>RC_Bp</i>	CJ_WORD	0..1000.0	10.0	Proportional band
<i>RC_Ti</i>	CJ_WORD	0..1000	300	Integral time (seconds)

Description

The *Reverse_Controller* library manage the feed backward proportional-integrative control with analog output function. From an input value it calculates the corresponding analog output with values between 0.00% to 100.00%.

This library is a template where can be configured the corresponding parameters.

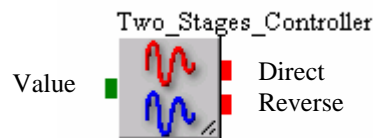


For further information check the standard *PI* library.

Note

The *RC_Set* e *RC_BP* parameters have a one decimal unit accuracy.

Two_Stages_Controller



<i>Input</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Value</i>	CJ_SHORT	-3276.8..3276.7	Input value

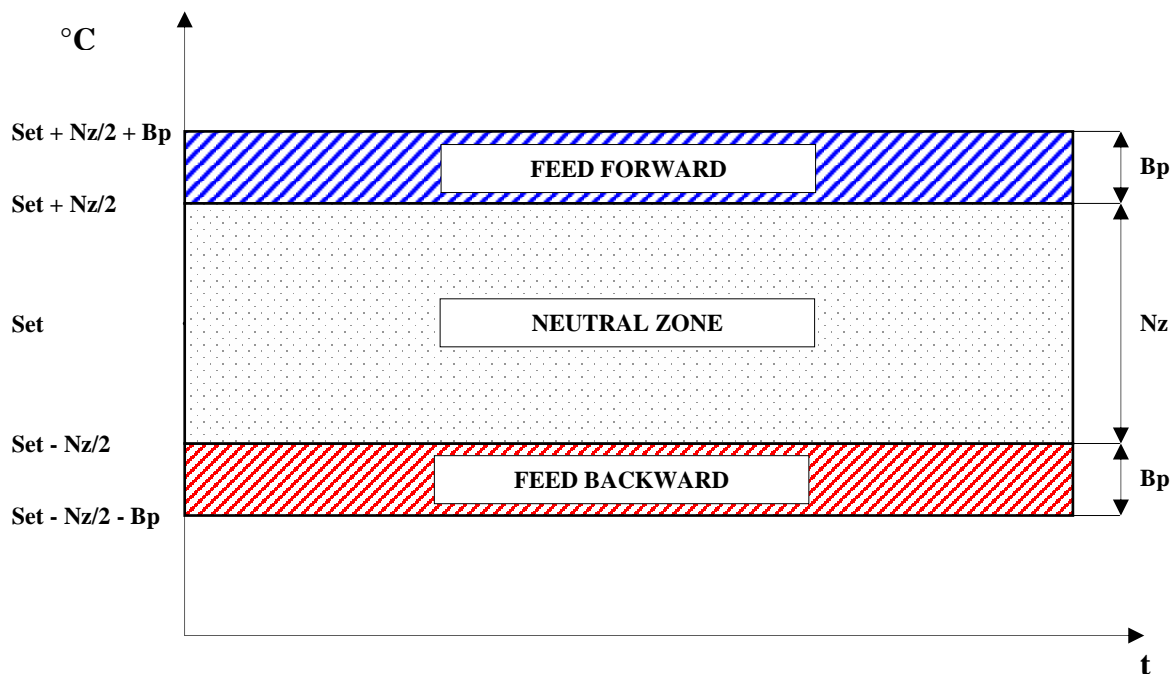
<i>Output</i>	<i>Type</i>	<i>Limits</i>	<i>Description</i>
<i>Direct</i>	CJ_WORD	0..100.00	Analog output to manage feed forward (0%-100%).

<i>Reverse</i>	CJ_WORD	0..100.00	Analog output to manage feed backward (0%-100%).
----------------	---------	-----------	--

<i>Parameters</i>	<i>Type</i>	<i>Limits</i>	<i>Default</i>	<i>Description</i>
2SC_Set	CJ_SHORT	-1000.0..1000.0	20.0	Set-point
2SC_Nz	CJ_WORD	0..1000.0	10.0	Neutral zone
2SC_Bp	CJ_WORD	0..1000.0	10.0	Proportional band
2SC_Ti	CJ_WORD	0..1000	300	Integral time (seconds)
2SC_Action_In_Nz	CJ_BIT	0..1	0	Neutral zone action

Descrizione

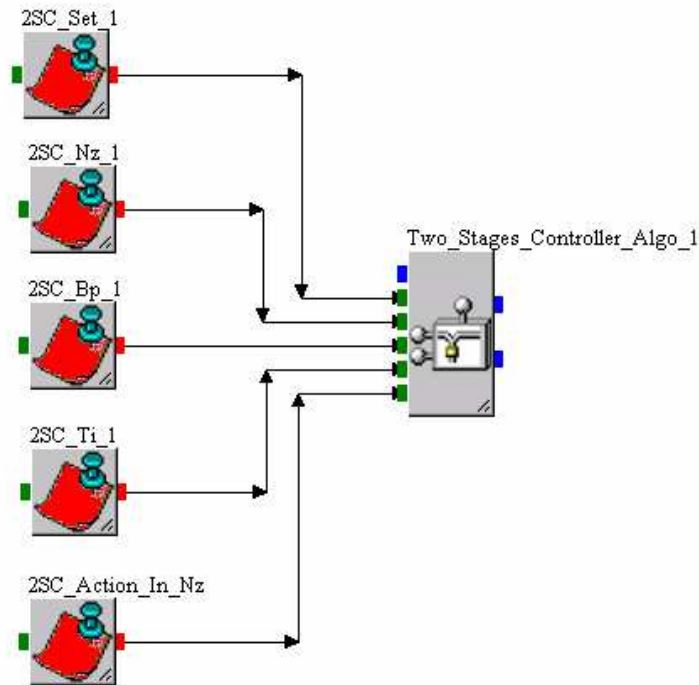
The *Two_Stages_Controller* library manage a two steps with neutral zone regulation function. With a input temperature value is made a proportional-integrative regulation based on configured values inside the library; there are two outputs, one of feed forward and one of feed backward.



Inside the library can be found the following configuration parameters:

- *2SC_Set*: Regulation reference set-point; it expresses the temperature that is desired to maintain in the environment, is defined in Celsius degrees with a one decimal unit accuracy.
- *2SC_Nz*: interval of neutral zone in which any type of action doesn't intervene (see *2SC_Action_In_Nz* parameter).
- *2SC_Bp*: It defines where the proportional action of the regulator is activated; the proportional band is equal for the feed forward and feed backward; is defined in Celsius degrees with a one decimal unit accuracy.
- *2SC_Ti*: It defines the time used by the integrative component of the regulator. If this parameter is set zero, the library it will behave as a proportional type simple regulator.

- *2SC_Action_In_Nz*: It allows to decide the behavior of the regulator in the neutral zone. If this parameter is activated, the regulator can operate in the neutral zone with the feed forward, the feed backward or both according to the situation and configuration. To do that in the neutral zone the regulator doesn't operate with any action, disable this parameter.



In base to input value and the parameters configuration, the library calculates an analog output with values from 0.00% to 100.00%. it's composed of:

- A directly proportional contribution to calculate error by the difference among the input temperature and the desired temperature.
- A integrative contribution of the *Ti* time.

Note

For further information check the standard PI library.
The *2SC_Set*, *2SC_Nz* e *2SC_BP* parameters have a one decimal unit accuracy.

UNI-PRO HVAC LIBRARIES MANUAL

UNI-PRO - HVAC libraries manual.

Version 1.0 - August 2010.

Code 114UPROHLE10.

File 114UPROHLE10.pdf.

This publication is the exclusive property of Evco. Evco forbids any form of reproduction and publication, unless specially authorised by Evco itself. Evco declines any responsibility regarding characteristics, technical data or any mistakes contained in this publication or consequential from usage of the same. Evco cannot be held responsible for any damages caused by non-compliance with warnings. Evco reserves the right to make any changes without previous notice and at any time, without prejudice to essential characteristics of functionality and safety.



HEADQUARTERS

Evco

Via Mezzaterra 6, 32036 Sedico Belluno ITALY
Tel. +39 0437-852468
Fax +39 0437-83648
info@evco.it
www.evco.it

OVERSEAS OFFICES

Control France

155 Rue Roger Salengro, 92370 Chaville Paris FRANCE
Tel. 0033-1-41159740
Fax 0033-1-41159739
control.france@wanadoo.fr

Evco Latina

Larrea, 390 San Isidoro, 1609 Buenos Aires ARGENTINA
Tel. 0054-11-47351031
Fax 0054-11-47351031
evcolatina@anykasrl.com.ar

Evco Pacific

59 Premier Drive Campbellfield, 3061, Victoria Melbourne, AUSTRALIA
Tel. 0061-3-9357-0788
Fax 0061-3-9357-7638
everycontrol@pacific.com.au

Evco Russia

111141 Russia Moscow 2-oy Proezd Perova Polya 9
Tel. 007-495-3055884
Fax 007-495-3055884
info@evco.ru

Every Control do Brasil

Rua Marino Félix 256, 02515-030 Casa Verde São Paulo SÃO PAULO BRAZIL
Tel. 0055-11-38588732
Fax 0055-11-39659890
info@everycontrol.com.br

Every Control Norden

Cementvägen 8, 136 50 Haninge SWEDEN
Tel. 0046-8-940470
Fax 0046-8-6053148
mail2@unilec.se

Every Control Shangai

B 302, Yin Hai Building, 250 Cao Xi Road, 200235 Shangai CHINA
Tel. 0086-21-64824650
Fax 0086-21-64824649
evcosh@online.sh.cn

Every Control United Kingdom

Unit 19, Monument Business Park, OX44 7RW Chalgrove, Oxford, UNITED KINGDOM
Tel. 0044-1865-400514
Fax 0044-1865-400419
info@everycontrol.co.uk