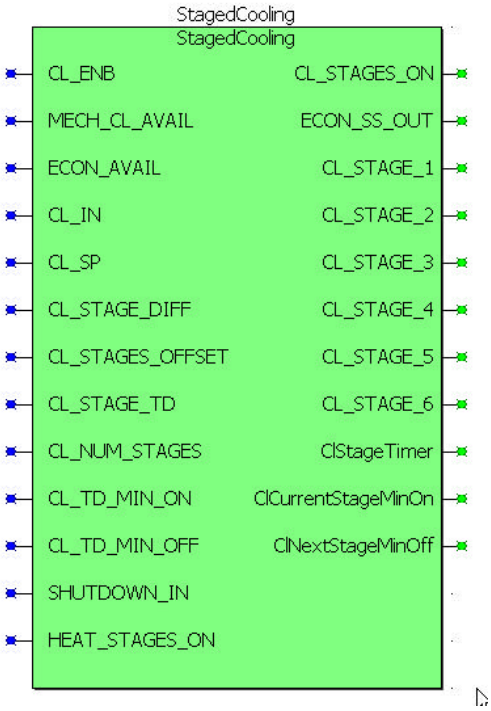
ADVANTECH
eAutomation

Function Library for Unitary Controllers



Staged Cooling FB

Description: Provides control of up to six mechanical cooling stages and enables an economizer based on the HVAC unit's zone temperature. The Zone Supervisor FB may enable or disables this FB based on the current cooling requirements of the zone (using CL_ENB).

Once enabled, this FB will cycle cooling stages on and off based on the Zone Temperature (CL_IN), Current Cooling Setpoint (CL_SP), and the parameters Cooling Stage Differential and Cooling Stage Offset.

If ECON_AVAIL is On and MECH_CL_AVAIL is On, ECON_SS_OUT will be used as the first cooling stage, and the effective number of cooling stages will be set to CL_NUM_STAGES + 1.

If only ECON_AVAIL is On ECON_SS_OUT will be used as the only cooling stage, and the effective number of cooling stages will be set to 1.

If only MECH_CL_AVAIL is On, ECON_SS_OUT will be set Off and the effective number of cooling stages will be set to CL_NUM_STAGES.

Note that the maximum possible number of cooling stages = 6, such that if an economizer used, only (5) additional mechanical cooling stages may be energized while the economizer is being used.

One input (HEAT STAGES ON) monitors if any Heat Stages are currently On. The FB's logic will not allow any cooling until all heating is off.

Usage Example: Two stage cooling with a 73 deg Setpoint, 1.0 degree differential, 0.5 degree offset, 10 minute Inter-stage Cooling Delay Time.

If the Zone Temperature is above the Current Cooling Setpoint + 50% of the differential, the first cooling stage will start (73 + 0.5 = 73.5 deg). Ten minutes later, when the Inter-stage Delay is satisfied, if the Zone Temperature is at least 0.5 degrees above the Stage 1 Setpoint (the "Offset" 73.5 + 0.5 = 74.0), the second stage will start.

Each stage must run at least the amount of time specified in Min On Time Delay. Once the minimum on time is satisfied, a stage may shut down when the Stage Setpoint minus 50% of the differential for that stage is satisfied.

In the above example, with a Current Zone Cooling Setpoint of 73 degrees, Cooling stages 1 & 2 will come on and off at the following temperatures

STAGE	ON Setpoint	OFF Setpoint
Cool 1	73.5 (cooling stpt + 50% diff)	72.5 (cooling stpt - 50% diff)
Cool 2	74.0 (cooling stpt + offset + 50% diff)	73.0 (cooling stpt + offset - 50% diff)

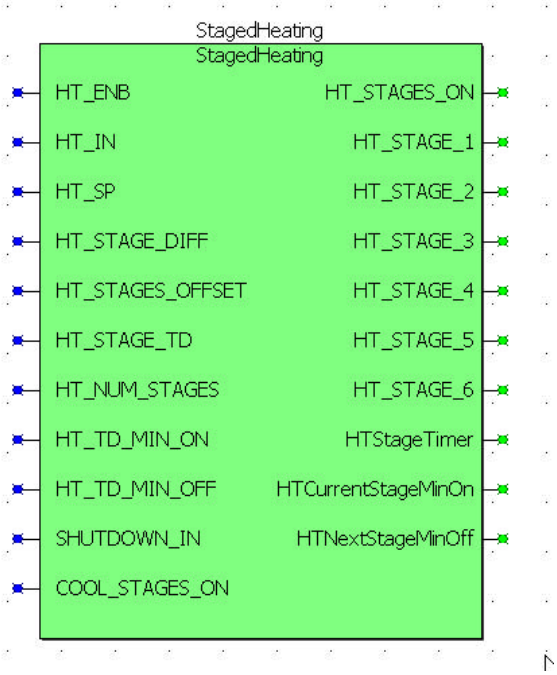
Note that this FB may be used for non-zone based cooling (such as Discharge Air Temp control). In this case it may be desirable to set CL STAGES OFFSET to 0.0 and rely on the time delay only (CL STAGE TD) for staging control.

INPUTS

CL_ENB :	BOOL := 0;(*Local Cooling Enable Input*)
MECH_CL_AVAIL :	BOOL := 1;(*Global Mechanical Cooling Available Input*)
ECON_AVAIL :	BOOL := 0;(*Global Economizer Cooling Available Input*)
CL_IN :	REAL;(*Input Value to Control to*)
CL_SP :	REAL;(*Setpoint to Control to*)
CL_STAGE_DIFF :	REAL := 1.0e0;(*Differential will be split above and below Stage Setpoint*)
CL_STAGES_OFFSET :	REAL := 0.5e0;(*For Stages 2 and higher, differential offset, cumulative*)
CL_STAGE_TD :	REAL := 1.2e2;(*Interstage Delay in Seconds*)
CL_NUM_STAGES :	INT := 6;(*Number of Mechanical Stages to Control*)
CL_TD_MIN_ON :	REAL := 1.2e2;(*Min On Time for each Stage in Seconds*)
CL_TD_MIN_OFF :	REAL := 1.2e2;(*Min Off Time for each Stage in Seconds*)
SHUTDOWN_IN :	BOOL := 0;(*Emergency Shutdown Input*)
HEAT_STAGES_ON :	INT := 0;(*Don't Run if Heat Stages On*)

OUTPUTS

CL_STAGES_ON :	INT;(*Actual number stages Now On*)
ECON_SS_OUT :	BOOL;(*Economizer Start/Stop Output*)
CL_STAGE_1 :	BOOL;(*Cool Stage 1 Output Point*)
CL_STAGE_2 :	BOOL;(*Cool Stage 2 Output Point*)
CL_STAGE_3 :	BOOL;(*Cool Stage 3 Output Point*)
CL_STAGE_4 :	BOOL;(*Cool Stage 4 Output Point*)
CL_STAGE_5 :	BOOL;(*Cool Stage 5 Output Point*)
CL_STAGE_6 :	BOOL;(*Cool Stage 6 Output Point*)
ClStageTimer :	REAL;(*Staging Time Elapsed Time*)
ClCurrentStageMinOn :	REAL;(*Current Stage Min On Timer Elapsed Time*)
ClNextStageMinOff :	REAL;(*Next Stage Min Off Timer Elapsed Time*)



Staged Heating FB

Description: Provides control of up to six heating stages based on the HVAC unit's zone temperature. The Zone Supervisor FB may enable or disables this FB based on the current heating requirements of the zone (using HT_ENB).

Once enabled, this FB will cycle heating stages on and off based on the Zone Temperature (HT_IN), Current Heating Setpoint (HT_SP), and the parameters Heating Stage Differential and Heating Stage Offset.

One input (COOL STAGES ON) monitors if any Cool Stages are currently On. The FB's logic will not allow any heating until all cooling is off.

Usage Example: Two stage heating with a 71 deg Setpoint, 1.0 degree differential, 0.5 degree offset, 10 minute Inter-stage Heating Delay Time.

If the Zone Temperature is below the Current Heating Setpoint + 50% of the differential, the first heating stage will start (71 - 0.5 = 70.5 deg). Ten minutes later, when the Inter-stage Delay is satisfied, if the Zone Temperature is at least 0.5 degrees below the Stage 1 Setpoint (the "Offset" 70.5 - 0.5 = 70.0), the second stage will start.

Each stage must run at least the amount of time specified in Min On Time Delay. Once the minimum on time is satisfied, a stage may shut down when the Stage Setpoint plus 50% of the differential for that stage is satisfied.

In the above example, with a Current Zone Heating Setpoint of 71 degrees, Heating stages 1 & 2 will come on and off at the following temperatures

STAGE	ON Setpoint	OFF Setpoint
Heat 1	70.5 (heating stpt - 50% diff)	71.5 (heating stpt + 50% diff)
Heat 2	70.0 (heating stpt - offset - 50% diff)	71.0 (heating stpt - offset + 50% diff)

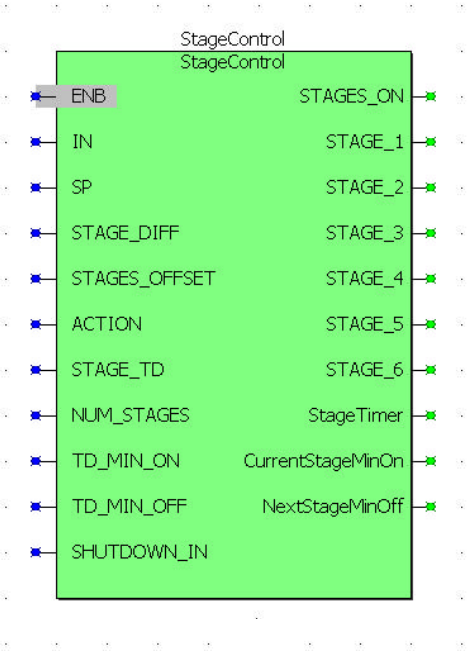
Note that this FB may be used for non-zone based heating (such as Discharge Air Temp control). In this case it may be desirable to set HT STAGES OFFSET to 0.0 and rely on the time delay only (HT STAGE TD) for staging control.

INPUTS

HT_ENB :	BOOL;(*Control Enable Input*)
HT_IN :	REAL;(*Input Value to Control to*)
HT_SP :	REAL;(*Setpoint to Control to*)
HT_STAGE_DIFF :	REAL := 1.0e0;(*Differential will be split above and below Stage Setpoint*)
HT_STAGES_OFFSET :	REAL := 0.5e0;(*For Stages 2 and higher, differential offset, cumulative*)
HT_STAGE_TD :	REAL := 1.2e2;(*Interstage Delay in Seconds*)
HT_NUM_STAGES :	INT := 6;(*Number of Stages to Control*)
HT_TD_MIN_ON :	REAL := 1.2e2;(*Min On Time for each Stage in Seconds*)
HT_TD_MIN_OFF :	REAL := 1.2e2;(*Min Off Time for each Stage in Seconds*)
SHUTDOWN_IN :	BOOL := 0;(*Emergency Shutdown Input*)
COOL_STAGES_ON :	INT := 0;(*Don't Run if Cool Stages On*)

OUTPUTS

HT_STAGES_ON :	INT;(*Actual number stages Now On*)
HT_STAGE_1 :	BOOL;(*Heat Stage 1 Output Point*)
HT_STAGE_2 :	BOOL;(*Heat Stage 2 Output Point*)
HT_STAGE_3 :	BOOL;(*Heat Stage 3 Output Point*)
HT_STAGE_4 :	BOOL;(*Heat Stage 4 Output Point*)
HT_STAGE_5 :	BOOL;(*Heat Stage 5 Output Point*)
HT_STAGE_6 :	BOOL;(*Heat Stage 6 Output Point*)
HTStageTimer :	REAL;(*Staging Time Elapsed Time*)
HTCurrentStageMinOn :	REAL;(*Current Stage Min On Timer Elapsed Time*)
HTNextStageMinOff :	REAL;(*Next Stage Min Off Timer Elapsed Time*)



Stage Control (Generic)

Description: Provides control of up to six stages of heating, cooling, pressure, etc., based on the *Control Input* (IN). The *Enable Input* (ENB) enables or disables the FB. When disabled, all stages will be set off.

When enabled, this block will cycle stages on and off based on the *Control Input* and *Setpoint*, and the parameters *Stage Differential* and *Stage Delay*.

(Control is the same as Staged Heating FB except that the control action may be set to either:

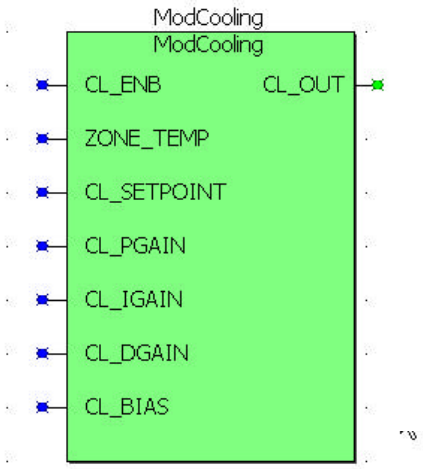
- 0 (Heating or Reverse Acting). Number of Stages increases as IN decreases.
- 1 (Cooling or Direct Acting). Number of Stages increases as IN increases.

INPUTS

ENB :	BOOL := 0;(*Enable Value*)
IN :	REAL;(*Input Value*)
SP :	REAL;(*Setpoint for Input*)
STAGE_DIFF :	REAL := 1.0e0;(*Differential will be split above and below Stage Setpoint*)
STAGES_OFFSET :	REAL := 0.5e0;(*For Stages 2 and higher, differential offset, cumulative*)
ACTION :	BOOL := 0;(*0 = Heating or Reverse 1= Cooling or Direct*)
STAGE_TD :	REAL := 1.2e2;(*Interstage Delay in Seconds*)
NUM_STAGES :	INT := 6;(*Number of Stages to Control*)
TD_MIN_ON :	REAL := 1.2e2;(*Min On Time for each Stage in Seconds*)
TD_MIN_OFF :	REAL := 1.2e2;(*Min Off Time for each Stage in Seconds*)
SHUTDOWN_IN :	BOOL := 0;(*Emergency Shutdown Input*)

OUTPUTS

STAGES_ON :	INT;(*Actual number stages Now On*)
STAGE_1 :	BOOL;(*Stage 1 Point*)
STAGE_2 :	BOOL;(*Stage 2 Point*)
STAGE_3 :	BOOL;(*Stage 3 Point*)
STAGE_4 :	BOOL;(*Stage 4 Point*)
STAGE_5 :	BOOL;(*Stage 5 Point*)
STAGE_6 :	BOOL;(*Stage 6 Point*)
StageTimer :	REAL;(*Elapsed Time for Interstage Timer*)
CurrentStageMinOn :	REAL;(*Elapsed Time for Current Stage Min On Timer in Seconds*)
NextStageMinOff :	REAL;(*Elapsed Time for Next Stage Min Off Timer in Seconds*)



Modulating Cooling

Description*:* Provides control of a modulating cooling device such as a valve or damper based on the HVAC unit's zone temperature. The Zone Supervisor FB may enable or disable the Modulating Cooling FB.

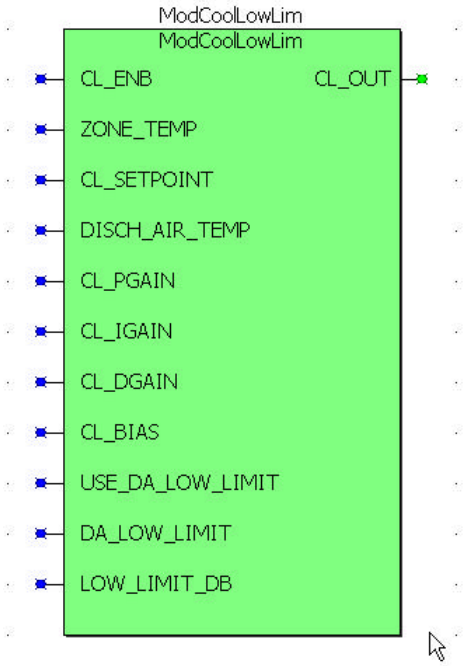
Once enabled, this block will modulate CL_OUT (damper or valve position 0 – 100%) based on the *Zone Temperature* and the *Current Cooling Setpoint* (CL_SETPOINT).

INPUTS

CL_ENB :	BOOL := 0;(*Input to Enable Cooling*)
ZONE_TEMP :	REAL;(*Zone Temperature Input*)
CL_SETPOINT :	REAL;(*Current Cooling Setpoint*)
CL_PGAIN :	REAL := 1.5E1;(*P Gain Setting*)
CL_IGAIN :	REAL := 1.0E-2;(*I Gain Setting*)
CL_DGAIN :	REAL := 0.0E0;(*D Gain Setting*)
CL_BIAS :	REAL := 2.0E1;(*Bias Setting*)

OUTPUTS

CL_OUT :	REAL;(*Cooling Output Signal (0 - 100%)*)



Modulating Cooling Low Limit

Description*:* Provides control of a modulating cooling device such as a valve or damper based on the HVAC unit's zone temperature *and a Discharge Air Temperature Low Limit*.

If "Use Disch Air Low Limit" is set On , the FB will "limit" the output signal such that the lowest allowed Discharge Air Temperature will be "Disch Air Low Limit" - "Low Limit Deadband".

The Zone Supervisor FB may enable or disable the Modulating Cooling Low Limit FB.

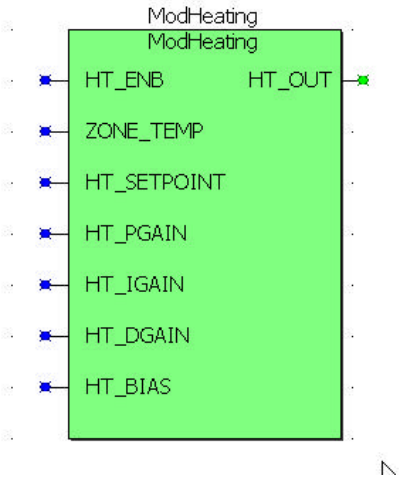
Once enabled, this block will modulate CL_OUT (damper or valve position 0 – 100%) based on the *Zone Temperature* and the *Current Cooling Setpoint (CL_SETPOINT),* limited by the Discharge Air Low Limit described above.

INPUTS

CL_ENB :	BOOL := 0;(*Input to Enable Cooling*)
ZONE_TEMP :	REAL;(*Zone Temperature Input*)
CL_SETPOINT :	REAL;(*Current Cooling Setpoint*)
DISCH_AIR_TEMP :	REAL;(*Discharge Air Temperature Input*)
CL_PGAIN :	REAL := 1.5E1;(*P Gain Setting*)
CL_IGAIN :	REAL := 1.0E-2;(*I Gain Setting*)
CL_DGAIN :	REAL := 0.0E0;(*D Gain Setting*)
CL_BIAS :	REAL := 2.0E1;(*Bias Setting*)
USE_DA_LOW_LIMIT :	BOOL := 0;(*0=Don't use Low DA Limit, 1=Use Low DA Limit*)
DA_LOW_LIMIT :	REAL := 5.0e1;(*Cooling Discharge Air Low Temperature Limit*)
LOW_LIMIT_DB :	REAL := 2.0E0;(*Low Limit Deadband (SP + / -)*)

OUTPUTS

CL_OUT :	REAL;(*Cooling Output Signal (0 - 100%)*)



Modulating Heating

Description*:* Provides control of a modulating heating device such as a valve or damper based on the HVAC unit's zone temperature. The Zone Supervisor FB may enable or disable the Modulating Heating FB.

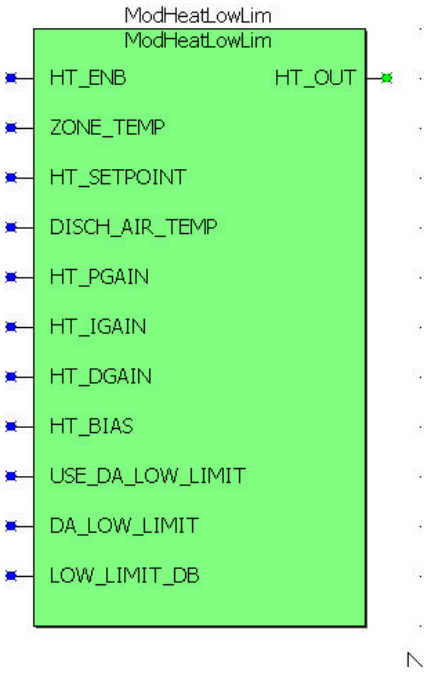
Once enabled, this FB will modulate HT_OUT (damper or valve position 0 – 100%) based on the *Zone Temperature* and the *Current Heating Setpoint* (HT_SETPOINT).

INPUTS

HT_ENB :	BOOL := 0;(*Input to Enable Heating*)
ZONE_TEMP :	REAL;(*Zone Temperature Input*)
HT_SETPOINT :	REAL;(*Current Heating Setpoint*)
HT_PGAIN :	REAL := 1.5E1;(*P Gain Setting*)
HT_IGAIN :	REAL := 1.0E-2;(*I Gain Setting*)
HT_DGAIN :	REAL := 0.0E0;(*D Gain Setting*)
HT_BIAS :	REAL := 2.0E1;(*Bias Setting*)

OUTPUTS

HT_OUT :	REAL;(*Heating Output Signal (0 - 100%)*)



Modulating Heating Low Limit

Description*:* Provides control of a modulating heating device such as a valve or damper based on the HVAC unit's zone temperature *and a Discharge Air Temperature Low Limit*.

If "Use Disch Air Low Limit" is set On , the FB will "limit" the output signal such that the lowest allowed Discharge Air Temperature will be "Disch Air Low Limit" - "Low Limit Deadband".

The Zone Supervisor FB may enable or disable the Modulating Heating Low Limit FB.

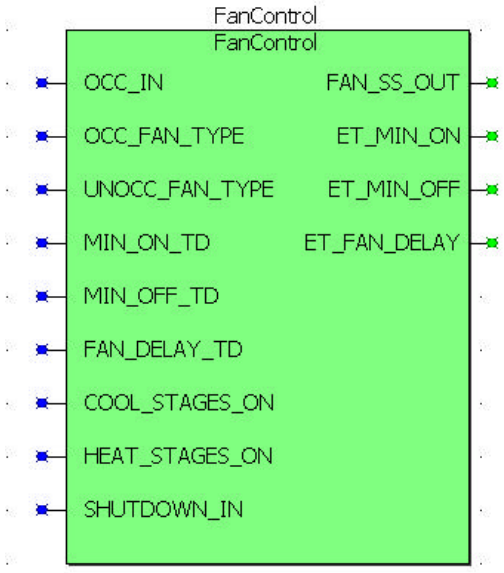
Once enabled, this block will modulate HT_OUT (damper or valve position 0 – 100%) based on the *Zone Temperature* and the *Current Heating Setpoint (HT_SETPOINT),* limited by the Discharge Air Low Limit described above.

INPUTS

HT_ENB :	BOOL := 0;(*Input to Enable Heating*)
ZONE_TEMP :	REAL;(*Zone Temperature Input*)
HT_SETPOINT :	REAL;(*Current Heating Setpoint*)
DISCH_AIR_TEMP :	REAL;(*Discharge Air Temperature Input*)
HT_PGAIN :	REAL := 1.5E1;(*P Gain Setting*)
HT_IGAIN :	REAL := 1.0E-2;(*I Gain Setting*)
HT_DGAIN :	REAL := 0.0E0;(*D Gain Setting*)
HT_BIAS :	REAL := 2.0E1;(*Bias Setting*)
USE_DA_LOW_LIMIT :	BOOL := 0;(*0=Don't use Low DA Limit, 1=Use Low DA Limit*)
DA_LOW_LIMIT :	REAL := 5.0e1;(*Heating Discharge Air Low Temperature Limit*)
LOW_LIMIT_DB :	REAL := 2.0E0;(*Low Limit Deadband (above & below Limit)*)

OUTPUTS

HT_OUT :	REAL;(*Heating Output Signal (0 - 100%)*)



Fan Control

Description*:* Provides On/Off control of a single speed fan. The Zone Supervisor block may set the Occupancy Mode (OCC_IN On= Occupied, Off= Unoccupied).

The Fan behaves differently during Occupied and Unoccupied modes based on the setting of OCC_FAN_TYPE and UNOCC_FAN_TYPE.

If fan type during the current OCC mode (Unocc or Occ) is Continuous, the Fan runs continuously during that OCC mode. If fan type is Auto, the Fan will start whenever COOL_STAGES_ON or HEAT_STAGES_ON > 0 and min Off time is satisfied. The fan will stop when all stages are off and Fan Delay Time Delay has been satisfied, and Min On time is satisfied.

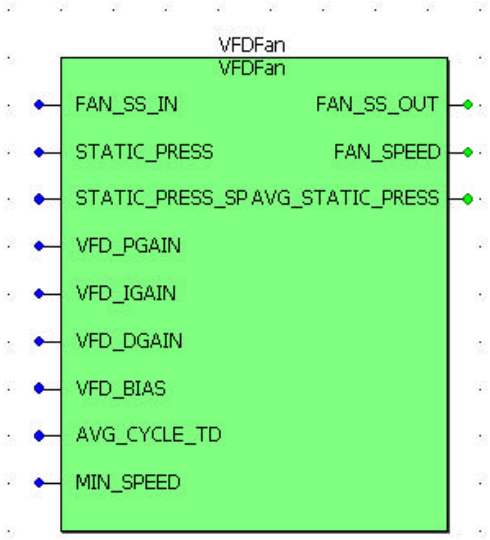
If SHUTDOWN_IN is On, the fan will shutdown immediately, regardless of timer status.

INPUTS

```
OCC_IN :          BOOL;(*Sched or Opt SS Input  0 = Sched Off, 1 = Sched On or Opt Start/Stop ON*)
OCC_FAN_TYPE :    INT := 0;(*0=Continuous, 1=Auto, 2= Cont Cool, Auto Heat*)
UNOCC_FAN_TYPE :  INT := 1;(*0=Continuous, 1=Auto, 2= Cont Cool, Auto Heat*)
MIN_ON_TD :       REAL := 1.2e2;(*Minimum On Time Delay in Seconds*)
MIN_OFF_TD :      REAL := 1.2e2;(*Minimum Off Time Delay in Seconds*)
FAN_DELAY_TD :    REAL := 6.0e1;(*Fan Off Time Delay in Seconds*)
COOL_STAGES_ON :  INT := 0;(*Number of Cool Stages ON (or Analog Cool Output A->D)*)
HEAT_STAGES_ON :  INT := 0;(*Number of Heat Stages ON (or Analog Heat Output A->D)*)
SHUTDOWN_IN :     BOOL := 0;(*Emergency Shutdown Input*)
```

OUTPUTS

```
FAN_SS_OUT :      BOOL;(*Fan Start/Stop Control Output*)
ET_MIN_ON :       REAL;(*Elapsed Time Min On Timer*)
ET_MIN_OFF :      REAL;(*Elapsed Time Min Off Timer*)
ET_FAN_DELAY :    REAL;(*Elapsed Time Fan Delay Off Timer*)
```



VFD Fan Control

Description*:* Used in conjunction with the Fan Control FB, this FB provides control of a VFD controlled Fan's Speed based on the HVAC unit's Supply Air Static Pressure. The Fan Speed will be modulated to maintain the Static Pressure Setpoint.

This FB gets Fan Start/Stop information from the Fan Control FB (Min On/Off times, etc. are controlled by that FB).

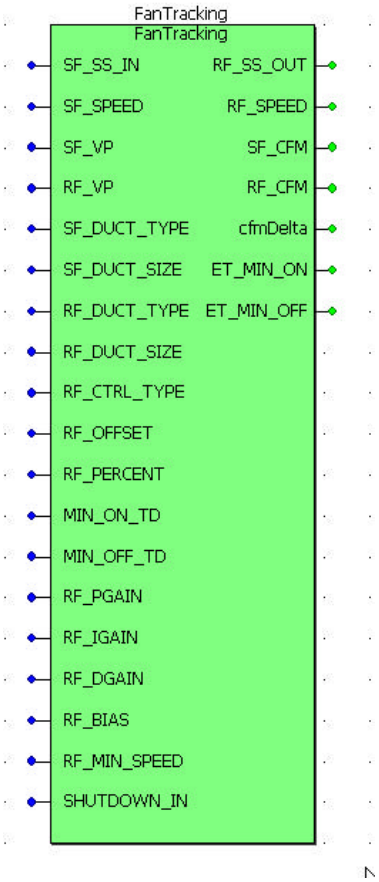
The Static Pressure Input will be averaged over time using the Sliding Window Averaging function to obtain a more stable input value to control to.

INPUTS

FAN_SS_IN :	BOOL;(*Fan Start/Stop Signal Input*)
STATIC_PRESS :	REAL;(*Static Pressure Input*)
STATIC_PRESS_SP :	REAL;(*Static Pressure Setpoint*)
VFD_PGAIN :	REAL := 1.5e1;(*VFD P Gain*)
VFD_IGAIN :	REAL := 1.0e-2;(*VFD I Gain*)
VFD_DGAIN :	REAL := 0.0e0;(*VFD D Gain*)
VFD_BIAS :	REAL := 3.0e1;(*VFD Bias*)
AVG_CYCLE_TD :	REAL := 1.0e0;(*Time between Averaging Cycles (secs)*)
MIN_SPEED :	REAL := 4.0e1;(*Minimum Fan Speed (0-100%)*)

OUTPUTS

FAN_SS_OUT :	BOOL;(*Fan Start/Stop Control*)
FAN_SPEED :	REAL;(*Fan Speed Control (0-100%)*)
AVG_STATIC_PRESS :	REAL;(*Averaged Static Pressure Signal*)



Fan Tracking

Description*:* Used in conjunction with the VFD Fan Control FB, this FB provides “Tracking” control of a Return Fan Start/Stop point and a Return Fan VFD Speed point based on either a percentage of Supply Fan speed, or a fixed CFM offset between Return Air CFM versus Supply Air CFM.

Usage Example: If the Return Fan Control Type is Percentage, and the value of Return Fan Tracking Percentage is 90%, the Return Fan Speed will always equal 90% of the Supply Fan Speed.

If the Return Fan Control Type is CFM Offset, and the value of Return Fan Tracking Offset CFM is 500 CFM, the Return Fan Speed will always be maintained such that the airflow in the Return Air-stream is 500 CFM lower than the airflow in the Supply Air-stream. This would indicate that 500 CFM of Outside Air is being introduced into the Supply Air-stream.

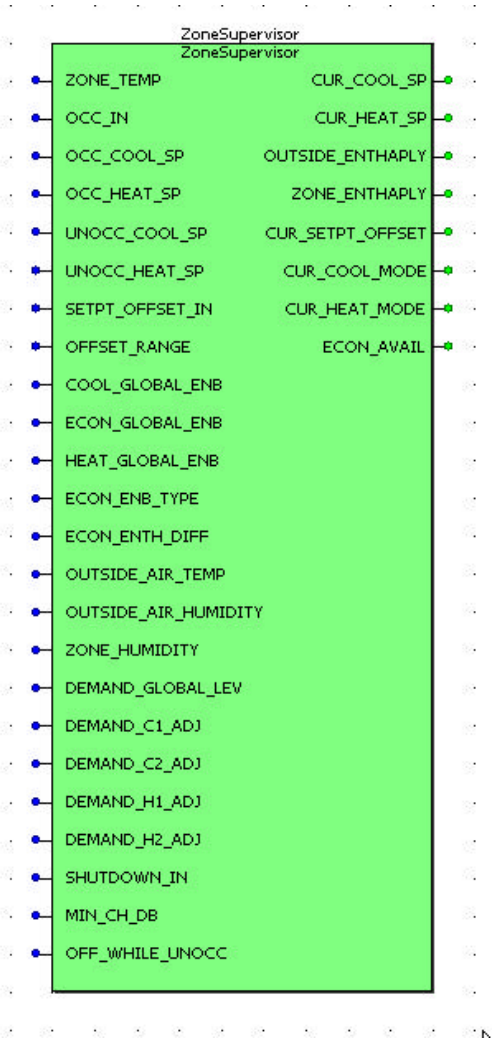
INPUTS

SF_SS_IN :	BOOL;(*SF Start Stop Input*)
SF_SPEED :	REAL;(*SF Speed Input*)
SF_VP :	REAL;(*SF Velocity Pressure*)
RF_VP :	REAL;(*RF Velocity Pressure*)
SF_DUCT_TYPE :	INT;(*SF Duct Type (0 = Rectangular, 1 = Round)*)
SF_DUCT_SIZE :	REAL;(*Duct Size (if Rect- Area in Sq Ft, if Round- Diameter in inches)*)
RF_DUCT_TYPE :	INT;(*RF Duct Type (0 = Rectangular, 1 = Round)*)
RF_DUCT_SIZE :	REAL;(*Duct Size (if Rect- Area in Sq Ft, if Round- Diameter in inches)*)
RF_CTRL_TYPE :	INT := 0;(*RF Control Type (0= Percent, 1= CFM Offset)*)
RF_OFFSET :	REAL := 5.0e2;(*If CFM Offset Type, enter CFM Offset*)
RF_PERCENT :	REAL := 9.0e1;(*If Percent of SF Type, enter Percent (0-100%)*)
MIN_ON_TD :	REAL := 1.2e2;(*Minimum On Time Delay in Seconds*)
MIN_OFF_TD :	REAL := 1.2e2;(*Minimum Off Time Delay in Seconds*)
RF_PGAIN :	REAL := 1.5e1;(*Return Fan P Gain Setting*)
RF_IGAIN :	REAL := 1.0e-2;(*Return Fan I Gain Setting*)
RF_DGAIN :	REAL := 0.0e0;(*Return Fan D Gain Setting*)
RF_BIAS :	REAL := 3.0e1;(*Return Fan Bias Setting*)
RF_MIN_SPEED :	REAL := 3.0e1;(*Return Fan Minimum Speed Setting*)
SHUTDOWN_IN :	BOOL := 0;(*Emergency Shutdown Input*)

OUTPUTS

RF_SS_OUT :	BOOL;(*RF Start Stop Control Output*)
RF_SPEED :	REAL;(*RF Speed Control Output*)

SF_CFM : REAL;(*SF CFM*)
RF_CFM : REAL;(*RF CFM*)
cfmDelta : REAL;(*CFM difference SF vs. RF*)
ET_MIN_ON : REAL;(*Elapsed Time Minimum On Timer*)
ET_MIN_OFF : REAL;(*Elapsed Time Minimum Off Timer*)



Zone Supervisor

Description: The Zone Supervisor FB reads in all building-wide information pertaining to the status of Schedules, Holidays, Free Cooling, Electrical Demand, Emergency and other conditions. It also contains all zone-specific setpoints and settings for how to respond to changes in the building-wide values.

Integrating building-wide and local information, it coordinates all function areas of the unitary controller (occupancy, heating, cooling, fan control, etc) enabling other control functions as needed.

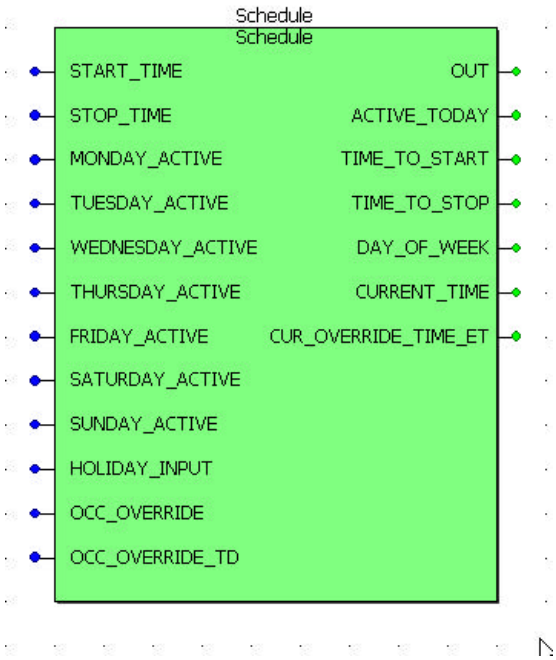
INPUTS

```
ZONE_TEMP :             REAL;(*Zone Temperature Input*)
OCC_IN :                BOOL;(*Occupancy Command Input*)
OCC_COOL_SP :           REAL := 7.3e1;(*Occupied Cooling Setpoint*)
OCC_HEAT_SP :           REAL := 7.0e1;(*Occupied Heating Setpoint*)
UNOCC_COOL_SP :         REAL := 7.8e1;(*Unoccupied Cooling Setpoint*)
UNOCC_HEAT_SP :         REAL := 6.5e1;(*Unoccupied Heating Setpoint*)
SETPT_OFFSET_IN :       REAL;(*Tenant Setpoint Adjust +/- Input (0-100%)*)
OFFSET_RANGE :          REAL := 2.0e0;(*+/- Number of Degrees Allowed Tenant Setpoint Adjust*)
COOL_GLOBAL_ENB :       BOOL := 1;(*Global Mech Cooling Lockout Input*)
ECON_GLOBAL_ENB :       BOOL := 1;(*Global Econ. Cooling Lockout Input*)
HEAT_GLOBAL_ENB :       BOOL := 1;(*Global Heating Lockout Input*)
ECON_ENB_TYPE :         INT := 0;(*Economizer Lock Out Type 0=No Econ,1=Dry Bulb,2=Enthalpy*)
ECON_ENTH_DIFF :        REAL := 2.0e0;(*For Enthalpy LO, damper enb if OA Enth < Zone Enth - Diff*)
OUTSIDE_AIR_TEMP :      REAL;(*Outside Air Temp (for Enthalpy Calc)*)
OUTSIDE_AIR_HUMIDITY :  REAL;(*Outside Air Humidity (for Enthalpy Calc)*)
ZONE_HUMIDITY :         REAL;(*Zone/RA Humidity (for Enthalpy Calc)*)
DEMAND_GLOBAL_LEV :     INT := 0;(*Global Demand Level Input*)
DEMAND_C1_ADJ :         REAL := 0.5e0;(*Cooling Setpoint Adjustment in response to Demand Level 1*)
DEMAND_C2_ADJ :         REAL := 1.25e0;(*Cooling Setpoint Adjustment in response to Demand Level 2*)
DEMAND_H1_ADJ :         REAL := 0.5e0;(*Heating Setpoint Adjustment in response to Demand Level 1*)
DEMAND_H2_ADJ :         REAL := 1.25e0;(*Heating Setpoint Adjustment in response to Demand Level 2*)
SHUTDOWN_IN :           BOOL := 0;(*Emergency Shutdown Input*)
MIN_CH_DB :             REAL := 2.0e0;(*Min Deadband allowed between Heat & Cool SPs*)
OFF_WHILE_UNOCC :       BOOL := 0;(*Unit is fully Off during Unocc Mode, no Unocc Setpoints are used*)
```

OUTPUTS

```
CUR_COOL_SP :       REAL;(*Current Zone Cooling Setpoint*)
CUR_HEAT_SP :       REAL;(*Current Zone Heating Setpoint*)
OUTSIDE_ENTHAPLY :  REAL;(*int var - Calculated Outside Enthalpy*)
ZONE_ENTHAPLY :     REAL;(*int var - Calculated Zone Enthalpy*)
CUR_SETPT_OFFSET :  REAL;(*Current Setpoint Adjust*)
CUR_COOL_MODE :     BOOL;(*Current Enable Signal to Cool Section*)
CUR_HEAT_MODE :     BOOL;(*Current Enable Signal to Heat Section*)
ECON_AVAIL :        BOOL;(*Economizer is enabled by Outside temp and/or Enthalpy*)
```

Schedule



Description: Provides scheduling capabilities within the unitary controller. The user may enter start and stop times, and identify which days of the week those times apply to. Multiple schedules may be used to handle different start and stop times on different days of the week.

Schedules have the ability to be linked to global holiday status values, and will output how many hours until the Next Start Time and Next Stop Time for use in Optimum Start and Stop calculations.

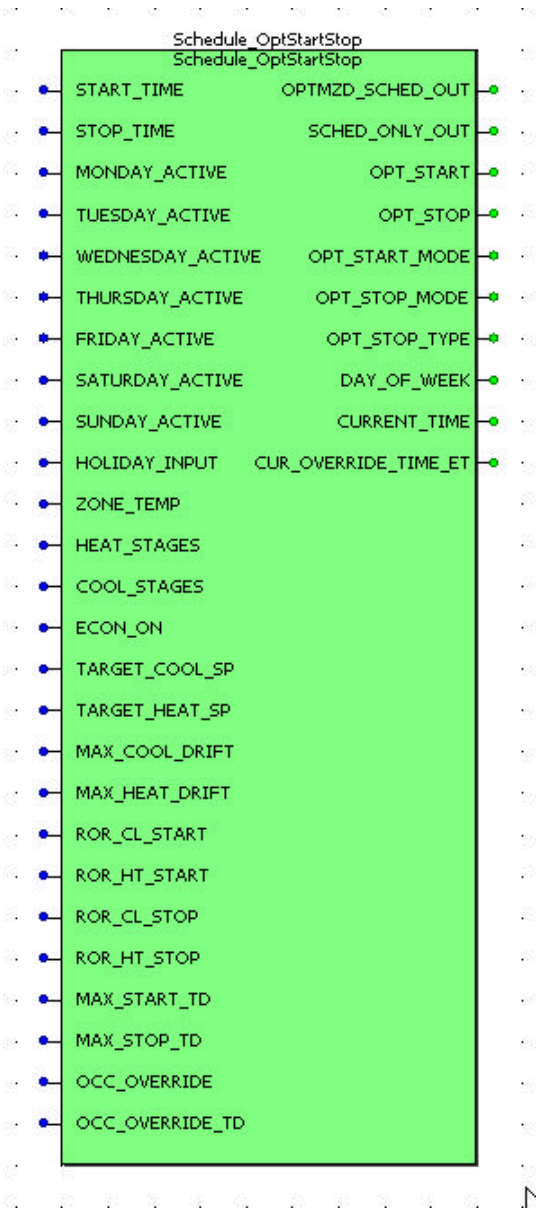
Schedules may also be linked to an OCCUPANCY OVERRIDE input. This is a momentary input which sets the Override Time timer to the value of OCC_OVERRIDE_TD (in minutes). If the Occupancy State is not currently Occupied (On), it will be set to Occupied until the timer expires.

INPUTS

START_TIME :	INT := 0800;(*Start Time for the Active Days*)
STOP_TIME :	INT := 1800;(*Stop Time for the Active Days*)
MONDAY_ACTIVE :	BOOL := 1;(*Schedule Active Monday No/Yes*)
TUESDAY_ACTIVE :	BOOL := 1;(*Schedule Active Tuesday No/Yes*)
WEDNESDAY_ACTIVE :	BOOL := 1;(*Schedule Active Wednesday No/Yes*)
THURSDAY_ACTIVE :	BOOL := 1;(*Schedule Active Thursday No/Yes*)
FRIDAY_ACTIVE :	BOOL := 1;(*Schedule Active Friday No/Yes*)
SATURDAY_ACTIVE :	BOOL := 1;(*Schedule Active Saturday No/Yes*)
SUNDAY_ACTIVE :	BOOL := 1;(*Schedule Active Sunday No/Yes*)
HOLIDAY_INPUT :	BOOL := 0;(*Holiday Today Input*)
OCC_OVERRIDE :	BOOL := 0;(*Afterhours Occupancy Override Input*)
OCC_OVERRIDE_TD :	REAL := 6.0e1;(*Occ Override Time Delay (in Minutes)*)

OUTPUTS

OUT :	BOOL;(*Schedule & Opt Start & Opt Stop Combined Output*)
ACTIVE_TODAY :	BOOL;(*Schedule Active for some period of today*)
TIME_TO_START :	REAL;(*Hours until Next Start Time if Schedule is Active Today*)
TIME_TO_STOP :	REAL;(*Hours until Next Stop Time if Schedule is Active Today*)
DAY_OF_WEEK :	INT;(*Current Day of Week Sun (1) - Sat (7)*)
CURRENT_TIME :	INT;(*Current System Time*)
CUR_OVERRIDE_TIME_ET :	REAL;(*Current Elapsed Override Time (in Minutes)*)



Schedule with Optimum Start / Optimum Stop

Description: In addition to all capabilities of the Schedule FB, this FB adds Optimum Start and Stop calculations.

Based on the user defined Cooling and Heating Rates of Recovery, this FB calculates the amount of Early Start Time required to achieve the Target Heat Setpoint (if zone temp is less than this) or the Target Cool Setpoint (if zone temp is greater than this). When the time until Occupancy is less than the calculated early start time, OPTMZD_SCHED_OUT is set On (as is OPT_START).

The FB also calculates the amount of Early Stop Time which will result in not exceeding the Target Heat Setpoint – MAX HEAT DRIFT, nor the Target Cool Setpoint + MAX COOL DRIFT. When the time until Unoccupied is less than the calculated early stop time, OPTMZD_SCHED_OUT is set Off (OPT_STOP is set On).

Schedule Only Out will be set On and Off based on the actual (not optimized) start and stop times.

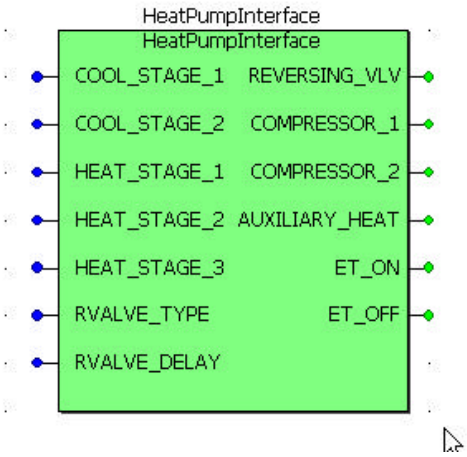
INPUTS

```
START_TIME :          INT := 0715;(*Start Time for the Active Days*)
STOP_TIME :           INT := 1830;(*Stop Time for the Active Days*)
MONDAY_ACTIVE :       BOOL := 1;(*Schedule Active Monday No/Yes*)
TUESDAY_ACTIVE :      BOOL := 1;(*Schedule Active Tuesday No/Yes*)
WEDNESDAY_ACTIVE :    BOOL := 1;(*Schedule Active Wednesday No/Yes*)
THURSDAY_ACTIVE :     BOOL := 1;(*Schedule Active Thursday No/Yes*)
FRIDAY_ACTIVE :       BOOL := 1;(*Schedule Active Friday No/Yes*)
SATURDAY_ACTIVE :     BOOL := 1;(*Schedule Active Saturday No/Yes*)
SUNDAY_ACTIVE :       BOOL := 1;(*Schedule Active Sunday No/Yes*)
```

HOLIDAY_INPUT :	BOOL := 0;(*Holiday Today Input*)
ZONE_TEMP :	REAL := 7.2E1;(*ZoneTemp*)
HEAT_STAGES :	INT;(*Heat Stages input from Heating FB (Any Int over 0) for Op Stop*)
COOL_STAGES :	INT;(*Cool Stages input from Heating FB (Any Int over 0) for Op Stop*)
ECON_ON :	BOOL;(*Econ Active input from Econ FB (Off or On) for Op Stop*)
TARGET_COOL_SP :	REAL := 7.4E1;(*Opt Start Cooling Target (May link to Cool SP)*)
TARGET_HEAT_SP :	REAL := 7.0E1;(*Opt Start Heating Target (May link to Heat SP)*)
MAX_COOL_DRIFT :	REAL := 1.5E0;(*Opt Stop Maximum allowed Cool Target to drift to*)
MAX_HEAT_DRIFT :	REAL := 1.5E0;(*Opt Stop Minimum allowed Heat Target to drift to*)
ROR_CL_START :	REAL := 2.5E0;(*Opt Start Cooling Rate of Recovery deg/hr*)
ROR_HT_START :	REAL := 2.5E0;(*Opt Start Heating Rate of Recovery deg/hr*)
ROR_CL_STOP :	REAL := 2.5E0;(*Opt Stop Cooling Rate of Recovery deg/hr*)
ROR_HT_STOP :	REAL := 2.5E0;(*Opt Stop Heating Rate of Recovery deg/hr*)
MAX_START_TD :	REAL := 1.0E0;(*Maximum allowed early start time (hours)*)
MAX_STOP_TD :	REAL := 1.0E0;(*Maximum allowed early stop time (hours)*)
OCC_OVERRIDE :	BOOL;(*Afterhours Occupancy Override Input*)
OCC_OVERRIDE_TD :	REAL := 6.0e1;(*Occ Override Time Delay (in Minutes)*)

OUTPUTS

OPTMZD_SCHED_OUT :	BOOL;(*Combined Sched & OpStart & OpStop Output to enable device*)
SCHED_ONLY_OUT :	BOOL;(*Schedule Only Output - On for all Scheduled Occupied Hours*)
OPT_START :	BOOL;(*On during Opt Start period*)
OPT_STOP :	BOOL;(*On during Opt Stop period*)
OPT_START_MODE :	INT;(*Current Opt Start Mode (0=off, 1=heat waiting,2=cool waiting, 3=heat OpStart Active, 4=cool OpStart Active*)
OPT_STOP_MODE :	INT;(*Current Opt Stop Mode (0=off, 1=heat waiting,2=cool waiting, 3=heat OpStop Active, 4=cool OpStop Active*)
OPT_STOP_TYPE :	INT;(*Current Opt Stop Type set by monitoring heat/cool/econ operation (0=none,1=heat,2=cool)*)
DAY_OF_WEEK :	INT;(*Current Day of Week Sun (1) - Sat (7)*)
CURRENT_TIME :	INT;(*Current System Time*)
CUR_OVERRIDE_TIME_ET :	REAL;(*Current Elapsed Override Time (in Minutes)*)



Heat Pump Interface

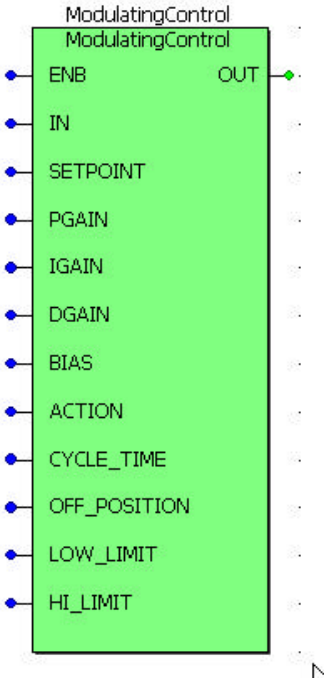
Description: Provides control of Heat Pump points based on outputs from Staged Cooling and Staged Heating FBs and the values of the listed configuration parameters.

INPUTS

COOL_STAGE_1 :	BOOL;(*Input from Cool Stage 1*)
COOL_STAGE_2 :	BOOL;(*Input from Cool Stage 2*)
HEAT_STAGE_1 :	BOOL;(*Input from Heat Stage 1*)
HEAT_STAGE_2 :	BOOL;(*Input from Heat Stage 2*)
HEAT_STAGE_3 :	BOOL;(*Input from Heat Stage 3*)
RVALVE_TYPE :	BOOL := 0;(*0 = On w/ Cooling 1 = On w/ Heating*)
RVALVE_DELAY :	REAL := 3.0E1;(*Delay between Valve & Compressor in Seconds*)

OUTPUTS

REVERSING_VLV :	BOOL;(*Reversing Valve Control Output*)
COMPRESSOR_1 :	BOOL;(*Compressor 1 Control Output*)
COMPRESSOR_2 :	BOOL;(*Compressor 2 Control Output*)
AUXILIARY_HEAT :	BOOL;(*Auxiliary Heat Control Output*)
ET_ON :	REAL;(*Elapsed Time Rev Valve On Delay*)
ET_OFF :	REAL;(*Elapsed Time Rev Valve Off Delay*)



Modulating Control (Generic)

Description*:* Provides control of any modulating device such as a cooling or heating valve or damper, or a pressure controlled VFD, based on the *Control Input* (IN). The *Enable Input* (ENB) enables or disables the block. When disabled, OUT will be set to 0.0%

When enabled, the block will modulate OUT (0 – 100%) based on IN and SETPOINT.

(Control ACTION may be set to either:

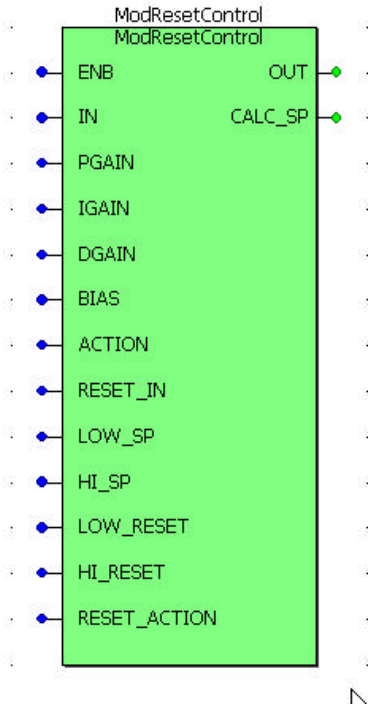
- 0 (Heating or Reverse Acting). OUT increases as IN decreases.
- 1 (Cooling or Direct Acting). OUT increases as IN increases.

INPUTS

ENB :	BOOL := 0;(*Input to enable control*)
IN :	REAL;(*Input to be controlled*)
SETPOINT :	REAL;(*Setpoint to control to*)
PGAIN :	REAL := 1.5e1;(*P Gain Setting*)
IGAIN :	REAL := 1.0e-2;(*I Gain Setting*)
DGAIN :	REAL := 0.0e0;(*D Gain Setting*)
BIAS :	REAL := 2.0e1;(*Bias Setting*)
ACTION :	BOOL := 0;(*0 = Reverse (Heating or Cooling) 1 = Direct*)
CYCLE_TIME :	REAL := 2.5e-1;(*Execute Cycle time (normally 250 ms)*)
OFF_POSITION :	REAL := 0.0e0;(*Position while not enabled*)
LOW_LIMIT :	REAL := 0.0e0;(*Low Limit of Output in operation*)
HI_LIMIT :	REAL := 1.0e2;(*High Limit of Output in operation*)

OUTPUTS

OUT :	REAL;(*Output Signal*)



Modulating Control with reset (Generic)

Description*:* Provides control of any modulating device such as a cooling or heating valve or damper, or a pressure controlled VFD, based on the *Control Input* (IN). The *Enable Input* (ENB) enables or disables the block. When disabled, OUT will be set to 0.0%

When enabled, the block will modulate OUT (0 – 100%) based on IN and the *Calculated Setpoint*. The Setpoint is calculated using the listed reset parameters (See RESET FB for more details).

(Control ACTION may be set to either:

- 0 (Heating or Reverse Acting). OUT increases as IN decreases.
- 1 (Cooling or Direct Acting). OUT increases as IN increases.

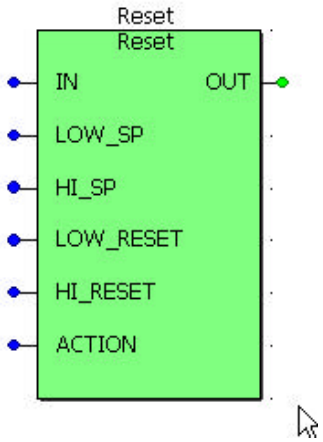
INPUTS

ENB :	BOOL;(*Input to enable control*)
IN :	REAL;(*Input to be controlled*)
PGAIN :	REAL := 1.5e1;(*P Gain Setting*)
IGAIN :	REAL := 1.0e-2;(*I Gain Setting*)
DGAIN :	REAL := 0.0e0;(*D Gain Setting*)
BIAS :	REAL := 2.0e1;(*Bias Setting*)
ACTION :	BOOL := 0;(*0 = Reverse or Heating 1= Direct or Cooling*)
RESET_IN :	REAL;(*Input to control reset*)
LOW_SP :	REAL;(*Lowest Setpoint in range*)
HI_SP :	REAL;(*Highest Setpoint in range*)
LOW_RESET :	REAL;(*Lowest Reset value in range*)
HI_RESET :	REAL;(*Highest Reset value in range*)
RESET_ACTION :	BOOL := 0;(*0 = Reverse (Heating or Cooling) 1 = Direct*)

OUTPUTS

OUT :	REAL;(*Output Signal*)
CALC_SP :	REAL;(*Current Calculated Setpoint*)

Reset Calculation



Description: Calculates a setpoint based on a linear reset schedule derived from user entered inputs for the endpoints of both the Setpoint and Reset ranges.

Control ACTION Type = Reverse

The *Calculated Setpoint* will increase as the *Setpoint Reset Input* Decreases.

Control ACTION Type = Direct

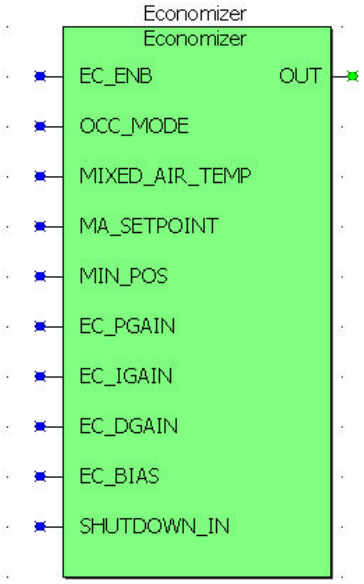
The *Calculated Setpoint* will increase as the *Setpoint Reset Input* Increases.

INPUTS

IN :	REAL;(*Reset Input Value*)
LOW_SP :	REAL;(*Lowest Setpoint Value*)
HI_SP :	REAL;(*Highest Setpoint Value*)
LOW_RESET :	REAL;(*Lowest Reset Input Value*)
HI_RESET :	REAL;(*Highest Reset Input Value*)
ACTION :	BOOL := 0;(*0 = Reverse (Heating or Cooling) 1 = Direct*)

OUTPUTS

OUT :	REAL;(*Output Calculated Setpoint Value*)



Economizer Control

Description: Provides control of a Modulating Outside Air Damper to be used for “free” or “economizer” cooling.

Control is based on the HVAC unit’s Mixed Air (MA) temperature. The Staged Cooling FB typically enables or disables the economizer FB.

Once enabled, this block will modulate *Economizer* damper position (OUT) 0 – 100% based on the MIXED AIR TEMP and MA_SETPOINT.

During OCC MODE (On) the OA Damper will not be allowed to close to less than *OA Damper Min Position,* regardless of the resulting MA Temperature).

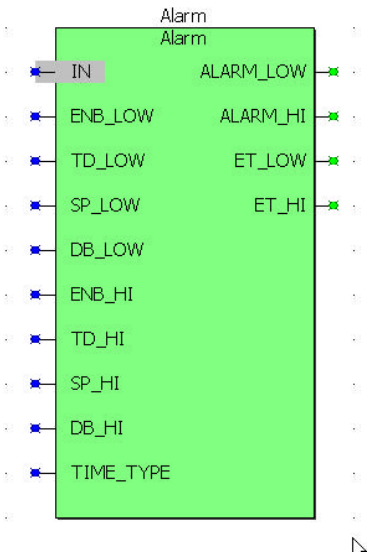
INPUTS

EC_ENB :	BOOL;(*Enable Cooling Input*)
OCC_MODE :	BOOL;(*Occupancy Mode Input*)
MIXED_AIR_TEMP :	REAL;(*Mixed Air Temperature Input*)
MA_SETPOINT :	REAL := 5.5e1;(*Current Economizer Setpoint*)
MIN_POS :	REAL := 1.5e1;(*Modulating Economizer Min Position (0 - 100%)*)
EC_PGAIN :	REAL := 1.0E0;(*P Gain Setting*)
EC_IGAIN :	REAL := 1.0E-2;(*I Reset Time Setting*)
EC_DGAIN :	REAL := 0.0E0;(*D Gain Setting*)
EC_BIAS :	REAL := 2.0E1;(*Output Starting Position*)
SHUTDOWN_IN :	BOOL := 0;(*Emergency Shutdown Input*)

OUTPUTS

OUT :	REAL;(*Economizer Modulating Output (0 - 100%)*)

Alarm



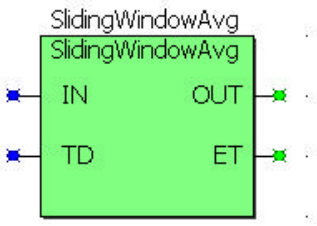
Description: Provides High and Low Zone temperature Alarming capabilities while enabled, based on user entries. TD_LOW sets the amount of time the value IN may be below SP_LOW before an Alarm is generated. The Alarm and the timer will be reset anytime IN > SP_LOW + DB_LOW. The High Alarm mirrors the Low Alarm's operation for IN > SP HI.

INPUTS

IN :	REAL;(*Input to Monitor*)
ENB_LOW :	BOOL;(*Enable Low Alarm*)
TD_LOW :	REAL := 1.0e1;(*Time Delay Low alarm*)
SP_LOW :	REAL;(*Alarm Setpoint Low Alarm*)
DB_LOW :	REAL;(*Off Deadband Low Alarm*)
ENB_HI :	BOOL;(*Enable High Alarm*)
TD_HI :	REAL := 1.0e1;(*Time Delay High alarm*)
SP_HI :	REAL;(*Alarm Setpoint High Alarm*)
DB_HI :	REAL;(*Off Deadband High Alarm*)
TIME_TYPE :	INT := 1;(*0= Seconds, 1= Minutes, 2= Hours*)

OUTPUTS

ALARM_LOW :	BOOL;(*Low Alarm Output*)
ALARM_HI :	BOOL;(*High Alarm Output*)
ET_LOW :	REAL;(*Elapsed Time Low Alarm*)
ET_HI :	REAL;(*Elapsed Time High Alarm*)



Sliding Window Average

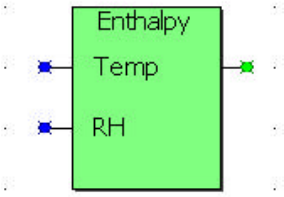
Description*:* Smoothes out fluctuating values by performing a sliding window average of the previous ten readings of the IN value using the user defined number of seconds between reads (TD).

INPUTS

IN :	REAL;(*Input Value to Average*)
TD :	REAL := 5.0e0;(*Time Delay between input reads in seconds*)

OUTPUTS

OUT :	REAL;(*Averaged Value Output*)
ET :	REAL;(*Elapsed Time Cycle Timer*)



Enthalpy Calculation

Description: Calculates the Total Heat Content for one zone or air stream. Typically two zones or air streams are compared and the air stream with the least total heat content is identified as the lowest cost cooling source.

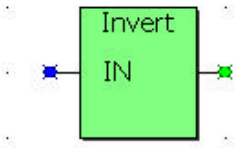
In the case of the Economizer FB, if the ECON_ENB_TYPE is Enthalpy, if Outside Air has the lowest heat content, the Zone Supervisor FB will set the *Economizer Enabled* value ON.

INPUTS

Temp :	REAL;(*Temperature input for this Enthalpy Calculation*)
RH :	REAL;(*Humidity input for this Enthalpy Calculation*)

OUTPUTS

ENTHALPY	REAL;(*Calculated Enthalpy Value*)



Signal Inversion

Description: Provides the inverse signal in the 0 – 100% range.

Usage Example: An *Input Signal* of 30.0 % would result in an *Output Signal* of 70.0 %

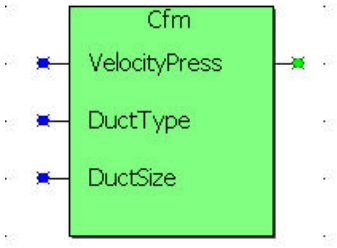
INPUTS

IN :	REAL;(*Input Value to be inverted (0-100%)*)

OUTPUTS

INVERT	REAL;(*Inverted Output Value (100-0%)*)

CFM Calculation



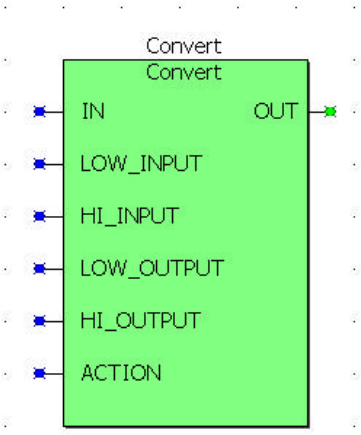
Description: Converts measured Velocity Pressure into CFM airflow, using the parameters listed.

INPUTS

VelocityPress :	REAL;(*Velocity Pressure Input*)
DuctType :	INT;(*0= Rectangular 1=Round*)
DuctSize :	REAL;(*if Rectangular, area in SQ FT, if Round, diam in inches*)

OUTPUTS

CFM : REAL;(*Inverted Output Value (100-0%)*)



Convert

Description: Provides a linear conversion from one user definable signal range to another.

Usage Example (Hardware Input to Eng. Units): A signal in the range of 4 – 20 mA may be converted to engineering units of 40.0 – 90.0 Degrees F.

Usage Example (Eng. Units to Hardware Output): A value in Engineering units of 0 –100 % may be converted to a signal range of 2 – 10 VDC.

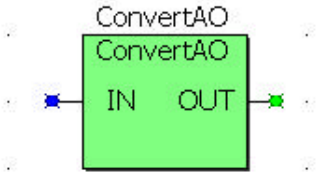
INPUTS

IN :	REAL;(*Control Input Value*)
LOW_INPUT :	REAL;(*Low Input end of Range*)
HI_INPUT :	REAL;(*High Input end of Range*)
LOW_OUTPUT :	REAL;(*Low Output end of Range*)
HI_OUTPUT :	REAL;(*High Output end of Range*)
ACTION :	BOOL := 1;(*0 = Reverse 1= Direct (Direction/Action)*)

END_VAR

OUTPUTS

OUT : REAL;(*Output Signal*)



Convert AO

(Special Case of Convert)

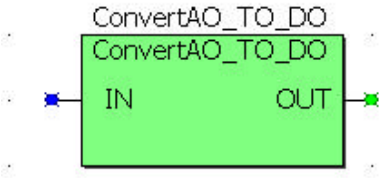
Description: Provides a linear conversion from 0 – 100% IN to 0 – 256 units OUT. This provides the required interface between the logical 0 – 100% signal and the required Unsigned Integer signal to physical Analog Output points.

INPUTS

IN : REAL;(*Input Signal (0-100%)*)

OUTPUTS

OUT : UINT;(*Output Signal (256 steps)*)



Convert AO to DO

(Special Case of Convert)

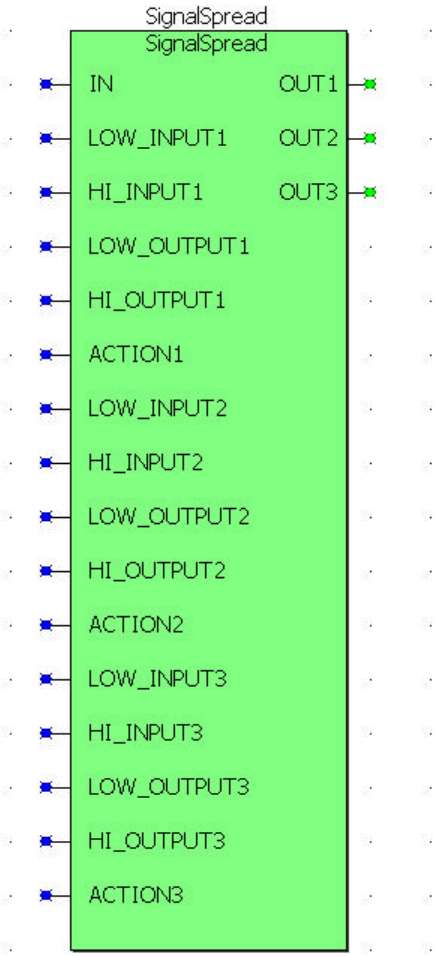
Description: Provides the ability to use AO points as DO. A Boolean IN of 0 is converted to an Unsigned Integer value of 0, while a Boolean IN of 1 is converted to an Unsigned Integer value of 256.

INPUTS

IN : BOOL;(*Input Signal (0-100%)*)

OUTPUTS

OUT : UINT;(*Output Signal (256 steps)*)



Signal Spread

Description: A single input signal is "split" into three output signals, each with it's own set of min and max input and output settings. When the input signal is less than or equal to a particular output's Minimum Input Value, the output will equal the user selected Minimum Output Value. When the input signal is greater than or equal to a particular output's Maximum Input Value, the output will equal the Maximum Output Value. When the input signal is between these two points, the output value will track the input value proportionally.

Usage Example: A single temperature control signal (0 – 100%) may be "spread" among three outputs as follows:

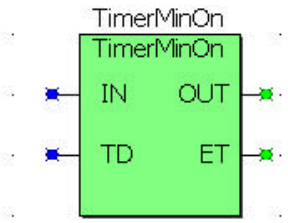
Input Value	Heat Output	Econ. Output	Cool Output
0 – 33 %	100 – 0 %		
34 – 66 %		0 – 100 %	
67 – 100 %			0 – 100 %

INPUTS

IN :	REAL;(*Control Input Value*)
LOW_INPUT1 :	REAL;(*Low Input end of Range (Range 1)*)
HI_INPUT1 :	REAL;(*High Input end of Range (Range 1)*)
LOW_OUTPUT1 :	REAL;(*Low Output end of Range (Range 1)*)
HI_OUTPUT1 :	REAL;(*High Output end of Range (Range 1)*)
ACTION1 :	BOOL;(*0 = Reverse 1= Direct (Direction/Action - Range 1)*)
LOW_INPUT2 :	REAL;(*Low Input end of Range (Range 2)*)
HI_INPUT2 :	REAL;(*High Input end of Range (Range 2)*)
LOW_OUTPUT2 :	REAL;(*Low Output end of Range (Range 2)*)
HI_OUTPUT2 :	REAL;(*High Output end of Range (Range 2)*)
ACTION2 :	BOOL;(*0 = Reverse 1= Direct (Direction/Action - Range 2)*)
LOW_INPUT3 :	REAL;(*Low Input end of Range (Range 3)*)
HI_INPUT3 :	REAL;(*High Input end of Range (Range 3)*)
LOW_OUTPUT3 :	REAL;(*Low Output end of Range (Range 3)*)
HI_OUTPUT3 :	REAL;(*High Output end of Range (Range 3)*)
ACTION3 :	BOOL;(*0 = Reverse 1= Direct (Direction/Action - Range 3)*)

OUTPUTS

OUT1 :	REAL;(*Output Signal Range 1*)
OUT2 :	REAL;(*Output Signal Range 2*)
OUT3 :	REAL;(*Output Signal Range 3*)



Minimum On Timer

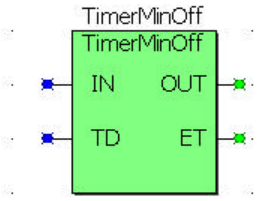
Description: *Minimum On Satisfied* (OUT) will be set On once the *Monitored Value* (IN) has been On at least the amount of time specified in *Min On Time Time Delay (TD) in seconds.*

INPUTS

IN :	BOOL := 0;(*Input Value to Read*)
TD :	REAL;(*Min On Time Delay in Seconds*)

OUTPUTS

OUT :	BOOL;(*Min On Time Satisfied*)
ET :	REAL;(*Elapsed Time*)



Minimum Off Timer

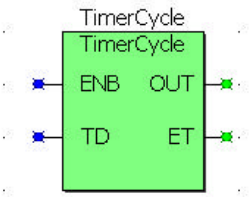
Description: *Minimum Off Satisfied* (OUT) will be set On once the *Monitored Value* (IN) has been Off at least the amount of time specified in *Min Off Time Time Delay (TD) in seconds.*

INPUTS

IN :	BOOL := 0;(*Input Value to Read*)
TD :	REAL;(*Min On Time Delay in Seconds*)

OUTPUTS

OUT :	BOOL;(*Min Off Time Satisfied*)
ET :	REAL;(*Elapsed Time*)



Cycle Timer

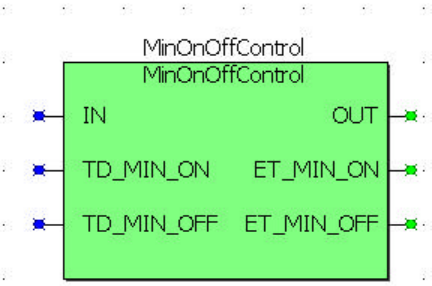
Description: If ENB is On, OUT will be set On (for one scan) each time the Time Delay (TD) in seconds is satisfied. This cycle will be repeated as long as ENB is On.

INPUTS

ENB :	BOOL;(*Enable input to start cycle*)
TD :	REAL;(*Time Delay for each cycle in Seconds*)

OUTPUTS

OUT :	BOOL;(*Output Pulses On once per cycle*)
ET :	REAL;(*Elapsed Time*)



Minimum On Off Control

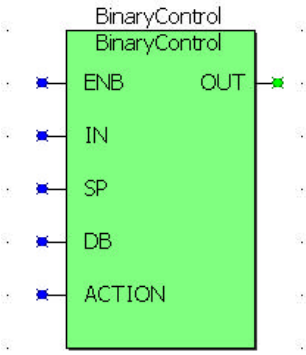
Description: This FB combines a Min On Timer, a Min Off Timer and internal logic such that a request to run a piece of equipment is made by setting IN On. Once the Minimum Off timer is satisfied, the FB will set OUT On. A request to stop a piece of equipment is made by setting IN Off. Once the Minimum On timer is satisfied, the FB will set OUT Off.

INPUTS

IN :	BOOL := 0;(*Input Signal to Start Equipment*)
TD_MIN_ON :	REAL;(*Time Delay for Minimum On Timer in Seconds*)
TD_MIN_OFF :	REAL;(*Time Delay for Minimum Off Timer in Seconds*)

OUTPUTS

OUT :	BOOL;(*Output Signal to Control Equipment*)
ET_MIN_ON :	REAL;(*Elapsed Time Min On Timer*)
ET_MIN_OFF :	REAL;(*Elapsed Time Min Off Timer*)



Binary Control

Description: When enabled, this FB will control OUT to maintain a Setpoint (SP) based upon the measured Input value (IN).

Control ACTION Type = Reverse

If ENB is ON: OUT will be set ON when IN <= SP. OUT will be set off when IN >= SP + DB. If ENB is OFF Out will be Off.

Control ACTION Type = Direct

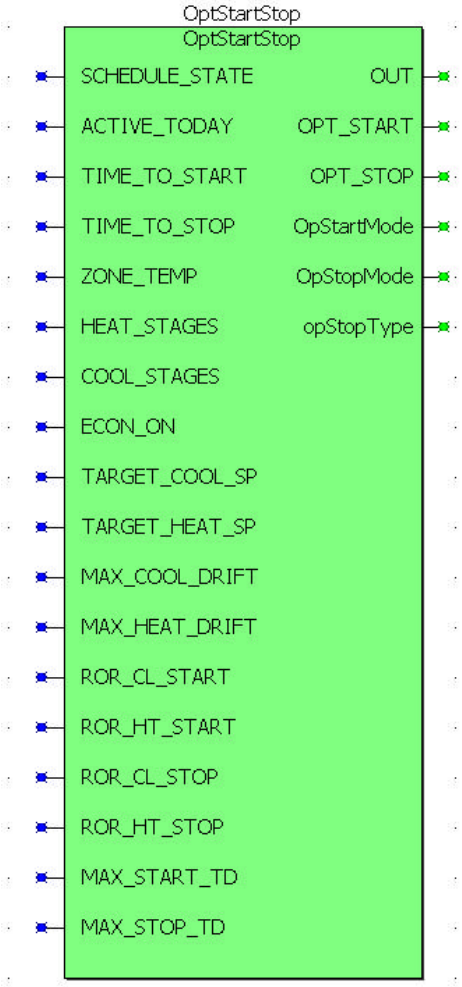
If ENB is ON:OUT will be set ON when IN >= SP. OUT will be set off when IN <= SP - DB. If ENB is OFF Out will be Off.

INPUTS

ENB :	BOOL;(*Enable Value*)
IN :	REAL;(*Input Value*)
SP :	REAL;(*Setpoint for Input*)
DB :	REAL;(*Deadband or Differential for Setpoint*)
ACTION :	BOOL := 0;(*0 = Heat/Reverse 1= Cool/Direct*)

OUTPUTS

OUT :	BOOL;(*Output Signal*)

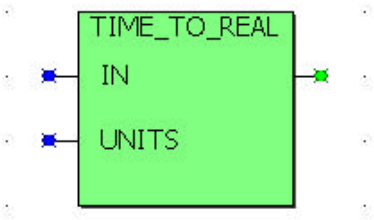


Optimum Start Stop

Description: This FB is not intended to only be used stand-alone, only as part of the Schedule with Optimum Start / Optimum Stop FB.

```
VAR_INPUTS
 SCHEDULE_STATE : BOOL;(*Input from Schedule (Off or On)*)
 ACTIVE_TODAY :   BOOL := 1;(*Input On if Schedule is used Today*)
 TIME_TO_START :  REAL := 2.0e0;(*Hours until Next Start Time if Schedule is Active Today*)
 TIME_TO_STOP :   REAL := 12.0e0;(*Hours until Next Stop Time if Schedule is Active Today*)
 ZONE_TEMP :      REAL := 6.9e1;(*Zone Temp input*)
 HEAT_STAGES :    INT;(*Heat Stages input from Heating FB (Any Int over 0) for Op Stop*)
 COOL_STAGES :    INT;(*Cool Stages input from Heating FB (Any Int over 0) for Op Stop*)
 ECON_ON :        BOOL;(*Econ Active input from Econ FB (Off or On) for Op Stop*)
 TARGET_COOL_SP : REAL := 7.4e1;(*Opt Start Cooling Target (May link to Cool SP)*)
 TARGET_HEAT_SP : REAL := 7.0e1;(*Opt Start Heating Target (May link to Cool SP)*)
 MAX_COOL_DRIFT : REAL := 2.0e0;(*Opt Stop Maximum allowed Cool Target to drift to*)
 MAX_HEAT_DRIFT : REAL := 2.0e0;(*Opt Stop Minimum allowed Heat Target to drift to*)
 ROR_CL_START :   REAL := 2.5e0;(*Opt Start Cooling Rate of Recovery  deg/hr*)
 ROR_HT_START :   REAL := 2.5e0;(*Opt Start Heating Rate of Recovery  deg/hr*)
 ROR_CL_STOP :    REAL := 2.5e0;(*Opt Stop Cooling Rate of Recovery  deg/hr*)
 ROR_HT_STOP :    REAL := 2.5e0;(*Opt Stop Heating Rate of Recovery  deg/hr*)
 MAX_START_TD :   REAL := 1.0e0;(*Maximum allowed early start time (hours)*)
 MAX_STOP_TD :    REAL := 1.0e0;(*Maximum allowed early stop time (hours)*)

VAR_OUTPUTS
 OUT :            BOOL;(*Combined Sched & OpStart & OpStop Output to enable device*)
 OPT_START :      BOOL;(*On during Opt Start period*)
 OPT_STOP :       BOOL;(*On during Opt Stop period*)
 OpStartMode :    INT;(*Current Opt Start Mode (0=off, 1=heat waiting,2=cool waiting, 3=heat OpStart Active, 4=cool OpStart Active
 OpStopMode :     INT;(*Current Opt Stop Mode (0=off, 1=heat waiting,2=cool waiting, 3=heat OpStop Active, 4=cool OpStop Active
 opStopType :     INT;(*Current Opt Stop Type set by monitoring heat/cool/econ operation (0=none,1=heat,2=cool)*)
```



Time To Real

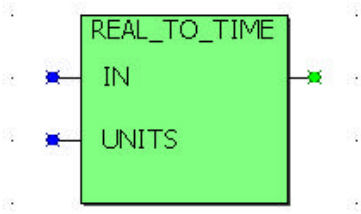
Description: Converts values in a time format to values in a real format, using the user selected Units, seconds, minutes or hours.

INPUTS

```
 IN :     TIME;(*Input Signal in Time Format*)
 UNITS :  INT;(*0= Seconds, 1= Minutes, 2= Hours*)
```

OUTPUTS

```
   Value                    Real;(*Output Signal*)
```



Real To Time

Description: Converts values in a real format to values in a time format, using the user selected Units, seconds, minutes or hours.

INPUTS

IN :	REAL;(*Input Signal in Real Format*)
UNITS :	INT;(*0= Seconds, 1= Minutes, 2= Hours*)

OUTPUTS

Value	Time;(*Output Signal*)